MODELLING AND DYNAMIC STABILISATION OF A COMPLIANT HUMANOID ROBOT, CoMan

A thesis submitted to the University of Manchester for the degree of Doctor of Philosophy in the Faculty of Engineering and Physical Sciences

2012

Houman Dallali
School of Electrical and Electronic Engineering
Contents

Abstract .......................................................... 23

Declaration ......................................................... 25

Copyright Statement ............................................. 26

Acknowledgements ............................................... 27

1 Introduction .................................................... 29
   1.1 Statement of the Problem .................................. 29
   1.2 Aims and Objectives ....................................... 30
   1.3 Thesis Contributions ....................................... 31
   1.4 Thesis Outline ............................................. 33
   1.5 List of Publications ....................................... 35
       1.5.1 Journals ............................................... 35
       1.5.2 Conferences ........................................... 35

2 Background Review ............................................ 37
   2.1 Humanoid Robots .......................................... 38
       2.1.1 ASIMO ................................................ 38
       2.1.2 Toyota Partner Robot ................................ 39
       2.1.3 Petman and Atlas ..................................... 40
       2.1.4 Humanoid Robotics Project .......................... 40
       2.1.5 WABIAN-2R and KOBIAN ............................... 42
       2.1.6 HUBO .................................................. 43
4.2.4.1 Simulation Set-up ........................................... 102
4.2.4.2 Discussion .................................................. 102
4.3 Decentralized LQR with Sparse Gradient Descent Feedback Design 104
  4.3.1 Background .................................................. 105
  4.3.2 Contribution .................................................. 105
  4.3.3 Problem Formulation ........................................ 105
  4.3.4 LQR-GraDe Simulation for Rigid Model of CoMan .............. 109
  4.3.5 LQR-GraDe Simulation for Compliant Model of CoMan ....... 112
4.4 Decentralized LMI-LQR Feedback Design .......................... 116
  4.4.1 Background .................................................. 116
  4.4.2 Contribution .................................................. 118
  4.4.3 Problem Formulation ........................................ 118
  4.4.4 Decentralized and Symmetric Gains .......................... 121
  4.4.5 LMI Well-Posedness ......................................... 122
  4.4.6 LMI-LQR Simulation for the Rigid Model of CoMan ............ 123
  4.4.7 LMI-LQR Simulation for the Compliant Model of CoMan ...... 125
  4.4.8 Future Work .................................................. 128
4.5 Velocity Observer .................................................. 129
  4.5.1 Difference and Averaging ................................... 130
  4.5.2 Luenberger Observer ........................................ 131
  4.5.3 Unknown Input Observer ..................................... 132
  4.5.4 Relative Stability .......................................... 134
4.6 Conclusions ......................................................... 136

5 Double Support Geometric Control 137
  5.1 Background ..................................................... 138
  5.2 Contribution ..................................................... 139
  5.3 Preliminary Definitions ......................................... 139
  5.4 Constrained Control Design ..................................... 143
    5.4.1 Control Design Method ..................................... 144
5.5 The Geometric Constrained Subspace .............................. 148
5.5.1 The Constrained Subspace for the Rigid Model .......... 148
5.5.2 The Constrained Subspace for the Compliant Model ... 149
5.6 Simulation of Double Support Control of the Complaint Model of CoMan 151
5.6.1 Controller Design .............................................. 152
5.6.2 Simulation Results ............................................. 154
5.7 Conclusions ....................................................... 158

6 Walking Trajectory Generation .................................. 160
6.1 Trajectory Generation Using Simple Models ................. 161
6.1.1 Linear Single Inverted Pendulum and the Cart-Table Model 161
6.1.2 Double Inverted Pendulum Model ............................ 164
6.1.3 Extended Two Link Inverted Pendulum Model .............. 166
6.1.4 Pattern generation by ZMP preview control .............. 167
6.1.5 ZMP Stabilization by Using the Torso ..................... 168
6.1.6 Comparison and Simulation Results ......................... 169
6.2 Selection of Walking Parameters ............................... 170
6.2.1 Assumptions and Set-up ..................................... 173
6.2.2 Bandwidth Requirement ..................................... 174
6.2.3 Mechanical Joint Limits ................................... 175
6.2.4 Bent Knees .................................................... 176
6.2.5 Use of Toes ................................................... 177
6.2.6 Use of Toes and Knees ..................................... 178
6.2.7 Torque Requirement ......................................... 179
6.2.8 Maximum Angular Velocity .................................. 179
6.2.9 ZMP Dynamic Limitation ................................... 180
6.2.10 Capture Point ............................................... 183
6.2.11 Numerical Examples ....................................... 183
6.3 Conclusions and Open Problems ............................... 185
7 Development of a Dynamic Walking Simulator

7.1 Development of CoMan’s Model in Robotran

7.1.1 The Single Support Model in Robotran’s GUI

7.1.2 CoMan’s Kinematics and Sensors

7.1.3 Joint Torques

7.1.4 Compliance and Actuator Dynamics

7.1.5 External Forces and Ground Reaction Force

7.1.6 Linearization and LQR Control Design

7.1.6.1 Inverse Dynamics

7.1.7 Effects of Quantization and Stiction

7.2 Generating the Walking Trajectories

7.2.1 Computing Posture Angles by Inverse Kinematics

7.2.2 Trajectory Smoothing

7.3 Robotran’s Simulation Results

7.4 Development of CoMan’s Model in MapleSim

7.5 Model Validation

7.6 Conclusions and Future Work

8 Experimental Results

8.1 Validation of Observer Based LQR on a Compliant Joint Unit

8.1.1 Description of the Prototype

8.1.2 Modelling and Controller Design

8.1.3 Reduced Order Observer Design

8.1.4 Simulation and Experimental Results

8.1.4.1 Tracking Performance

8.1.4.2 Validation of the Observer Estimates

8.2 Implementation of Observer Based LQR on CoMan

8.2.1 Homing the Joints’ Positions

8.2.2 Modelling and Control Design Details

8.3 Velocity Estimation
8.3.1 Experimental Set-up .............................................. 238
8.3.2 Difference and Averaging .................................... 239
8.3.3 Luenberger Observer ............................................ 243
8.3.4 Unknown Input Observer ....................................... 245
8.4 Implementation of Sway Motions Using the LQR Control ........... 249
  8.4.1 Sway Motion Experimental Set-up .......................... 250
  8.4.2 Sway Motion Experimental Results ......................... 250
8.5 Implementation of Squat Motions Using the LQR Control .......... 253
  8.5.1 Squat Motion Experimental Set-up .......................... 255
  8.5.2 Squat Motion Experimental Results ......................... 255
8.6 Sway Tests Using the PID Control ................................ 260
  8.6.1 PID Experimental Set-up ................................. 263
  8.6.2 PID Experimental Results ................................. 263
8.7 Tracking Walking Trajectories Using the PID Control ............ 266
  8.7.1 PID Experimental Set-up ................................. 266
  8.7.2 PID Experimental Results ................................. 269
8.8 Tracking Walking Trajectories Using the LQR Control ............ 273
  8.8.1 LQR Experimental Set-up ................................. 274
  8.8.2 LQR Experimental Results ................................. 275
8.9 Technical Challenges ............................................ 278
  8.9.1 Communication Bandwidth .................................. 280
  8.9.2 Cable Drives on the Ankles ............................... 281
  8.9.3 Homing and Calibration of Multiple Sensors .............. 283
  8.9.4 Limit Cycles due to Stiction and Quantization ............ 287
  8.9.5 High Frequency Oscillatory Behaviour ...................... 288
8.10 Conclusions ...................................................... 291

9 Conclusions and Future Work ........................................ 294
  9.1 Future Work ..................................................... 298

Appendices .............................................................. 299
## List of Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1</td>
<td>Phases and events of the walking hybrid model</td>
<td>83</td>
</tr>
<tr>
<td>3.2</td>
<td>Ground model coefficients</td>
<td>92</td>
</tr>
<tr>
<td>4.1</td>
<td>CoMan’s independent joint PID gains (right leg)</td>
<td>97</td>
</tr>
<tr>
<td>6.1</td>
<td>Maximum feasible walking speed for CoMan (with flat feet)</td>
<td>184</td>
</tr>
<tr>
<td>7.1</td>
<td>Parameters of the ground contact model</td>
<td>194</td>
</tr>
<tr>
<td>7.2</td>
<td>The joints trajectories for the 5 postures of the walking cycle</td>
<td>202</td>
</tr>
<tr>
<td>7.3</td>
<td>The swing foot and CoM positions corresponding to the 5 postures</td>
<td>202</td>
</tr>
<tr>
<td>8.1</td>
<td>Compliant joint’s model parameters (reflected to the joint side)</td>
<td>221</td>
</tr>
<tr>
<td>8.2</td>
<td>The tracking bandwidth corresponding to various LQR input penalties</td>
<td>234</td>
</tr>
<tr>
<td>C.1</td>
<td>Mechanical parameters of CoMan</td>
<td>307</td>
</tr>
<tr>
<td>C.2</td>
<td>CoMan’s actuator parameters</td>
<td>309</td>
</tr>
<tr>
<td>D.1</td>
<td>Compass gait parameters</td>
<td>314</td>
</tr>
<tr>
<td>E.1</td>
<td>CoMan’s calibration data (Angles in degree and link encoder positions are in counts)</td>
<td>315</td>
</tr>
<tr>
<td>E.2</td>
<td>CoMan’s calibration data (Angles in degree and link encoder positions are in counts)</td>
<td>316</td>
</tr>
<tr>
<td>E.3</td>
<td>CoMan’s input(voltage)-output(motor/joint position) polarity convention</td>
<td>316</td>
</tr>
</tbody>
</table>
List of Figures

2.1 Development history of robots that led to the design of ASIMO [17],
courtesy of Honda Motor Co. Ltd. ........................................ 39
2.2 HRP-2 (a), HRP-3 (b), HRP-4 (c) and HRP-4C (d) developed in
Japan. Courtesy of Kawada Industries, Inc. and National Institute
of Advanced Industrial Science and Technology [26, 27]. .......... 42
2.3 WABIAN-2R (left) and KOBIAN (right) built at Waseda University
in Japan [31], courtesy of Prof. Takanishi’s biped humanoid robot group. 43
2.4 Albert Hubo (left), Hubo (middle) and Hubo FX-1 (right), courtesy
of Korean Institute of Science and Technology (KAIST) [32]. ...... 44
2.5 JOHNNIE (left) and LOLA (right), courtesy of the Institute of Applied
Mechanics, Technical University of Munich in Germany [38]. ...... 45
2.6 Computational Brain (CB) developed by JST-ICORP Computational
Brain Project and Sarcos company [40]. Courtesy of Prof. Jun Morimoto. 46
2.7 Rabbit (left) built in Grenoble, France and Mable (right) built at Uni-
versity of Michigan, courtesy of Prof. Christine Chevallereau and Prof.
Jessy Grizzle [10, 47]. ......................................................... 47
2.8 Pneumatic biped Lucy, courtesy of the Robotics and Multibody Me-
chanics Research Group of the Department of Mechanical Engineering
at the Vrije University of Brussel [48]. ................................. 48
2.9 iCub humanoid robot, courtesy of Istituto Italiano di Tecnologia [49]. 49
2.10 CoMan legs, courtesy of the Department of Advanced Robotics at
Istituto Italiano di Tecnologia. .......................................... 51
2.11 The centralized LQR (solid line) and decentralized PID (dashed line) feedback architectures. The time derivatives are denoted by a dot and the estimates are denoted by a hat sign. ................. 53

2.12 The compliant prototype leg used to demonstrate the result of distributed control [64]. Courtesy of Robotics Research Lab, at the Department of Computer Sciences, University of Kaiserslautern. ....... 56

2.13 The compliant Jena Walker II, courtesy of Lauflabor Locomotion Lab, University of Jena [66]. ......................... 57

2.14 A coupled neural oscillators (left) and a typical stable limit cycle (right). 62

2.15 Virtual forces applied to a robot. .......................... 64

3.1 (a) The compliant CoMan legs. (b) The corresponding revolute joints, R, P & Y denote Roll, Pitch and Yaw rotations, respectively. ........ 80

3.2 (a) The compact series elastic actuator. (b) The sagittal ankle joint and the passive DoF at the toe. ................................. 80

3.3 A full cycle of walking, [159]. ............................. 82

3.4 Hybrid representation of a full walking cycle with events and transitions between continuous phases of motion. ..................... 82

3.5 CoMan angle conventions. a) Pitch DoF in \( Z - X \) plane with positive rotations about \( Y \) axis in the direction of the arrow. b) Roll DoF in \( Z - Y \) plane with negative rotations about \( X \) axis in the direction of the arrow. ................................. 84

3.6 Model of the legs in double support with feet separation denoted by \( d \). The area enclosed in the dashed line forms a closed kinematic chain that has a constrained motion. The 3 DoF in the waist are not part of the closed kinematic chain and can have independent motion. .... 86

3.7 Electrical circuit of a DC motor. ............................. 88

3.8 Mechanical diagram of a compliant joint driving a load inertia. \( N \) represents the gearbox reduction ratio. ............................. 89
3.9 The simplified model of ground reaction force for each foot. The spring-damper force is applied only when the spring-dampers are compressed to simulate the unilateral direction of the ground reaction force.

4.1 Proposed CoMan’s control system architecture.

4.2 Independent PID joint control structure for a rigid joint.

4.3 Independent joint control structures for a compliant joint. (a) The proposed feedback structure. (b) CoMan’s existing feedback structure. The power amplifier, sensor, noise and disturbance are omitted for simplicity.

4.4 The white configuration of the robot legs represents the initial condition (that is the upright position) and the coloured configuration of the legs represents the desired and final posture of the legs that corresponds to \( r \).

4.5 LQR closed loop step response.

4.6 LQR closed loop control voltages.

4.7 Cost of the LQR objective function \( \hat{J}(K_i) \) shown against the number of iterations.

4.8 A comparison between CoMan’s decentralized (dashed line) and centralized LQR (solid line) joints’ positions step response.

4.9 A comparison between the decentralized (dashed line) and centralized LQR (solid line) joints’ torques step response.

4.10 Centralized and decentralized LQR cost \( \hat{J}(K_i) \) against iterations.

4.11 CoMan joint positions step response. The support leg is denoted by dashed line and the swing leg is denoted by a solid line.

4.12 CoMan motor voltages corresponding to the step response.

4.13 Closed loop step response of CoMan’s lateral (top) and sagittal (bottom) joint angles using the decentralized gains.

4.14 The control input (torques) of CoMan’s lateral (top) and sagittal (bottom) joints using the decentralized gains.
<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.15</td>
<td>Closed loop step response of CoMan’s lateral (top) and sagittal (bottom) joint angles using the symmetric and decentralized gains.</td>
<td>125</td>
</tr>
<tr>
<td>4.16</td>
<td>The control input (torques) of CoMan’s lateral (top) and sagittal (bottom) joints using the symmetric and decentralized gains.</td>
<td>126</td>
</tr>
<tr>
<td>4.17</td>
<td>Closed loop step response of CoMan’s lateral joint and motor angles.</td>
<td>127</td>
</tr>
<tr>
<td>4.18</td>
<td>Closed loop step response of CoMan’s sagittal joint and motor angles.</td>
<td>127</td>
</tr>
<tr>
<td>4.19</td>
<td>Closed loop decentralized controller voltages sent to the motors.</td>
<td>128</td>
</tr>
<tr>
<td>4.20</td>
<td>Control system diagram of observer based LQR. The mark “x” is used to show where the loop is broken to investigate the relative stability.</td>
<td>130</td>
</tr>
<tr>
<td>4.21</td>
<td>The determinant of the return difference at the input of the plant as a measure of relative stability margins among different observers.</td>
<td>135</td>
</tr>
<tr>
<td>5.1</td>
<td>Constrained motion of CoMan in the double support phase.</td>
<td>140</td>
</tr>
<tr>
<td>5.2</td>
<td>A conceptual illustration of the constrained dynamics flow in the double support phase.</td>
<td>141</td>
</tr>
<tr>
<td>5.3</td>
<td>Graphical representation of the constrained subspace $V$ and its orthogonal complement $V^\perp$.</td>
<td>145</td>
</tr>
<tr>
<td>5.4</td>
<td>Joint angles in lateral (left) and sagittal (right) planes in double support.</td>
<td>155</td>
</tr>
<tr>
<td>5.5</td>
<td>Joint velocities in lateral (left) and sagittal (right) planes in double support.</td>
<td>155</td>
</tr>
<tr>
<td>5.6</td>
<td>Motor angles in lateral (left) and sagittal (right) planes in double support.</td>
<td>156</td>
</tr>
<tr>
<td>5.7</td>
<td>Motor velocities in lateral (left) and sagittal (right) planes in double support.</td>
<td>156</td>
</tr>
<tr>
<td>5.8</td>
<td>Control inputs (motor voltages) in lateral (left) and sagittal (right) plane in double support.</td>
<td>157</td>
</tr>
<tr>
<td>5.9</td>
<td>The constraints $h(q)$ violation during the dynamic response of the system in X, Y and Z directions.</td>
<td>158</td>
</tr>
</tbody>
</table>
6.1 In (a) constrained linear inverted pendulum is shown that assumes that ZMP is in the origin (0,0,0) (origin corresponds to the ankle of robot) and torque in the ankle is zero. In (b) the cart table model is shown for Z-X plane where the ZMP can move in the foot (table's contact with the ground) and the torque in the ankle does not need to be zero.

6.2 The step response of the cart table model with poles placed at pole1

\[ \text{pole1} = [-2,-4,-6], \text{ and } \text{pole2} = [-4,-6,-10]. \]

6.3 The two mass inverted pendulum model.

6.4 The proposed pendulum-cart-table model in \( X \) direction that includes the upper body.

6.5 ZMP tracking for walking pattern generation.

6.6 The pattern generation using the extended pendulum cart table model (top) and the corresponding control inputs (bottom).

6.7 The role of torso in minimizing the amount of undershoot, hence improving the ZMP stabilization.

6.8 The hip (left) and torso (right) trajectories of the extended cart table model. The plots from top to bottom are position, velocity and acceleration of hip and torso.

6.9 Representation of a more complex pair of legs with the compass gait model, during toe push-off.

6.10 Compass gait model of walking.

6.11 The trigonometric relations of a bent knee can be used to calculate the virtual leg length.

6.12 The trigonometric relations when using the toe joint can be used to calculate the virtual leg length.

6.13 The trigonometric relations when using the toe joint can be used to calculate the step size.

6.14 Toe-off during walking (left), and flat foot walking (right).
6.15 The linear and angular distances travelled by the ankle and the hip during a single step. (The stance leg is denoted by a solid line and the swing leg is distinguished by a dashed line). ........................................ 180
6.16 The ZMP of the LIP model driven by a sine wave in single support phase. .......................................................... 182
6.17 The ZMP of the LIP driven by a sine wave in double support phase. . 182
6.18 Estimated minimum required bandwidth for ASIMO as a function of the average walking speed. ............................. 183
6.19 Minimum required bandwidth for CoMan as a function of the average walking speed. ........................................ 184

7.1 The flowchart illustrating the development of the walking simulator. . 188
7.2 A model of CoMan in Robotran with 10 DoF. ................................. 190
7.3 3D visualization of CoMan programmed separately in Matlab. ...... 191
7.4 Locking a joint in Robotran’s GUI. ........................................ 197
7.5 Magnified illustration of quantization effect on position. ............... 199
7.6 Estimated velocity using first order difference of the quantized position measurements. ........................................ 199
7.7 An example of the 5th order polynomial used for the stance leg’s ankle sagittal joint. .................................................... 203
7.8 Initial posture of the robot in the walking cycle. .......................... 203
7.9 In the second posture of the robot, the centre of gravity is shifted to the support leg. ..................................................... 204
7.10 The robot lifts the trailing leg. .................................................. 204
7.11 The robot swings the trailing leg forward. ................................. 205
7.12 The robot completes the first step in symmetry with the initial posture.205
7.13 The joint trajectories during a full walking cycle (two steps). ....... 207
7.14 Joint tracking results on four joints of the stance leg. ................. 208
7.15 Visualization of the first step with trace of the trajectories’ history. . 208
7.16 The corresponding joints’ torques during the walking cycle. ........ 209
7.17 The cartesian position of the swing foot (top) and the corresponding ground reaction force (bottom). .............................................. 209
7.18 The position of the swing foot (top) and the corresponding robot’s CoM trajectory (bottom). ..................................................... 210
7.19 The model of legs developed in MapleSim 4.5 with X pointing to the forward walking direction. ................................................. 211
7.20 The lateral joints positions during the sway test. ...................... 213
7.21 The lateral joints voltages during the sway test. ...................... 213
7.22 A comparison between the simulation and experimental lateral joints positions during the sway test. ........................................... 214
7.23 A comparison between the simulation and experimental lateral motor positions during the sway test. ........................................... 214
7.24 The lateral joints voltages simulation during the sway test. ........... 215
7.25 Representation of CoMan with a 6 DoF floating base attached between the world frame and the pelvis. The ground reaction forces acting under the feet are only activated when an impact event is detected and the feet height are at the ground level. ........................................... 216
8.1 Close-up view of the compliant joint set up (left) and the final set up of the equipment in the experiment (right). ............................. 220
8.2 Nyquist plot of the stability margins for the compliant joint unit. .. 224
8.3 Sine wave tracking experiment on compliant joint (Experiment). . . 225
8.4 Bode diagram of the closed loop system with the reference signal as the input and the motor position as the output (Simulation). ........ 225
8.5 Comparison between the experimental and simulation input voltage corresponding to the sine wave tracking. ............................ 226
8.6 Comparison between the experimental and the simulation input volt- age with stiction and viscose friction included. ............................ 227
8.7 Comparison between estimated motor velocity derived by the observer and numerical differentiation (Experiment). .......................... 228
8.8 Comparison between estimated motor and link velocities computed by
the observer (Experiment) .................................................. 229
8.9 Reproducing the experimental observer’s estimated motor’s velocity
computed in simulation by including stiction and viscose friction. . . 229
8.10 Reproducing the experimental observer’s estimated link’s velocity com-
puted in simulation by including stiction and viscose friction. . . . . 230
8.11 Comparison between estimated link position and real motor position
(Experiment) ................................................................. 230
8.12 Comparison of experimental link position estimate with the simulation
prediction with stiction and viscose friction included. .............. 231
8.13 Closed loop bode magnitude plot of the 10 joints with the horizontal
axis being the frequency in (rad/sec) .................................... 236
8.14 Diagram of the lateral LQR Feedback Gains (FBG) versus state variables. 236
8.15 Diagram of the sagittal LQR Feedback Gains (FBG) versus state vari-
ables ................................................................. 237
8.16 Determinant of the return difference at the output of the plant with the
observer dynamics included (solid line) and only the feedback (dashed
line). ................................................................. 238
8.17 The hip lateral steps with LQR control and velocities computed by
difference and averaging .................................................. 240
8.18 The hip lateral velocities computed by difference and averaging. . . 240
8.19 The hip lateral voltages ................................................. 241
8.20 The effects of quantization on the right hip lateral position .......... 241
8.21 The effects of quantization on the right hip lateral velocity .......... 242
8.22 The effects of quantization on the right hip lateral voltage .......... 242
8.23 Regulation of the left hip lateral position to zero using the slow observer. 243
8.24 The left hip lateral velocity obtained using difference and averaging
during homing and the slow observer after the homing ............... 244
8.25 The left hip lateral voltage with velocities derived using the slow observer. 244
8.26 The sagittal plane joint positions with velocities derived with the fast observer. ........................................... 246
8.27 The sagittal plane joint velocities derived with the fast observer. ........................................... 246
8.28 The sagittal plane motor voltages with velocities derived using the fast observer. ................................. 247
8.29 The sagittal plane joint positions with velocities derived with the unknown input observer. ......................... 248
8.30 The sagittal plane joint velocities derived with the unknown input observer. ........................................... 248
8.31 The sagittal plane joint voltages with velocities derived using the unknown input observer. ......................... 249
8.32 Snapshots of CoMan doing a sway test. ........................................... 250
8.33 Experimental data from the lateral joints. ........................................... 251
8.34 The tracking errors of the sway test. ........................................... 252
8.35 Experimental data from the right ankle lateral. ........................................... 252
8.36 Slow observer’s velocity estimates from the sway test. ........................................... 253
8.37 Sagittal joints regulated to zero during the sway motion. ........................................... 254
8.38 Snapshots of CoMan doing a squat test (from left to right). ........................................... 254
8.39 Sagittal joints tracking of the first squat test. ........................................... 256
8.40 Sagittal joints tracking errors of the first squat test. ........................................... 257
8.41 Estimated motors’ velocities of the first squat test. ........................................... 257
8.42 Input voltages of the first squat test. ........................................... 258
8.43 Sagittal joints tracking of the second squat test with 2 kg added on the waist. ........................................... 258
8.44 Sagittal joints tracking errors of the second squat test with 2 kg added on the waist. ........................................... 259
8.45 Estimated velocity of motors for sagittal joints during the second squat test with 2 kg added on the waist. ........................................... 259
8.46 Input voltages for sagittal joints during the second squat test. ........................................... 260
8.47 Sagittal joints tracking of the third squat test with 2 kg added on the waist. .......................... 261
8.48 Sagittal joints tracking errors of the third squat test with 2 kg added on the waist. .................. 261
8.49 Estimated velocity of motors for sagittal joints during the third squat test with 2 kg added on the waist. ................... 262
8.50 Input voltages for sagittal joints during the third squat test with 2 kg added on the waist. .............. 262
8.51 Sway’s step response of independent PID control. ................. 264
8.52 Sway’s tracking errors of the PID sway’s step response. .............. 265
8.53 Lateral motors velocities corresponding to the PID sway’s step response. 265
8.54 Lateral input voltages corresponding to the PID sway’s step response. 266
8.55 Sine sway test by independent PID control. ....................... 267
8.56 Lateral tracking errors corresponding to the sine sway test. .......... 267
8.57 Lateral motors velocities corresponding to the sine sway test. .......... 268
8.58 Lateral input voltages corresponding to the sine sway test. .......... 268
8.59 Sagittal joints tracking of the walking trajectory. .................. 269
8.60 Lateral joints tracking of the walking trajectory. .................. 270
8.61 Sagittal joints tracking errors of the walking trajectory. .................. 270
8.62 Lateral joints tracking errors of the walking trajectory. .................. 271
8.63 Sagittal joints’ velocity estimate of the walking trajectory. .................. 272
8.64 Lateral joints’ velocity estimate of the walking trajectory. .................. 272
8.65 Sagittal plane input voltages during the air walking experiment. .... 273
8.66 Lateral plane input voltages during the air walking experiment. .... 274
8.67 Sagittal joints tracking of the walking trajectory using LQR. ......... 275
8.68 Lateral joints tracking of the walking trajectory using LQR. ......... 276
8.69 Sagittal joints tracking errors of the walking trajectory using LQR. .... 276
8.70 Lateral joints tracking errors of the walking trajectory using LQR. .... 277
8.71 Sagittal joints’ velocity estimate of the walking trajectory using LQR. .... 277
8.72 Lateral joints’ velocity estimate of the walking trajectory using LQR. .... 278
8.73 Sagittal plane input voltages during the air walking experiment using LQR.  
8.74 Lateral plane input voltages during the air walking experiment using LQR.  
8.75 A plot of the sample time variations in an experiment on iCub with sample time of 10 $ms$.  
8.76 A plot of the regular sample times in an experiment on CoMan with sample time of 2 $ms$.  
8.77 Robust mechanical design of CoMan’s ankle joint built at the department of advanced robotics at IIT.  
8.78 Oscillations observed in an experiments on CoMan on the ground.  
8.79 Estimated joint’s and motor’s velocity by numerical differentiation.  
8.80 Initial transient leading to oscillations in the experiments on CoMan on the ground (close-up view of Fig. 8.78).  
8.81 Nonlinear simulation of CoMan with an offset between the motors and links positions predicts large oscillations on the ground. The dotted, dash-dotted and dashed lines represent the hip, knee and ankle of the support leg in sagittal plane.  
8.82 Once homing is done, the unstable transient diminishes and the robot responds to external disturbances.  
8.83 Typical oscillations observed experiments on CoMan.  
8.84 Estimated joint’s and motor’s velocity.  
8.85 Experimental data from ankle sagittal with homing and integrators closed on the ankle encoder.  
8.86 Experimental data from knee joint with homing and integrators closed on the knee encoder.  
8.87 Experimental data from sagittal joints’ positions (with homing and integrators closed on the motors’ encoders).  
8.88 Sagittal joints’ voltages corresponding (with homing and integrators closed on the motors’ encoders).
A.1 Diagram of a planar double pendulum .......................... 301

C.1 This diagram illustrate the multi-body model and its various parameters used for CoMan model where bodies are denoted by $B_1, ..., B_9$, lengths (distances) are denoted by $d_0, ..., d_9$ and centers of mass are denoted by $c_1, ... c_9$ ........................................ 306

D.1 Planar compass gait with hip and ankle actuation ................. 310

D.2 Closed-loop step response of actuated compass gait .......... 311

D.3 Control inputs of compass gait closed-loop step response .... 312

D.4 Closed-loop step response of actuated compass gait .......... 314

D.5 Control inputs of compass gait closed-loop step response .... 314
This dissertation presents the results of a series of studies on dynamic stabilisation of CoMan, which is actuated by series elastic actuators. The main goal of this dissertation is to dynamically stabilise the humanoid robot on the floor by the simplest multivariate feedback control for the purpose of walking. The multivariable scheme is chosen to take into account the joints’ interactions, as well as providing a systematic way of designing the feedback system to improve the bandwidth and tracking performance of CoMan’s existing PID control. A detailed model is derived which includes all the motors and joints state variables and their multibody interactions which are often ignored in the previous studies on bipedal robots in the literature. The derived dynamic model is then used to design multivariable optimal control feedback and observers with a mathematical proof for the relative stability and robustness of the closed loop system in face of model uncertainties and disturbances. In addition, two decentralized optimal feedback design algorithms are presented that explicitly take the compliant dynamics and the multibody interactions into account while providing the mathematical proof for the stability of the overall system. The purpose of the proposed decentralized control methods is to provide a systematic model based PD-PID design to replace the existing PID controllers which are derived by a trial and error process. Moreover, the challenging constrained and compliant motion of the robot in double support is studied where a novel constrained feedback design is proposed which directly takes the compliance dynamics, interactions and the constraints into account to provide a closed loop feedback tracking system that drives the robot inside the constrained subspace. This method of control is particularly interesting since most control methods applied to closed kinematic chains (such as the double support phase) are over complicated for implementation purposes or have an ad-hoc approach to controller design.

In terms of walking trajectory generation, an extension to the ZMP walking trajectory generation is proposed to utilise the CoMan’s upper body to tackle the non-minimum phase behaviour that is faced in trajectory generation. Simple inverted pendulum models of walking are then used to study the maximum feasible walking speed and step size where parameters of CoMan are used to provide numerical upperbounds on the step size and walking speed. Use of straight knee and toe push-off during walking is shown to be beneficial for taking larger step lengths and hence achieving faster walking speeds.

Subsequently, the designed tracking systems are then applied to a dynamic walking simulator which is developed during this PhD project to accurately model the compliant walking behaviour of the CoMan. A walking gait is simulated and visualized to show the effectiveness of the developed walking simulator.

Moreover, the experimental results and challenges faced during the implementation of the designed tracking control systems are discussed where it is shown that
the LQR feedback results in 50% less control effort and tracking errors in comparison with CoMan’s existing independent PID control. This advantage directly affects the feasible walking speed. In addition, a set of standard and repeatable tests for Co-Man are designed to quantify and compare the performance of various control system designs. Finally, the conclusions and future directions are pointed out.
Declaration

No portion of the work referred to in this thesis has been submitted in support of an application for another degree or qualification of this or any other university or other institute of learning.
Copyright Statement

i. The author of this thesis (including any appendices and/or schedules to this thesis) owns any copyright in it (the “Copyright”) and s/he has given The University of Manchester the right to use such Copyright for any administrative, promotional, educational and/or teaching purposes.

ii. Copies of this thesis, either in full or in extracts, may be made only in accordance with the regulations of the John Rylands University Library of Manchester. Details of these regulations may be obtained from the Librarian. This page must form part of any such copies made.

iii. The ownership of any patents, designs, trade marks and any and all other intellectual property rights except for the Copyright (the “Intellectual Property Rights”) and any reproductions of copyright works, for example graphs and tables (“Reproductions”), which may be described in this thesis, may not be owned by the author and may be owned by third parties. Such Intellectual Property Rights and Reproductions cannot and must not be made available for use without the prior written permission of the owner(s) of the relevant Intellectual Property Rights and/or Reproductions.

iv. Further information on the conditions under which disclosure, publication and exploitation of this thesis, the Copyright and any Intellectual Property Rights and/or Reproductions described in it may take place is available from the Head of the School of Electrical and Electronic Engineering.
Acknowledgements

I would like to sincerely thank my supervisor Dr. Martin Brown for providing me the conditions and full support to carry out independent research in the exciting field of humanoid robotics. This thesis would have not been possible without his insightful suggestions and honest and constructive criticism during the entire course of this PhD project. I also thank him for the valuable lessons on scientific writing and presentation skills.

I would like to thank Dr. Gustavo Medrano-Cerda, for numerous inspiring discussions about robotics and control engineering as well as useful and constructive feedback about my research during the course of this PhD. I am grateful for the knowledge he shared with me. It has been a pleasure to work with him.

I wish to express my sincere gratitude to the CICADA management team, Prof. David Broomhead, Prof. Paul Glendinning, Dr. Jonathan Shapiro, Prof. Hong Wang and Mrs. Helen Harper for continuously supporting me by all means to have access to the software and hardware equipment which was required for my research as well as supporting me to carry out several research visits for experiments on iCub and CoMan.

I also would like to thank Prof. Darwin Caldwell and Dr. Nikos Tsagarakis who really supported me in Italy to have access to iCub and recently CoMan to carry out the experiments that I needed for my PhD. The results presented in this doctoral thesis would not have been possible without their support.

I would like to thank the CICADA research associates, Dr. Piotr Kowalczyk, Dr. Margarita Korovina, Dr. Younes Chahlaoui and Dr. Pablo Shmerkin for their friendship and numerous discussions about science, mathematics, philosophy and
life in general. I have learnt so much from them not only about different areas in mathematics such as dynamical systems and numerical analysis, which I did not know much about at the beginning of my PhD, but also improved my skills in some of my favourite sports such as table tennis, badminton and swimming. I wish them good luck and success in their future careers.

This dissertation is dedicated to my family, for their unfailing love, support and invaluable advice throughout my life. I thank my mother for her love and for the sacrifices that she has made in life, so that I can live up to my potential. I thank my brother Dr. Hamed Dallali for his advice and friendship. Last but not the least, I thank my uncles Dariush, Firose, Cyrus and Behrouz for their love and support, especially during the years that I have been away from home.

*Houman Dallali*

November, 2011,

The University of Manchester.
Chapter 1

Introduction

Control of bipedal walking is an interesting area of research in robotics and hybrid systems. A bipedal gait is inherently hybrid due to the periodic sequence of continuous motions followed by discrete transitions. The continuous motions are due to single support and double support phases during walking. The discrete transitions are due to impacts between the foot and the ground when switching between the continuous phases. This PhD project is a part of the CICADA project [1] that studies complex hybrid systems and it involves modelling, controller design and implementation on a compliant humanoid robot called CoMan.

1.1 Statement of the Problem

In this thesis, the stabilisation problem of a compliant humanoid robot during walking is studied. The development of humanoid robots with compliant joints is a recent and novel idea. The main aim of using passive compliance is to improve shock tolerance, energy efficiency, force control and safety of these robots.

It is well known that a humanoid robot when walking has unstable dynamics that resemble an inverted pendulum which has to be stabilized before walking can be achieved. Bipedal walking stability can be divided into two categories. The first category is concerned with the stability of joint trajectory tracking control systems. The second category concerns the stability of walking, which is related to maintaining the
robot’s balance. This is fundamentally dependent on the stability and performance of the joint tracking system. Although both these issues have been widely studied for rigid robots, the idea of using passive elasticity and compliance for walking robots is the topic of current research. Hence, the main focus of this thesis is to design a systematic model based tracking system with a mathematical proof of closed loop stability. Moreover, the second category is also studied to use the robot’s torso to improve the walking robustness.

Most rigid humanoid robots resort to simple, independent PID control without a mathematical proof for the stability of the overall closed-loop system. Hence, the problem of designing an optimal tracking control system which directly takes the compliant dynamics of the robot into account is studied. Two feedback schemes with centralized and decentralized architectures are proposed. The centralized feedback requires high communication bandwidth which is available on the robot and can achieve a high tracking bandwidth. The decentralized feedback can be implemented locally on the DSP controllers but has less tracking bandwidth in comparison to the centralized feedback.

Moreover, the existing simulation packages are often inaccurate in terms of dynamics or are hard to reuse for other robots (for instance some packages are written in Java or C++). In order to carry out the dynamic modelling and simulations in Matlab, Robotran [2] was chosen to generate the equations of motion which can benefit from Matlab’s control design tools. The result of this work is implemented successfully in simulation and on the real robot.

1.2 Aims and Objectives

The main aim of this dissertation is to dynamically stabilise the humanoid robot, CoMan on the floor using model based feedback control designs for the purpose of walking. To this end the key objectives of the project were:

1. To derive the detailed dynamic model and equations of motion including the compliance and actuator dynamics of CoMan;
2. To design the simplest multi-variable feedback controller to improve the tracking bandwidth by taking the interactions into account;

3. To design decentralized PD-PID feedback control for each joint of the robot while taking the links’ interactions into account, to reduce the amount of trial and error in the joint feedback design;

4. To study novel methods for design of controllers for control of the constrained motion of the robot in double support phase;

5. To improve the velocity estimation, given the 12 bit resolution of the encoders on the robot;

6. To use a Zero Moment Point (ZMP) based walking trajectory generator in simulations and experiments to test the effectiveness of the designed feedback controllers;

7. To develop an open source dynamic walking simulation to simulate walking as well as quickly updating the models of the robot to keep the models consistent with the hardware upgrades;

8. And to implement and validate the designed LQR feedback controllers on the 10 DoF of the robot and compare the performance of the tracking system with CoMan’s existing PID controllers.

1.3 Thesis Contributions

A brief summary of the thesis contributions are listed as follows.

1. The first contribution of this project was to derive the dynamic model and equations of motion which include the significant compliance and actuator dynamics of the robot.

2. Linear Optimal feedback control (LQR) and velocity observers were designed and implemented successfully on CoMan. The centralized feedback provided a
CHAPTER 1. INTRODUCTION

performance improvement in terms of tracking bandwidth which directly affects the walking speed.

3. Two novel decentralized algorithms are proposed for tuning the PID gains which provide a mathematical proof of the robot’s closed loop stability and robustness. Both these algorithms are based on minimizing the LQR cost function. The first novel method is based on a sparse gradient descent search method. The second method is formulated as Linear Matrix Inequalities (LMI). A key idea behind the decentralized feedback design is to take the link interactions into account which is lacking in the commonly used independent PID design methods.

4. Deriving a “clean” velocity signal is a common problem in robotics which is directly influenced by the sensor resolution. In order to obtain an improved velocity estimate three different schemes, namely reduced order Luenberger observer, Unknown Input Observer and the traditional numerical differentiation and averaging were studied. All three schemes are implemented on the robot in different experiments.

5. The problem of controlling the robot in double support was considered and a novel constrained feedback scheme was proposed where new physical insights were obtained into the nature of the compliant dynamics of the robot. This problem is interesting due to both the constrained motion of the robot in double support phase (the legs form a closed kinematic chain) and the under actuated degrees of freedom related to the passive compliance in the joints’ actuators.

6. A Zero Moment Point (ZMP) based walking trajectory generation method was proposed to utilize the upper body to improve the walking robustness. Moreover, simple rules and relations were derived to relate the tracking bandwidth, kinematics, torques and ZMP stability condition to walking parameters such as maximum feasible step length and walking speed.

7. Subsequently, a dynamic walking simulation was developed based on Robotran
and Matlab which modelled various phases of walking (single support and double support with a ground contact model) as well as the effect of external and unknown disturbances such as stiction and quantization noise. These models are reusable (based on Matlab) and available for download in [1].

8. The observer based LQR feedback was initially implemented on a compliant joint prototype to validate the modelling and control system design method.

9. In the subsequent experiments, centralized observer and LQR feedback were implemented on the 10 DoF of the robot. Several sway and squat experiments were carried out to validate the control designs. In addition, several sine tracking tests were carried out to partially validate the tracking bandwidth. However, numerous technical challenges were faced during the experiments which were tackled to pave the way for dynamic stabilization of the robot on the ground.

10. Air walking experiments were conducted on CoMan to study the effects of the designed centralized LQR and independent PID feedback control schemes.

1.4 Thesis Outline

A general outline of the contents of each chapter in this thesis is provided in this section.

In chapter 2, a review of the state of the art in bipedal robots and the open problems in terms of their control system design is presented, which is linked to enhancing their walking speed, robustness and energy efficiency. The role of passive compliance in walking is also briefly summarized. Finally, a brief survey of dynamic walking simulation software is given.

In chapter 3, the modelling details of the compliant humanoid robot in single support and double support are introduced. Moreover, compliance and actuator dynamics are integrated into the mechanical model. The contact model of the walking robot with the ground is discussed. Finally, the overall hybrid model of walking which consists of single support and double support phases is presented (contribution 1).
In chapter 4, the lower level joint tracking system is studied in detail. Four types of joint control systems for compliant robots are studied. Initially, the limitations related to independent PID joint control scheme are pointed out. Then the centralized LQR feedback control for compliant robots is proposed (contribution 2). The centralized architecture is then modified to design decentralized feedback control schemes with overall proof of the closed loop stability (contribution 3). Finally, the theory behind velocity estimation using linear observers is briefly summarized (contribution 4).

In chapter 5, the single support model of the robot is used to propose an elaborate constrained feedback control to control the robot in double support phase while catering for the uncontrollable modes of the dynamic motion (contribution 5). This method has the advantage of using the controllable single support model without using the common reduced order models.

In chapter 6, firstly, an online trajectory generation method based on preview control of the extended cart table model is proposed that utilizes the upper body for improving walking robustness. Secondly, new results on the selection of bipedal walking parameters, for a humanoid robot, such as the tracking bandwidth, walking speed and step length are presented (contribution 6).

In chapter 7, a dynamic simulator is developed for CoMan to simulate the generated walking trajectories. The dynamic simulator includes the multibody model of the bipedal robot, actuator dynamics, compliance, sensor noise and the ground contact model (contribution 7).

In chapter 8, the results of experiments on CoMan are discussed. These experiments consist of: a test on a single flexible joint unit (contribution 8), implementation of the observer based LQR, several tests to investigate the effect of the observer (contribution 9) and several tests to increase the bandwidth of the overall tracking system. In addition, the results of implementing a sway, squat and air walking are presented, where it is shown that the LQR controller provides 50% less control efforts and tracking errors in comparison with the PID (contribution 10).

Finally, in chapter 9, the work presented in this thesis is summarized and the
future directions are pointed out.

1.5 List of Publications

The major part of the contributions of this PhD thesis are published or submitted to well-known national and international journals and conferences. Two journal papers associated with this thesis are accepted for publication with the details provided in section 1.5.1. Five conference papers are also written with four papers already published with the details provided in section 1.5.2.

1.5.1 Journals

The journal papers are:


1.5.2 Conferences

The conference papers are listed in chronological order as:


2. H. Dallali, M. Brown and G. Medrano-Cerda, Toward Dynamic Walking with a Compliant Humanoid Robot, in Proc. of European Nonlinear Oscillations
Conferences (ENOC), Rome, Italy, 24-29 July, 2011.


Chapter 2

Background Review

The study of walking bipedal robots is one of the most important and challenging areas in robotics research. The control of bipedal walking involves significant challenges due to the high Degrees of Freedom (DoF), the presence of under actuated DoF, the hybrid nature of the system switching back and forth between single support and double support and external disturbances such as ground impacts. The design of a control system for humanoid locomotion with such complex dynamics has attracted much attention in the literature [3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14]. In this chapter, an overview of the state of the art in bipedal walking research and the key concepts and ideas behind the walking control and simulation are presented.

This chapter is organised as follows. In section 2.1, the most advanced humanoid robots built around the world are reviewed and their mechatronic characteristics as well as their unique research achievements are pointed out. Section 2.2 concerns the walking control system for humanoid robots which leads to the main dynamic walking stability criteria discussed in section 2.3. In addition, the latest improvements in humanoid walking is discussed in section 2.4. The role of compliance in walking is explained in section 2.5. Finally, the existing simulation software for dynamic walking simulation is reviewed in section 2.6.
2.1 Humanoid Robots

A humanoid robot is an anthropomorphic and complex electromechanical machine which is controlled by a computer. There is a wide research area concerning the humanoid robotics which consists of mechatronic design, software and hardware architectures, dynamic stability, locomotion, learning approaches, perception, human-humanoid interaction, planning and cognition and numerous applications of daily life tasks that are highly non-trivial for humanoid robots.

Humanoid robots are being designed to operate in close proximity to people and to be able to adapt to human environments. In addition to these challenges, a major shortcoming of the currently designed humanoids is a lack of reliability in their mobility. There are two issues associated with their mobility. The first is that they are expected to have a high degree of robustness in their locomotion to be able to navigate in crowded environments without falling. The second issue is related to safety of the people who are in contact with the robot. Numerous research groups in the world are studying scientific approaches to tackle these two obstacles to pave the way for these robots to help the aging population in terms of health care, welfare and many potential future applications. The latest and the most common humanoid robots are briefly reviewed in the next sections.

2.1.1 ASIMO

ASIMO (Advanced Step in Innovative MObility) [3] is the first humanoid robot to demonstrate remarkable dexterity and agility in walking with a maximum walking speed of 2 km/h. This robot was developed during an intensive research program funded by the Honda company which started in 1986 with development of E0 and subsequently in 1996 more advanced versions such as P2 [15] and a year after P3 [16] were unveiled. The robot designs evolved over 26 years leading to ASIMO with 57 DoF and 48 kg weight as shown in Fig. 2.1. ASIMO stands 1.3 m tall. In addition to walking (at the speed of 2.7 km/h), ASIMO can demonstrate grasping, navigation inside a home environment, climbing stairs, jumping while changing direction and
even running at the speed of 9 km/h (5.6 mph), which makes ASIMO the fastest running robot in the world.

However, the mechatronic design used in ASIMO [18] is based on rigid joints which are actuated by powerful DC motors and controlled by high gain independent PID controllers.

2.1.2 Toyota Partner Robot

The second fastest bipedal running robot is the Toyota Partner Robot [4] with a running speed of 7 km/h (4 mph). This robot uses the foot placement method for running and it can recover from a push using this foot placement method which adds robustness to its gait. The Toyota running robot which was first unveiled in 2009, weighs 56 kg and stands 1.52 m tall.
2.1.3 Petman and Atlas

The fastest walking robot is Petman by Boston Dynamics with a human like walking speed of 4.4 \textit{mph} \cite{20}. The aim of the Petman project is to study the feasibility of chemical testing using a fully articulated robotic mannequin for the US military \cite{21}. The development of this robot was started with a $26.3$ million military grant. Petman weighs 80 $kg$ and stands about 1.75 $m$ tall and has hydraulic actuators which in addition to high power output, provide a degree of compliance to absorb the ground impacts. There is not much information regarding the control system of the biped but it is evident from the videos that Petman is also using an algorithm based on foot placement since it can also recover from a push by placing its foot on the right location. Currently the robot can walk, go up the stairs, squat and kneel. Recently, Boston Dynamics has received more funding from DARPA (Defense Advanced Research Projects Agency in USA) to develop a more advanced version of Petman which is called Atlas.

2.1.4 Humanoid Robotics Project

The Humanoid Robotics Project (HRP) is an initiative by the Ministry of Economy, Trade and Industry in Japan with a budget of $40$ million which was launched in 1998. The main goal of the project was the development of general domestic service humanoid robots. This project is led by the National Institute of Advanced Industrial Science and Technology (AIST) and Kawada Industries in Japan. An impressive series of humanoid robots has been built as shown in Fig. 2.2. In phase 1 of HRP, the R&D department of Honda Ltd. developed the first prototype, HRP-1 \cite{22}. The legs and arms of HRP-1 could only be controlled separately which was not suitable for further development of humanoid applications.

HRP-2 \cite{5, 23} is the main outcome of this project with 30 DoF, 58 $kg$ weight and 154 $cm$ height. HRP-2 was the first finished prototype in the HRP series to demonstrate walking on uneven surfaces, getting up from a fallen position and to interact with humans and help in domestic service tasks. This robot uses ZMP criteria.
for walking with a maximum speed of 2 km/h. Subsequently, HRP-3 was developed based on the HRP-2 experience with additional dust and splash-proof protection, enhanced hand coordination, improved cooling systems, and prolonged operation time (2 hrs compared to HRP-2s 1 hr battery operation). Due to added degrees of freedom to the hand and arms, HRP-3 has a total of 42 DoF. In terms of control system both HRP-2 and HRP-3 have a distributed control system which is implemented via CAN (Controller Area Network - bandwidth= 1 Mbps). In particular, the prototype of HRP-3 [24] was developed using a distributed control system and a real-time Ethernet communication network. Advantages and design criteria such as the size of the data packets for motor commands and sensor signals that affect the choice of communication network and required bandwidth (about 10 Mbps) for real-time control were discussed in [24].

HRP-4 has a new slim, lighter and athletic mechanical design (weight of 39 kg, height of 151 cm) to improve its functioning in the domestic environment. In addition to the mechanical upgrades, this robot has several key improvements which include LAN (bandwidth = 10/100 Mbps) and Wireless LAN (IEEE802.11a/b/g) networks for internal and external communications which is an important design decision to avoid communication bottle neck. Moreover, the robot’s operating system is based on Linux open source and real time software which is an important feature for sharing, reusing, maintaining and improving the software by a group or a larger research community.

Finally, HRP-4c [25] (also referred to as the cybernetic humanoid robot) is the most recently developed humanoid robot by AIST in the HRP series with a female appearance which is 158 cm tall and weighs 43 kg. HRP-4c has human like facial expressions and by recently performing an impressive singing and dancing act with a group of professional dancers at the Digital Content Expo 2010 in Japan, it moved one step closer toward coexisting with humans. It should be noted that all the HRP series robots have rigid joints which are controlled by local PID controllers and an online ZMP based trajectory generation.
2.1.5 **WABIAN-2R and KOBIAN**

Waseda University in Japan has been involved in humanoid robotics research since 1970. Some of the early developments include the Waseda Legs (WL) [28, 29] which were a pair of legs with 12 DoF used as a platform to study the walking problem. Two of their latest humanoid robots are WABIAN-2R [6] with 41 DoF (stiff joints), 150 cm in height and 64 kg in weight, and KOBIAN, similar to WABIAN-2R but with facial expressions (as shown in Fig. 2.3). WABIAN was the first humanoid robot to demonstrate human-like walking with stretched knees [30], where an improvement in the walking energy efficiency was reported. Moreover, this humanoid has passive toe joints and has demonstrated human like walking with heel contact and toe push off. The aim of these projects at Waseda University is to develop a human motion simulator and also a human partner in daily tasks. The walking control system of WABIAN-2R is based on local independent PID control with online ZMP trajectory generation.
Figure 2.3: WABIAN-2R (left) and KOBIAN (right) built at Waseda University in Japan [31], courtesy of Prof. Takanishi's biped humanoid robot group.

2.1.6 HUBO

Korean Advanced Institute of Science and Technology (KAIST) [32] has been developing a series of walking robots since 2001 which consist of KHR-1, KHR-2 and KHR-3 where the mechatronic design was evolved over three versions. Hubo (KHR-3) [7] is the latest humanoid robot developed at KAIST in 2004 which has 41 DoF, stands 125 cm tall and weighs 55 kg as shown in Fig. 2.4. The joints and frames used in the design of Hubo were made with high stiffness. The aim of the Hubo project is to provide a reliable platform for implementing dynamic walking, navigation and image processing algorithms [7]. In 2008, Hubo was integrated with an android head resembling Albert Einstein. The head used RC servo motors for facial expressions and the robot was called Albert Hubo [33] (shown in Fig. 2.4).

In late 2009, a running experiment was reported on Hubo [34] with the maximum speed of 3.24 km/h. The tracking control system which used in Hubo humanoid robot is based on independent PD position feedback loops running at 1 kHz which is the traditional approach for trajectory tracking in robotics.

In addition to the work on humanoid robots, a human-riding biped chair, Hubo FX-1 [35] has been developed at KAIST which has much larger size and it can carry a human passenger (as shown in Fig. 2.4).
2.1.7 JOHNNIE and LOLA

The Institute of Applied Mechanics at the Technical University of Munich (TUM) in Germany developed an anthropomorphic robots called JOHNNIE [8] with the main aim of fundamentally studying and realizing a fast walking machine with a human-like gait. JOHNNIE has a total of 23 DoF, stands 180 cm tall and weighs 40 kg. Two control systems based on feedback linearization and traditional independent joint PID control were implemented on the robot. It was reported that although the feedback linearization method works correctly in theory, it did not result in an optimal solution for JOHNNIE, as it is very computationally expensive, requires an accurate model of the dynamics and very accurate sensors with a high bandwidth. The implemented control strategy was then chosen to be the industrial independent PID joint control, operating at high sampling rate [36]. Although, JOHNNIE demonstrated ZMP based walking with the maximum speed of 2.4 km/h, it could not reach higher jogging speeds. Therefore, based on the experience of developing JOHNNIE, a new lightweight version called LOLA [9] was developed in 2006 (as shown in Fig. 2.5). The goal of the LOLA project is to achieve fast bipedal walking using powerful stiff actuators and a lightweight structure. The control system of LOLA is reported to be based on decentralized joint control. LOLA has 22 actuators and a SERCOS-based communication system which is able to realize an accurately timed, high speed serial interface with the speed of 16 Mbps required for realtime closed loop control (which
is more efficient than 10Mbps LAN network). In 2010, LOLA was unveiled in Hannover Messe (the world’s biggest industrial fair) where it demonstrated walking and navigation capabilities [37].

2.1.8 Computational Brain

In a collaboration between the Department of Brain Robot Interface (BRI) at Brain Information Communication Research Laboratory Group in Japan and SARCOS Research Corporation in USA, a 50 DoF humanoid robot called the Computational Brain (CB) was developed to study the behaviour of the human brain in real world situations [39]. CB stands 157 cm tall and weighs 92 kg. Fig. 2.6 shows a picture of CB. In addition to an on-board PC104 CPU stack with an Intel 1.4 GHz Pentium-M processor (which controls all the 50 DoF at 1 kHz via sensory feedback), the robot has a 100 Mbps Ethernet and wireless communications allowing it to be controlled by an external PC cluster. The balance controller for the robot implements a 3D force-based control to resemble the human balance system.

In 2007, an impressive full body compliant control was implemented on CB which balanced the robot in the presence of unknown external pushes [41]. In another experiment [42] a stepping controller was added to CB’s control system which helped the robot to take a step in face of external unknown disturbances.

An earlier version of the CB humanoid robot, built by SARCOS, is based at
Figure 2.6: Computational Brain (CB) developed by JST-ICORP Computational Brain Project and Sarcos company [40]. Courtesy of Prof. Jun Morimoto.

the Robotics Institute at Carnegie Mellon University (CMU) in Pittsburgh, USA. Research on this robot at CMU is mainly focused on push recovery startegies using simple inverted pendulum type models and force control strategies [43].

2.1.9 Rabbit and Mable

In a collaboration between several French research laboratories and the University of Michigan, a bipedal robot called Rabbit was developed in 1998 to achieve planar high speed walking and running gaits [10]. The main aim of this project was to study the feedback-control of bipedal locomotion. A distinct feature of this robot was its point feet which essentially made the control of walking harder since the ZMP criterion could not easily be used. In addition to achieving planar walking speed of 1 m/sec, studies on Rabbit led to development of the exciting theory of Hybrid Zero Dynamics (HZD) with application to control of bipedal locomotion [44]. The mathematical analysis done on Rabbit, not only provided a complete proof for the closed loop stability, but also reduced the debugging and development time that is often present in robotic experiments.

Although the robot could achieve stable walking at different speeds, once it tried running the gait was not stable. This was mainly down to the large energy loss during impacts with the ground. At high speed running, the impacts are much larger and
rigid actuators often fail to maintain the kinetic energy of the overall biped. The lessons learnt from Rabbit were later used in development of a compliant version of Rabbit which is called Mable [11]. The mechanical design of Mable was carefully chosen to resemble the dynamics of the SLIP (Spring Loaded Inverted Pendulum) model [45]. A detailed identification was done to derive the dynamic parameters of the robot [46] and, Moreover, the theory of HZD was extended to include the additional compliance dynamics. Currently Mable is able to walk at the speed of 1.5 \( m/sec \) with a stable gait. Fig. 2.7 shows a picture of Rabbit and Mable.

2.1.10 LUCY

LUCY [12] is a bipedal robot built at Vrije University of Brussel in Belgium which unlike other bipedal robots uses pleated pneumatic artificial muscles. The torque and the compliance of these muscles are controllable. The aim of this bipedal robot was to explore the role of compliance in walking and running as these muscles have a high power to weight ratio and their compliant behaviour helps in absorbing impact shocks and storing and releasing of energy.

LUCY’s joint tracking control system was based on inverse dynamics with an addition of PID control, and the walking trajectories were generated based on ZMP.
Due to the slow dynamics of the valves which controlled the muscles, the robot was able to walk at the maximum speed of 0.15 m/sec. Fig. 2.8 shows a picture of LUCY on a treadmill.

2.1.11 iCub

iCub is a humanoid robot that was developed in five years by the RobotCub [49] consortium to provide the cognitive science research community with an open source platform to study cognition. In other words, iCub is used as an open source robotic platform to study how a human child learns the basic motor skills as well as learning and recognizing different objects. iCub is a full humanoid robot with a head, arms, hands, waist, and legs that are actuated by 53 motors (as shown in Fig. 2.9). It is the size of a 3.5 year old child with a compact, modular, mechatronic architecture, stands 104 cm tall, and weighs less than 23 kg. The robot is equipped with two cameras, microphones, an inertial sensor, 12 bit Hall effect position sensors (encoders) in all the actuated joints. Its internal communication network is based on CAN, which connects the local DSP joint controllers to the central PC104 computer (in the robot’s head). An ethernet communication is used to connect the PC104 to an external PC cluster.

It should be noted that iCub was initially planned to be designed as a 2 year old
child and later it was decided to develop the robot with the size of a 3.5 year old child, mainly due to the mechatronic and actuation limitations. Hence, crawling was considered in the mechatronic design while the feasibility of walking was raised later in the project.

Therefore, iCub uses a cable drive system for the ankles that results in lowering the joint stiffness which leads to several problems for walking. One of these problems was dislocation of cables during standing. Another problem was the time varying delays (about 10 ms) in communication between the central PC and the robot, due to the combined effects of the CAN and the LAN network. A presentation at the humanoid robotics symposium at the IET discusses these limitations [50]. A comprehensive review of issues regarding control system design under limited data rate is provided by [51].

In the early stages of this PhD project several experiments were conducted on iCub to implement centralized observer based LQR feedback control. Although these tests resulted in stable motion while the robot was held up on the frame, as shown in Fig. 2.9, the robot could not be stabilized while standing on the ground. As mentioned earlier, the key problems were the communication network delays (the CAN network bottleneck), and the mechanical design of the legs (which were not robust enough to enable the robot to walk since on several occasions the cable system which connected
the motor to the ankle joint broke). These issues were considered in development of the CoMan humanoid robot as described next. Further details about the technical challenges faced are provided in chapter 8.

2.1.12 CoMan

CoMan (COmpliant huMANoid) is being developed at the department of advanced robotics at the Italian Institute of Technology (IIT) based on the experience of developing the iCub but with the aim of exploring the role of compliance and natural dynamics in fast walking, and more explosive tasks such as jumping and running. Fig. 2.10 shows the pair of legs that has been developed and used on this PhD project to implement dynamic stabilizing controllers based on the theory of optimal control. CoMan has the size of a 4-5 year old child with a revised mechatronic design that makes it robust for legged locomotion. The 10/100 Mbps Ethernet communication between the robot and the central PC also provides the required bandwidth for implementation of advanced control theories on this robotic platform.

2.1.13 Summary and Comparison

There are four main directions of research being pursued in the field of walking robots. The first area aims at designing an optimal tracking control system to stabilize the naturally unstable dynamics of a bipedal robot. This area of research has been somewhat neglected by opting for the simplest PID control solution which provides adequate performance but ignores the joint interactions which become more significant at high speeds of motion. In addition to independent PID control some research groups have used a combination of feedback linearization and PID. This line of research has been studied in depth in this PhD project to design and study centralized and decentralized control schemes.

Research on bipedal walking is pursued in the three other major areas that seek to enhance the walking robustness, speed and energy efficiency. Among these, robustness to uneven terrain and external disturbances is the most ambiguous research
Figure 2.10: CoMan legs, courtesy of the Department of Advanced Robotics at Istituto Italiano di Tecnologia.

question as no rigorous measure of robustness exists to provide an answer to the question, “how stable (or robust) a bipedal robot is?” Therefore, comparison of walking stability margins and walking robustness between robots is limited to the common conservative measures such as ZMP, or the basin of attraction of a limit cycle walker. In fact, several attempts for formalising and quantifying the bipedal walking stability margin have been made [52, 53, 54], but all efforts have only provided a partial answer to this question. Therefore, choosing the most stable, or the most robust bipedal robot remains controversial.

On the other hand, well defined measures for walking speed and energy efficiency exist. Walking speed can be simply defined as the distance travelled in a unit of time. In order to provide a better comparison, the square root of the Froude number is used - that is \( \frac{\sqrt{v}}{\sqrt{gL}} \), where \( v \) is the peak speed, \( g \) is the gravity constant, and \( L \) is the leg length. Most bipedal robots such as Mable have a dimensionless velocity of 0.48. The fastest bipedal walking robot is Petman [20] with maximum peak velocity
Walking energy efficiency is measured in terms of Cost of Transport (CoT). CoT is defined as energy consumed per unit weight per unit distance travelled. CoT for an average human is about 0.2. The record for the most energy efficient bipedal robot is currently held by Cornell Biped called Ranger [55] that has a CoT of 0.28. In May 2011, this robot travelled 40.5 miles non-stop while consuming a total power of 16 watts including the actuators and the electronics with an average speed of $0.59 \text{ m/sec}$. A prize of $200K$ has been assigned for any robot that can provide a combination of speed, energy efficiency, robustness and dexterity [56]. Also, a dedicated meeting (Dynamic Walking [57]) is being held annually which brings together the researchers from bio-mechanical, mathematics and control engineering.

In the next section, a more detailed review of the control system for bipedal robots is presented.

### 2.2 Bipedal Walking Control System

The control system architecture implemented on current humanoid robots is quite complex. This is due to the various tasks that they perform. In this project the focus is on walking control systems with a hierarchical architecture [58]. The inner layer (which is latency critical) must run at a fast sample rate (about 1 kHz) and be responsible for trajectory tracking (which is studied extensively in chapter 4). The outer layer runs at a slower sample rate and generates dynamically stable and feasible walking trajectories for the tracking system (discussed in detail in section 2.3 and an extended method is proposed in chapter 6).

The inner layer can have two types of control architectures: The distributed or decentralized versus the centralized scheme as shown in Fig. 2.11. Although the distributed and decentralized terms are often used with ambiguity in the literature [63], a distinction is made between the two in this thesis. Distributed control refers to design of controllers for each subsystem (e.g. a robot joint) using the local sensor information independently of other subsystems (e.g. other DoF of a robot) which are
treated as unknown disturbances. Decentralized control refers to the design of controllers for each subsystem using local sensor information, including the interactions with other subsystems in the control design, often by using optimisation.

![Diagram of control architectures](image)

Figure 2.11: The centralized LQR (solid line) and decentralized PID (dashed line) feedback architectures. The time derivatives are denoted by a dot and the estimates are denoted by a hat sign.

In section 2.2.1, a general overview of the existing tracking methods is presented. Subsequently, in sections 2.2.2 and 2.2.3, the inner layer control architectures - namely distributed, decentralized and centralized - are reviewed. It should be noted that the focus of this thesis is on position trajectory tracking, since the majority of the existing bipedal walking robots are controlled in position mode. However, other control concepts such as force/torque control, impedance control and hybrid position/impedance control can be implemented in centralized and distributed architectures.

### 2.2.1 Overview of Tracking Methods

The tracking control problem is to determine the joint input for a desired trajectory in face of disturbances, such as stiction, modelling errors and external forces. Depending on the model used for the controller design, the control inputs maybe joint torques or actuator voltages. In this thesis, the control inputs are actuator voltages that are sent to the local DSP boards and the measurements are joints’ positions. At
this point in time, the torque sensors have just been installed on CoMan and this is the reason that the proposed controller designs are based on position sensors rather than torque sensors. In addition, the controller commands are formulated as motor voltages rather than torques because CoMan electronics are designed to control motor voltages.

There are many control approaches that can be applied for the joint tracking control problem. The choice of control method has a significant impact on the robot’s performance in terms of bandwidth, robustness to disturbances and uncertainty, and possibly energy efficiency. Single Input-Single Output (SISO) PID control is the most popular and widely used method in bipedal robots due to its simplicity. The most advanced humanoid robot ASIMO [3, 18] uses a distributed control design with traditional high gain joint PID controllers. The recently developed Korean humanoid HUBO2 also uses a distributed PD based joint motion controller [34]. The design strategy for this robot was to design the joints to be as stiff as possible [7]. The latest version of the HRP series robots, HRP-4c also implements distributed SISO PD control for joint tracking [25].

The planar Spring Flamingo robot used simple intuitive control ideas for walking (VMC as discussed in section 2.3) and also SISO PID control at the joint level [110]. M2V2, a much more advanced force controlled humanoid with series elastic actuation also uses independent and decoupled PID controllers [111].

The planar bipedal robot Rabbit [10] used independent joint level PD control for its rigid joints with 12 Hz bandwidth to realize a limit cycle walking trajectory. It was reported that a set of PD controllers could be found that result in stable operation for double support, single support and flight phases, but the design process for the derivation of the PD gains was not mentioned. SISO PID gains are often derived by trial and error and this process gets more complicated for compliant robots when several sensors are providing measurements from different states of a compliant joint (namely the motor and link positions and velocities).

Inaccurate reference tracking was reported for the planar compliant bipedal robot Mable when high gain PD control was used. A solution was proposed to incorporate
a feed-forward term, that was realized by a polynomial fit to the nominal torque profile plus conventional PD control which resulted in a more accurate tracking [11].

Another control strategies is feedback linearization and inverse dynamics that consist of an inner loop that in ideal case (when the dynamic parameters are known exactly) linearizes the nonlinear system and an outer loop that implements the traditional control designs. The pneumatic biped Lucy [48] used a feed-forward inverse dynamics method plus a PID feedback loop to improve the tracking performance. In most cases there is a considerable uncertainty about the robot’s parameters and this becomes more important as the humanoid picks up unknown loads that will result in deterioration of the guaranteed performance [59].

Dealing with uncertainties naturally lead to the adaptive and robust control methods. Robustness in this context refers to robustness of the tracking controller in terms of stability and performance to parameter uncertainty that is often described as a worst case scenario and adaptive tracking controller refers to a controller that uses an online parameter estimation algorithm to tune the controller parameters.

These methods can be used to guarantee robust stability and a worst case performance in the face of parametric uncertainty, disturbances and modelling errors. Robust controllers such as $\text{H}_\infty$ are fixed parameter methods, while adaptive control methods utilize a parameter estimation algorithm to tune their parameters in face of uncertainty [60, 61]. In the case of a repetitive task such as steady state walking, the robust control tracking error is expected to be also repetitive while adaptive control aims at improving the error over time. However, the main drawbacks of adaptive control methods are poor transient response and sensitivity to disturbances, hence research is being done to improve these drawbacks [62].

The next two sections review the tracking control systems from the architectural point of view.
2.2.2 Distributed and Decentralized Control

Most inner layer control architectures fall into the distributed control category. Many famous bipedal robots such as ASIMO, HRP-2, HRP-4c, iCub and JOHNNIE implement a distributed and independent joint control scheme. In the case of recently developed compliant robots, such a scheme was proposed in [64, 65] where the realtime joint control was assigned to each joint independently of the other joints’ movements (Fig. 2.12 shows their prototype leg which was used to test the distributed control scheme).

Jena Walker II as shown in Fig. 2.13, is the latest compliant planar bipedal robot developed in the locomotion laboratory (Lauflabor) at University of Jena, which uses independent PD joint control [66].

However, decentralized control has not been applied in practice to high DoF bipedal robots. This is mainly due to the simplicity of the traditional independent PID control design (the distributed architecture) and the challenging theory of optimal decentralized control (which is known to be a complex non-convex optimisation problem) [67, 68, 69]. In addition to optimal decentralized control, decentralized pole-placement methods have also been studied in the literature [70, 71]. A survey of the literature supports the fact that the obtained results on decentralized control
There is a gap in the literature to study the application of decentralized control strategies on bipedal robots and in particular compliant walking. This topic is studied in detail in sections 4.3 and 4.4 in chapter 4 where new novel results such as a sparse gradient descent scheme and an LMI based feedback design algorithm are obtained. In addition, part of this research, which compares the decentralized and centralized LQR based control schemes, is presented in [73].

2.2.3 Centralized Control

Centralized modern control theory has been applied to some of the early rigid bipedal robots with low DoF (mainly planar) [74, 75, 76]. The simplest and computationally most affordable centralized control scheme is LQR optimal control which is used in these studies.

Following the rapid achievements in computational power, recently developed HRP-3 [24] and HRP-4 humanoid robots are designed with powerful on-board centralized computers with a high bandwidth hardware to enable researchers to implement both centralized and decentralized control schemes.
Among those recently developed, JOHNNIE [36] was used to explore two centralized and distributed control schemes. The centralized scheme (which was implemented on the robot) was based on feedback linearization and the computed torque method. The distributed control method was based on the traditional independent joint control. In [36] the importance of high bandwidth to realize centralized control is pointed out.

The key benefits that could improve bipedal walking machines using the centralized control scheme are: increased bandwidth, enhanced disturbance rejection and, more importantly, the inclusion of the joint interactions in the control system design, which assists with the natural dynamics of the walking robot instead of fighting against the interacting dynamics. Moreover, centralized control benefits from advanced mathematical proofs of stability and performance of the overall closed loop system, an area which is often neglected in the distributed control literature.

Hence, there is a gap in the literature to study and implement centralized feedback control on high DoF compliant bipedal robots. The centralized LQR optimal control scheme is studied in detail in section 4.2 and the experimental results on CoMan are reported in chapter 8.

2.3 Dynamic Walking Stability

Numerous trajectory generation methods have been proposed in the literature. This section reviews the main dynamic walking control approaches that are used in bipedal robots. The stability measure is the crux of walking trajectory generation. Having a precise and meaningful stability margin is the most crucial requirement for locomotion, as it can be utilized to predict the bipedal walking relative stability and methods of avoiding a fall [52].

2.3.1 Zero Moment Point

In bipedal walking, the overall indicator of stability is the point where the influence of all acting forces on the robot can be replaced by a single force. This point is termed
as the Zero Moment Point (ZMP). In other words, ZMP is the point on the ground where the net moment generated from the ground reaction force is zero [77]. In order to ensure the biped’s stability, the ZMP must always reside in the convex hull of the all contact points on the ground. If ZMP lies on the edge of the support polygon the trajectories may not be dynamically feasible and the robot may fall. ZMP always coincides with Centre of Pressure (CoP) in a dynamically balanced gait [77].

ZMP is the most widely used criterion in humanoid robots to obtain joint trajectories [78, 79, 80, 81, 82]. A combination of preview control and ZMP is applied to a commercial humanoid robot [83]. In [84] multi-objective optimisation was utilized for ZMP trajectory generation. In [85] a stochastic hardware in the loop optimisation was carried out on a small sized robots. An offline trajectory generation method which considered the constraints, and the relationship between the CoM and ZMP is proposed in [86]. Therefore, ZMP is the most popular method among humanoid robots.

However, the drawbacks of using ZMP are energy inefficient walking, little disturbance rejection ability and limited walking speed. The conservative condition of having the robot foot flat on the ground imposes constraints on the maximum achievable speed. It can be argued that ZMP requires full local controllability which is not a necessary condition for stable walking as it can be seen in limit cycle walkers [87]. In addition, ZMP provides no solution when the robot’s balance is lost. Further details about ZMP trajectory generation are given in chapter 6 in section 6.1.

2.3.2 Foot Rotation Indicator and Angular Momentum

A Foot Rotation Indicator (FRI) is introduced as the point on the foot/ground contact surface where the net ground-reaction force would have to act to keep the foot stationary [53]. In order to ensure the foot will remain stationary, the FRI must remain within the support polygon. It is worth pointing out that the FRI method is only applicable only during the single support phase and the rotation of the stance foot is the main focus [90]. This dynamic approach aims at providing a prediction of
pure foot rotation, which is an important criterion for gait’s stability. When the foot is maintained on the ground, FRI and ZMP and CoP are all the same point. If the foot starts rotating then ZMP and CoP are on the edge of the support polygon while FRI is outside this polygon [53]. Therefore, FRI provides more information than ZMP or CoP and the distance between the support polygon and the FRI point can quantify the gait’s instability (in terms of both magnitude and direction). However, there is a caveat as a recent study reported that, in practice, the distance between the FRI and ZMP is within the measurement accuracy (0.1 % of the foot length) [88].

The FRI point is related to the rotational stability of the bipedal robot as the rate of angular momentum appears explicitly in the FRI formulation [53]. In fact, during standing, walking and running, humans tend to maintain, or correct, their balance by appropriate changes in their angular momentum, in addition to taking steps. Change of angular momentum by itself is neither a necessary nor a sufficient condition for stable walking, since one can move one’s arms violently while walking stably (though it is obviously not energy efficient). Since angular momentum can be used to recover the balance when pushed, it is a good reserve for rejecting disturbances. Moreover, this gyroscopic effect has been used for humanoid walking control [89, 90].

### 2.3.3 Limit Cycle Walking

The emphasis in the limit cycle approach is on the periodicity of the walking gait. In implementation of the limit cycle approach, the ZMP criterion is violated, thereby proving its conservativeness. The limit cycle method involves an extensive search to find energy efficient and periodic gaits using optimisation tools, and hence it is not suitable for online adaptation of the trajectories. Moreover, the method suffers from small basin of attraction and sensitivity to uneven terrain.

Limit cycle walkers have made an impressive progress in dynamic and energy efficient walking. The most recent progress on limit cycle walkers was achieved in Delft university [87, 91, 92]. Cyclic stability of a limit cycle walker is analysed by obtaining the eigenvalues of Poincaré return map. The following linear relationship
describes a typical Poincaré return map.

\[ X_{n+1} = KX_n \]

If the eigenvalues of the matrix \( K \) are less than one, small perturbations will decay and hence the state \( X_n \) will converge to a steady state periodic trajectory. The major drawback of this method is that it is limited to analysis of small perturbations. This disadvantage has recently been partially addressed with the introduction of active lateral foot placement methods [93]. Therefore, abrupt change of speed or large perturbations (when the robot is pushed) cannot be analyzed by this criterion. In addition, this stability criterion relies on periodicity which is not a necessary condition for bipedal walking. In summary, this method does not provide a complete solution to achieve high walking speed, robustness and disturbance rejection.

### 2.3.4 Central Pattern Generators

Central Pattern Generators (CPG) are believed to be a more biologically representative way of walking [94, 95]. A CPG is a network of coupled oscillators (a set of coupled differential equations) which can produce a stable limit cycle. These coupled oscillators can be tuned to produce rhythmic signals with desired amplitude, frequency and phase difference. In the case of bipedal walking, a CPG is tuned (often offline) to generate the joints’ trajectories, corresponding to stable walking. The joints’ trajectories can be generated by assigning an oscillator to each joint or by assigning an oscillator to a particular point on the biped’s body (such as the feet or the hip) where inverse kinematics is then required to obtain the joints’ trajectories.

Trajectory generation based on CPG is often implemented in open loop without feedback. As a result, provided the nominal gait is stable, small external disturbances are tolerated and in principle the biped can achieve a degree of robustness. The stability of CPG based walking is analysed using the Poincaré tools from dynamical systems as described earlier. Fig. 2.14 illustrates a couple of neural oscillators that can be tuned to produce a stable limit cycle or a fixed point. The input to these oscillators can be robot’s posture or velocity information and the output is often
Figure 2.14: A coupled neural oscillators (left) and a typical stable limit cycle (right).

the leg position in lateral and sagittal directions. The joints’ positions can then be computed via inverse kinematics [96].

A review of the literature suggest that, although this method has been initially successful in simulation and in several cases implemented on planar robots and 3D toy-sized robots (namely, QRIO [97] and HOAP), successful implementation on larger 3D humanoid robots requires further research. In 2004, a planar biped (similar to a compass gait) demonstrated planar stable walking using neural oscillators [98]. A year later, this work was extended and realized successfully in 3D on QRIO robot [96]. In the same year, a toy-sized humanoid robot called HOAP was used to demonstrate stable walking on varying slopes (2.35° – 8°) and at varying speeds (2.9 - 11.4 cm/sec) [99]. In 2006, a planar robot called RunBot, with four actuated DoF (Hip and knee in each leg) used a set of neural oscillators to achieve a speed of 0.8 m/sec (equivalent to speed of 3.5 leg lengths per second with leg length of 0.23 m) [100]. In the same year, a programmable CPG was applied to a simulation of HOAP humanoid robot [101]. The parameters of the CPG were used to program limit cycle type trajectories to modify the gait. Implementation on the real robot was left for future work. In 2008, CPG was applied to iCub for drumming and crawling which showed an improvement in terms of programming CPGs for several tasks. In 2010, it was reported that a CPG was combined with supervised learning and applied to CB humanoid robot. This
experiment was motivated by the general belief in neuroscience that the motor control task is achieved by distributed CPGs, creating a stable limit cycle. Although CPG could be used for balancing (in single and double support) and quasi-static walking, dynamic walking or stepping could not be achieved [102]. The joints’ tracking errors and delays were reported as the cause of this problem. However, a combination of CPG and a balance controller based on regulating the overall Centre of Mass (CoM) resulted in a marching motion. Also, a combination of CPG, balance control and speed modulation led to a more dynamic walking gait.

Apart from the mentioned practical challenges, their main drawback is the difficulty in applying a particular CPG to arbitrary periodic motions, as required when performing various tasks. In most cases hand tuning or optimisation is involved. Several attempts have been made to integrate learning algorithms with the CPG method, to adapt the trajectories for different tasks (as discussed next).

### 2.3.5 Learning Approaches

An attractive research area is to use machine learning approaches to achieve stable walking gaits. In most robotic experiments a precise model of the robot is necessary to carefully compute the walking trajectories. However, learning approaches are used to overcome the modelling errors which are motivated by people’s ability to learn by demonstration or repetition.

In [103], an imitation learning approach was integrated with a CPG method to adapt the frequency of trajectories for a planar five link robot. Moreover, reinforcement learning [104, 105] and Iterative Learning Control (ILC) [106] are among the most popular methods proposed for walking. In [107], policy gradient reinforcement learning and particle swarm optimisation were applied to a toy sized humanoid robot to optimise the walking speed. In [108], iCub combined the CPG method with learning for drumming, where microphones were used for feedback. However, in the case of walking, ensuring the safety and stability of a humanoid robot is one of the drawbacks in directly applying learning methods to a real robot.
2.3.6 Virtual Model Control

Virtual Model Control (VMC) is an intuitive motion control framework which is based on a transformation from a high level task specification to low level motion control [109]. This control method relies on assuming fictitious spring-dampers attached to different points on the biped which are then translated to the required joint torques via kinematic relationships (ie. Jacobian). Fig. 2.15 illustrates this idea where the relation between the force and torques is

\[ \tau = J f \]  

(2.1)

where \( J \) is the Jacobian. Equation (2.1) can be derived from the robot’s kinematics and the principle of virtual work. There are two interesting points about Equation (2.1). Firstly, no dynamic model of the robot is required and secondly, it converts a cartesian quantity \( f \) to a joint-space quantity \( \tau \), without computing any inverse kinematics. This intuitive method was first applied to the torque controlled Spring Flamingo [110] and recently a new humanoid robot called M2V2 [111] is also being controlled with the same principle.

2.3.7 Capture Point

This point of view approaches walking from a different perspective. It focuses on recovering the balance when the robot is perturbed as opposed to the ZMP and other methods, which focuses on maintaining balance throughout the walking gait.
The capture point criterion is based on the claim that the most difficult subtask of bipedal walking is to regulate the CoM velocity. This is difficult because the DoF that contribute to the velocity are under-actuated in the continuous phase of walking. Once the CoM has travelled outside the support polygon a discrete transition is required to regulate the speed with which the biped must take a step. It is this hybrid nature of the problem that makes walking interesting and challenging, since during the majority of a human-like walking gait, the only way to prevent a fall is to take a step.

A capture point is a point that a bipedal robot must step into to come to a complete stop. The angular momentum of the upper body and arms can be used to extend the capture point to a capture region. The size of the capture region depends on how fast the swinging leg can reach a capture point. The faster the swinging leg can move, the larger the region of capture. In addition, one can define a one-step capture point, a two-step capture point, or an N step capture point. If the angular momentum form the torso is used, the capture point becomes a capture region.

Based on the N-step capture point, a stability margin is defined and simple formulas can be obtained to estimate the capture point of a bipedal robot [52, 112, 113]. However, it is worth noting that impacts are not included for capture point calculations. The calculations are based on the Linear Inverted Pendulum (LIP) model, which provides a conservative stability measure. The capture point for inverted pendulum walking is

$$r_c = v \sqrt{\left(\frac{h_0}{g}\right)} \sqrt{\frac{v^2}{4g^2}}$$

and the capture point for LIP walking is

$$r_c = v \sqrt{\left(\frac{h_0}{g}\right)}.$$

In summary, the capture region provides an area on the ground whose distance from the support polygon serves as a meaningful stability margin.
2.3.8 Foot Placement

A similar approach known as the foot placement estimator is based on the claim that the key to a dynamically balanced gait is foot placement [114]. This method seeks to restore the balance of a perturbed bipedal robot by calculating where the foot needs to be placed to restore the balance. Therefore, this is a strategy on how to recover from a disturbance and how to come to a complete stop. This method has been extended to 3D and it includes impacts in its calculations. However, it is based on a rocking type motion of a two link rigid robot which is quite limited and assumes little control of the robot during the rocking motion.

2.3.9 Stochastic Meta-Stability

A novel stochastic measure called meta-stability was proposed in [115] to quantify the robustness of a compass gait walker over uneven terrain. It was argued that walking on uneven terrain can be modelled as a stochastic process where the slope of the ground can be considered as a random variable. Subsequently, a discrete (step to step) stochastic state transition matrix was analysed in order to quantify the mean time to failure (the number of steps before the biped falls). An iterative learning algorithm was then used to maximize the largest eigenvalue of the state transition matrix (which corresponds to the expected time to failure). However, this method suffers from the curse of dimensionality. A presentation with further details about the eigenvalue analysis of meta-stability is available in [116].

2.3.10 Summary

To sum up, ZMP (or CoP), FRI are important ground reference points which are used for dynamic generation of walking trajectories in closed loop. The angular momentum appears in both measures which plays an important role in stabilizing the robot. A more comprehensive review of these methods is available in [88]. Most existing trajectory generation methods apply the ZMP method to simple models (LIP) to be able to quickly modify the walking trajectories, on the fly.
Furthermore, limit cycle and CPG methods rely on periodicity of walking which is not a necessary condition. Their stability analysis is local (based on linearized maps). As these approaches are implemented in open loop (without feedback), the walking trajectories can not be modified online in response to external disturbances. However, an active area of research aims at combining effective online learning with these methods to provide the capability of online modification and tuning of the trajectories.

The remaining approaches (VMC, capture point or foot placement) are mainly applied to simulations or planar robots. Therefore, practical applications of these methods on a real robot would require further research.

## 2.4 Human Like Walking

In this section, two criticisms about humanoid walking are briefly discussed. The first criticism is that humanoid robots walk with flat feet, instead of using the heel contact and toe push-off. The reason behind walking with a flat foot, and the recent robots which have overcome this issue are mentioned (discussed in section 2.4.1). The second criticism is that humanoid robots walk with bent knees which is discussed in section 2.4.2.

### 2.4.1 Non-Flat Foot (Heel-Toe) Walking

One of the features that is apparent in most humanoid walking is that these robots constantly keep their foot sole on the ground during walking, while in natural human walking, one uses heel contact to begin the double support phase and uses the toe push-off to leave the double support and swing the trailing leg forward.

The reason for humanoid robots walking with flat feet is related to ZMP stability which all humanoid robots, so far use for walking. Therefore, until recently their walking gait was only realized by keeping a large support area on the ground to ensure stability.

In addition to a natural and graceful gait, there is an energy advantage in non-flat
foot walking. Several recent attempts have been made, including WABIAN-2R built at Waseda University, which successfully demonstrated that the ZMP trajectories could still be used while the robot uses the heel contact and toe push-off [117]. A more robust bipedal robot which recently demonstrated heel to toe walking was PETMAN [20]. This robot responds actively to lateral pushes while walking with heel to toe transitions.

In the next section, the bent knee walking problem, which is related to a common problem in computing the inverse kinematics, is studied and several solutions are compared.

### 2.4.2 Walking with Straight Knees

The bent knee problem is one of the criticism made about bipedal walking. The conventional methods often impose over-conservative limitations on the biped which result in the walking gait looking unnatural. The main reason for bipedal robots walking with bent knees is a singularity that arises in inverse kinematic computations [118]. A singularity often occurs at the boundaries of a robotic workspace where a DoF is effectively lost (fully stretched) and other DoFs have to move faster to compensate for this problem. This is a well-known problem in robotic manipulators, which can be expressed mathematically using kinematic relations. According to forward kinematics the relation between the cartesian end effector and joint angles is

\[ x = f(q). \]  \hfill (2.2)

Equation (2.2) is nonlinear and the problem of finding the joint angles \( q \) for a given cartesian coordinate \( x \) is called the inverse kinematics. The inverse kinematics can be computed by linearizing Equation (2.2) about \( q_0 \) as

\[ x = f(q_0) + J(q_0)\delta q \]  \hfill (2.3)

where \( J \) is the Jacobian matrix. The Newton-Raphson iteration can then be applied to find \( q \). The Jacobian \( J \) also appears in velocity kinematics if Equation (2.2) is differentiated as

\[ \dot{x} = J(q)\dot{q}. \]  \hfill (2.4)
However, bipedal robots with 6 DoF in each leg have a singularity associated with the straight knee configuration that causes numerical problems in the inverse kinematics. In other words, when $J$ is singular the angular rate in Equation (2.5) becomes extremely large and creates problems in solving the inverse kinematics.

\[
\dot{q} = \frac{1}{\det(J)} (\text{adj } J) \dot{x}
\]

(2.5)

This problem has been widely studied in robotics, particularly for manipulators, in the past two decades. Several methods are reviewed in [119] such as damped least squares methods and robust damped least squares methods as a solution for singularity avoidance. Also weighted least-norm method, is mentioned for joint limit avoidance. The problem of singularity and avoiding joint limits in solving the inverse kinematics is also considered by the gradient projection method [120] and the extended Jacobian method [121].

On the other hand, several approaches are also proposed for humanoid walking with straight knees. Levenberg-Marquardt [122] was proposed as an alternative to widely used Newton-Raphson iterations. In [128] an improvement in the bent knee walking was reported by modifying the walking trajectory but the problem was not completely resolved. Another solution for Wabian humanoid was proposed in [117] that utilizes a hip motion to avoid the singularity [123]. More recently, Hubo [124] proposed another solution. Moreover, a singularity consistent approach was proposed in [125] to treat the singularity in the inverse kinematics by putting an upper bound on the Jacobian inverse in the neighbourhood of the singularity. In [126] the walking trajectory was modified to avoid a singular configuration while obtaining nearly straight knee walking. An extra heel joint was added to the robot in [127] to add more redundancy and treat the singularity. Although various solutions exist for the bent knee walking problem, CoMan walks with bent knees and there is no recipe to deal with singularities in the trajectory generation code. Therefore, several solutions can be adapted for CoMan as mentioned above.
2.5 Advantages of Compliance in Walking

Human locomotion is powered by muscles which are connected via tendons to the skeletal system. Both the muscle and the tendon have elasticity as reported in the literature [129, 130]. Three uses for springs in animal locomotion were studied in [131]. The first benefit of springs is that the animal or robot can save energy. The energy saving use of springs is more significant in running. Although there is evidence from human walking data that swing phase is performed passively. In fact, electro-myographic data from human walking shows very little activity in the leg muscles once the leg is lifted. This suggests that leg swinging is largely done passively similar to a pendulum [132]. The second benefit of springs in walking and running is to reduce the impact forces between the feet and the ground which are moderated by compliant ankles and foot pads. The third use of springs is to improve the grip on the road by adapting the foot passively to the uneven surfaces as well as reducing chatter while the feet impact with the ground.

In addition to safety and energy efficiency advantages which can result from using springs and elasticity in the actuators, the newly developed humanoid robots such as CoMan aim at realizing a more human like gait due to the passivity and natural dynamics which are inherent in the overall multi body system.

2.5.1 The Need for Compliant Robots

The traditional belief in robotics was to make the joints of a robotic system as stiff as possible in order to allow for precise and high bandwidth position tracking. Stiff joints with high gear reduction ratios result in a hardly back-drivable joint which can easily transfer the mechanical shocks. However, in light of the recent insights into human walking and the emergent fields in robotics (such as human-robot interaction) as well as safety considerations, the interest in the study and development of soft, flexible joints in robots has increased considerably [11, 12, 14, 91, 133, 134, 135].

Moreover, several experiments conducted on bipedal robots showed the limitations of rigid joints in terms of energy efficiency as reported by the Rabbit project.
[10]. The problem faced in Rabbit was related to actuator saturation once the attempt to make a transition from walking to running failed. This idea played a key role in development of Mable [11] which is the compliant version of rabbit and it is designed to achieve planar bipedal running by using compliance in its actuation system. Therefore, in the rigid robots the compliance is introduced in software (as high gain PID controllers) and in flexible joint robots the compliance is introduced passively using elastic elements.

2.5.2 Control of Compliant Robots

While individual compliant actuators have been widely studied [136, 137], there is a gap in the literature for systematic and linear multivariable control designs when these actuators are employed in a multi degree of freedom robot. Multivariable controllers account for the dynamic interactions in a system instead of treating the motion of other joints as unknown disturbances. A multivariable controller can facilitate the joints motion instead of fighting the dynamic interactions.

Several control methods based on feedback linearization [138], the theory of integral manifold [139] and singular perturbation [140] are proposed for control of flexible joint robots which are complex from an implementation point of view and require a good knowledge of the robot’s dynamic parameters to achieve a good performance in practice. In theory, a multivariable computed torque control considers the links interactions through the nonlinear equations of motion and cancels out the nonlinearities and decouples the dynamics at the cost of computing the nonlinear dynamics at every sample time. Traditional PID control is then applied to the resulting decoupled closed loop system.

On the contrary, in [141], a simple PD controller with gravity compensation was proved to stabilize flexible joint robots about a reference position. The PD feedback controller was introduced on the motor’s position and velocity. However, the position of the link was shown to be sensitive to uncertainties in gravitational and elastic parameters, since the link position was controlled indirectly via the motor position
without direct feedback or integral action on the link position. In [142], a full state feedback controller was proposed which utilized both the motor and link states to achieve a better performance compared to simple PD controller introduced in [141] and still keep the controller simple for implementation. The implementation problems of the complex control schemes were pointed out as the motivation for the full state feedback method. However, the full state feedback design was based on independent joint control and the dynamic links’ interactions were not considered. This drawback is often addressed via linear multivariable control approaches.

Linear multivariable control was applied to a robot manipulator in [143] and a comparison was made to the computed torque control in terms of simplicity and low computational burden of the linear multivariable controller. Although, multivariable control has been applied to manipulators, it has been ignored in bipedal robots to simplify the control system. The bipedal walking robot M2V2 that uses controllable series elastic actuators, has a decoupled and independent joint control system [111].

In practice, linear control systems have demonstrated a good trade off between simplicity and acceptable tracking performance. As mentioned earlier, the feedback linearization method was not suitable for control of JOHNNIE, and the independent PID joint control provided better results [36]. A similar problem was encountered in the LUCY project, where joint tracking control system required PID control, on top of the nonlinear computed torque method. In [46], it was reported that use of PD control for the planar bipedal robot Mable caused inaccurate reference tracking.

Therefore, the problem is how to gradually improve the tracking control system by systematically tuning the PD-PID decentralized feedback gains while directly taking the joint’s elasticity and multibody interactions into account. The centralized feedback architecture is also studied as a viable solution to this problem. One of the contributions of this study is to design and implement linear multivariable controllers for compliant walking robots, using the detailed models derived in this chapter 3.
2.6 Overview of Walking Simulation Software

There are numerous general multi-body simulation packages available commercially and academically for general robotic simulation. A survey of the available software was done in the initial stages of the project. After considering the mathematical and physical accuracy, friendly user interface and the cost of each software, Robotran [144] was chosen. An important contribution of this project is development of the most suitable simulation tool for bipedal control system design and simulation in Robotran and Matlab. The advantages and disadvantages of the most common software packages are presented in this section. A further survey of the other software packages is given in [145].

2.6.1 Open Architecture Humanoid Robotics Platform

Open Architecture Humanoid Robotics Platform (OpenHRP) [146] is a simulator and motion control library for humanoid robots developed by NIAIST and General Robotix Inc. It is a dynamic simulator since the forward dynamics of the robot is computed. It also has a user friendly GUI and it is customized to send the result of the simulations directly to PC clusters connected to HRP robot series. The reflexive controller and trajectory design for this simulation is discussed in [147]. Moreover, several improvements on the accuracy of the ground contact simulation for walking were reported in [148].

However, it does not allow for modifications to include various actuator and compliance models and to customize the software to another humanoid robot such as CoMan which must necessarily model the significant behaviour of its compliant actuator dynamics. In addition, the documentation available on this software is very limited. Therefore, other modelling and simulation solutions for walking with CoMan were sought.
2.6.2 Open Dynamic Engine

Open Dynamic Engine (ODE) [149] is an open source library created initially by Russell Smith for simulating rigid body dynamics and it is currently maintained by a software community. ODE has been used for simulating vehicles, objects in virtual reality environments, particularly in computer games. ODE in robotics simulations has been mainly used for simulating the kinematics of motion where the geometry is computed accurately but the dynamics are not computed accurately.

There are several commercial and open source robotic simulation environments which are based on ODE for the physics calculations such as [150, 151, 152]. The iCub simulator [152] was tested at the beginning of this PhD project, which did not satisfy all the requirements that are needed in creating a realistic walking simulator since the important dynamic parameters such as the CoM or the inertia were not included in the simulator and also Jacobian and linearization information had to be computed separately. However, the iCub simulator is perfectly suitable for simulating the kinematics of the robot at speed levels where the accelerations are not significant.

2.6.3 Webots

Webots [150] is a commercial robotic simulation tool developed by Cyberbotics to program and simulate mobile robots which at its heart relies on the ODE (Open Dynamics Engine) [149] to perform the physics simulation. As mentioned in section 2.6.2, this is adequate for stable mobile robots but it lacks the accurate multibody and contact dynamics needed for unstable and complex walking robots. Moreover, important mathematical analysis tools ranging from Jacobian at different points on the multibody system to explicit access to inverse and direct dynamic equations and linearization tools about an operating point are not available in the software. Finally, the source code of this simulation software is not open and hence complete control over the numerical integration features and other numeric elements can not be accessed by the users.
2.6.4 Adams

Adams [153] multibody dynamics software is a commercial professional software which incorporates precise physics by simultaneously solving equations for kinematics, statics, quasi-statics and dynamics. This software has an integrated numerical analysis and finite element analysis tools. An interesting case study on simulation of human walking done by Jet Propulsion Laboratory and UCLA in USA using Adams is available in [154]. However, the license fee for this software is quite expensive due to the high demand for it in various industrial research which could create difficulty in sharing the developed models researchers who do not have access to this software.

2.6.5 Robotran

One choice among the software packages for bipedal robots is Robotran [2, 144] which is used in this project mainly due to its accurate dynamic modelling of multibody systems. It does not rely on inaccurate physics engines such as ODE and it provides the user with the symbolic differential equations of the motion which makes the simulation a bit slow but it has proven to work well in experiments on CoMan as explained in chapters 7 and 8. Another advantage of using symbolic models is that the user can easily change the numerical value of CoMan parameters without having to change the equations of the robot. In addition, this package provides total freedom for the user for addition of actuator models, introduction of mechanical or user-defined constraints, external forces on any point on the CoMan, as well as addition of sensors to acquire position, linear and angular velocity and acceleration of any point of interest on CoMan. For instance, it is desired to attach a sensor to CoM of each link to observe the overall CoM in cartesian coordinates while the control system is performing the tracking control on the model. Robotran works as a toolbox under Matlab that makes all the toolboxes in Matlab available to the user (i.e. the control toolbox, optimisation toolbox for trajectory generations or any other user defined function in Matlab). Robotran has been made available for free by the developers at Centre for Research in Mechatronics (CEREM) in Catholic University of Louvain in
Belgium for the research and academic purposes.

However, Robotran has certain limitations including little documentation on the software. The visualization for walking as well as adding other features that were not included in Robotran are done separately in Matlab as discussed in detail in chapter 7.

2.6.6 MapleSim

In the later stages of this PhD project, a recently developed multibody and multidomain (including the electrical, hydraulic and pneumatic physics) simulation tool called MapleSim was investigated which has the potential to improve the walking simulations developed in Robotran in a more easy to use and user-friendly environment. A case study is being developed in MapleSim to improve the walking simulation features for CoMan [155].

MapleSim has a symbolic modelling engine which can be used for deriving the dynamic equations of the robot including the actuator and compliant dynamics, and also to generate optimised stand alone C code for real-time implementation. Actuators can be selected from various libraries within the software as well as allowing the user to integrate equation based custom components. The multibody equations can then be manipulated within Maple to derive inverse kinematics and inverse dynamics relations. The graphical user interface provides easy access to the robot parameters which reduces the possibility of errors. In terms of analysis tools, several built-in templates are available within MapleSim for linearization, Jacobian relations, sensitivity analysis, optimisation and control design. This tool has already been used in development of WABIAN-2 [156].

2.7 Conclusions

To conclude, a survey of the most advance humanoid robots was presented and their control system architectures were discussed. An overview of the iCub and CoMan humanoid robots was given to familiarize the reader with the aims of these two
The key areas in dynamic walking, including dynamic walking stability criteria and the tracking control systems were reviewed. The need for introducing compliant elements in the actuators as well as their impact on the control system were discussed. Three advantages of compliant walking were introduced to be shock tolerance, energy storage and increased robustness. Two gaps in the literature were identified, with the first being the need for improved control systems and the second being the need for enhancing the walking robustness. These two objectives are being realized by the recently developed compliant humanoid robots (in terms of hardware) and the implementation of advanced centralized and decentralized control solutions (in terms of software).

In addition, an overview of the existing walking simulation software was given and the advantages and disadvantages of each package were pointed out. In this PhD project, a customized walking simulation was developed using Robotran and Matlab which is discussed in detail in chapter 7. The major part of this is carried out on CoMan. A detailed model of the compliant humanoid robot, CoMan is presented in chapter 3. The developed model is then used for centralized and decentralized control system designs in chapter 4 as well as simulations.
Chapter 3

Modelling CoMan

This chapter derives a detailed model of the CoMan for simulation and control system design. During a full walking cycle the bipedal robot switches between single support (swing) phase when one foot is in contact with the ground, and double support phase when both feet are in contact with the ground. These phases of walking have qualitatively different dynamics in the sense that the former is an open kinematic chain and the latter is a closed kinematic chain.

An overview of the novel humanoid robot CoMan is provided in the next section. The hybrid model of walking is then presented which is followed by the derivation of single support phase, double support phase and impact models. The actuator and compliance dynamics are then coupled to the single and double support models. Finally, all these phases are combined in a unified hybrid model. It is shown that both phases of walking can be combined if the ground reaction forces are activated or deactivated in the model based on the state dependent events such as heel strike or toe off. It should be noted that the derivation of the state space models are provided in Appendix B and detailed implementation of the phases of walking and the ground models in Robotran and Matlab software packages are discussed in chapter 7.
3.1 Overview of CoMan

The CoMan compliant humanoid [14] is being developed at the Italian Institute of Technology (IIT) as a derivative of the original iCub, which has passive compliance in the major joints of the legs. As part of the AMARSI European project [157], the use of passive compliance will provide shock protection, robust locomotion, safer interaction and potentially energy efficient locomotion.

3.1.1 Mechanical Overview

CoMan’s lower body has 17 DoF with 15 DoF being actuated and 2 DoF being passive compliant joints that form the toes on each foot. The 15 actuated DoF consist of 6 DoF for each leg and 3 DoF for the torso. In addition, CoMan uses brushless DC motors and harmonic drives. The frameless motors are chosen from the RBE series manufactured by Kollmorgen. These motors use rare earth magnets (that are the strongest permanent magnets) placed on the rotor to reduce the rotor inertia and increase the acceleration. Apart from the ankle lateral joints that use RBE713 brushless motors, the rest of the joints use RBE1211 motors and all joints use CSD17 harmonic drives. The main parameters of these actuators are provided in Appendix C. The sagittal joints are equipped with physical elasticity in series with the output of the gearbox to improve the robot’s shock tolerance and robustness during locomotion. Each joint is equipped with three position sensors (2 absolute and 1 relative) and one torque sensor. The ankle, knee and hip DoF of each leg in sagittal plane are actuated by compliant joints that are compact series elastic actuators [158]. A picture of the legs is shown in Fig. 3.1 and Fig. 3.2 shows the compact soft actuator and the ankle joint.

3.1.2 Electronics and Data Acquisition

Each joint has a DSP controller that processes the signals from the sensors and also implements an Ethernet communication with 10 (Mbps) between the joints. The robot has high communication bandwidth that can be used for real-time centralized
Figure 3.1: (a) The compliant CoMan legs. (b) The corresponding revolute joints, R, P & Y denote Roll, Pitch and Yaw rotations, respectively.

Figure 3.2: (a) The compact series elastic actuator. (b) The sagittal ankle joint and the passive DoF at the toe.
and decentralized control architectures. Without this high speed network, adopting multivariable and centralized control architectures would not be possible due to large and varying delays (more than 10 ms) that are present in iCub that uses a CAN bus (Control Area Network) with 1 (Mbps) speed for internal communication.

The size of the data packets for each motor commands and sensor signals can provide an estimated required bandwidth which determines the communication network for real-time control. These issues have been considered for CoMan at the design stage. In addition to the on-board controllers, the central computer for CoMan is an Intel Pentium Xeon 3.33 GHz dual processor with 12 CPUs and 24 GB RAM running a 64 bit Windows professional operating system that is connected to the Ethernet network. CoMan uses 12 bit encoders for measuring the motor and link positions. The encoders placed at the output of the motors have the advantage of the gear-box reduction ration, providing a higher resolution. Currently custom build torque sensors are being integrated at each joint to provide feedback for torque control.

The next section, provides the big picture for the hybrid model of walking. Subsequently, the modelling details of each phase of walking are discussed.

3.2 Hybrid Model of Walking

In human like walking, single support, double support, heel strike impact and toe push-off phases are present and this is independent of the model representations. Fig. 3.3 illustrates a full cycle of human walking. In this section, this hybrid nature of walking is described. A full cycle of walking can be described by the hybrid diagram shown in Fig. 3.4. The robot starts from the swing (single support) phase $Q_1$ and begins the double support phase following the leading leg’s heel impact (event 1). While in double support the robot moves forward ($Q_2$ phase) that is followed by the leading leg’s toe impact and the robot enters $Q_3$ where both feet are flat on the ground. This is followed by event 3 when the trailing heel leaves the ground and the robot continues to move forward with the leading foot flat on the ground and trailing toe pushing the robot forward ($Q_4$ phase). Once the toe push-off is complete
the trailing leg leaves the ground (event 4) and then next swing phase starts. A summary of the continuous phases and discrete events is provided in Table 3.1. In the walking simulation, discrete events are defined to be state dependent and once they are detected the ground contacts on the supporting legs are activated to model the ground reaction force and the impacts.

In order to fully understand the dynamics of walking the next sections provide a detailed model of the single support phase (open kinematic chain) and the double support phase (closed kinematic chain). Moreover, these models are then used to design joint tracking controllers in chapters 4 and 5.
Table 3.1: Phases and events of the walking hybrid model.

<table>
<thead>
<tr>
<th>Phase</th>
<th>Description</th>
<th>Event</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q1</td>
<td>Swing</td>
<td>1</td>
<td>Heel (leading leg) impact</td>
</tr>
<tr>
<td>Q2</td>
<td>Push-off 1</td>
<td>2</td>
<td>Toe (leading leg) impact</td>
</tr>
<tr>
<td>Q3</td>
<td>Push-off 2</td>
<td>3</td>
<td>Heel (trailing leg) leaves the ground</td>
</tr>
<tr>
<td>Q4</td>
<td>Push-off 3</td>
<td>4</td>
<td>Toe (trailing leg) leaves the ground</td>
</tr>
</tbody>
</table>

It is worth mentioning that in this project the single support and double support phases are studied and the heel strike and toe push-off phases of walking will be considered in future work. For the purpose of walking simulation all different phases of walking are implemented in Robotran and Matlab using the ground contact model discussed in the previous section. The ground reaction forces are applied externally to the robot’s feet whenever an impact event is detected and feet are at the ground level. The external forces are removed from the body as soon as the foot leaves the ground (toe-off event). The torques to the joints are provided by the dynamic model of the actuators given in (3.9). In chapter 7, it is shown how the hybrid model and actuator dynamics are integrated to Robotran simulation environments.

3.3 Single Support Model

The single support phase is when the robot has one of its feet on the ground and the other foot is swinging forward. The single support phase of walking can be modelled as an open chain tree structure multi-body system which can be linearized and used for controller design. The joint angles in CoMan are measured as relative angles with respect to the vertical $Z$ axis. Fig. 3.5 shows the robot’s Pitch and Roll DoF and their angle conventions. The Newton Euler or Lagrange formulations can be used to derive the equations of motion. In this thesis, due to the large number of DoF and to speed up the development process and reduce errors, the Robotran multi-body software [144] developed by the Centre for Research in Mechatronics at the Catholic University of Louvain in Belgium, is used to generate the equations symbolically in Matlab. The single support equation is
Figure 3.5: CoMan angle conventions. a) Pitch DoF in $Z - X$ plane with positive rotations about $Y$ axis in the direction of the arrow. b) Roll DoF in $Z - Y$ plane with negative rotations about $X$ axis in the direction of the arrow.

\[ M(q)\ddot{q} + c(q, \dot{q}) = \tau \]  

(3.1)

where $q$, $\dot{q}$, $\ddot{q}$ are vectors of $n$ joints angles, velocities and accelerations, respectively, $M(q)$ is the positive definite and symmetric mass inertia matrix and $c(q, \dot{q})$ is the combined vector of Coriolis, centripetal and gravitational forces and $\tau$ is the generalized torque vector applied to the actuated joints. It should be noted that several models of multibody systems, including planar double pendulum, were derived by hand and compared to the models generated by Robotran to ensure the correctness of the generated models for CoMan, as discussed in Appendix A. Further details about this is provided in the Walking Simulation chapter 7.

### 3.3.1 Linearized Model

The unconstrained Equation (3.1) (single support model) is linearized around the operating point (CoMan’s upright position that corresponds to zero angles and zero angular velocities) as

\[ M\ddot{q} + C\dot{q} + Gq = \tau \]  

(3.2)
where $M$ is the linearized mass inertia matrix, $C$ is the linearized damping matrix and $G$ is the linearized stiffness matrix that represents the gravitational force on the joints. $C$ and $G$ are obtained by linearizing $c(q, \dot{q})$. $C$ is a zero matrix if the joints’ velocities for the linearization point are zero, as is the case in this thesis. From a control design point of view, this equation has full controllability and can be represented in state space form for the purpose of feedback design (discussed in section 3.5).

3.4 Double Support Model

CoMan in double support is subject to a geometric mechanical constraint which represent three nonlinear relationships when both feet are on the ground in $XYZ$ directions.

On a flat ground, the feet must satisfy three additional linear constraints that result from the feet being parallel to the ground. The three constraints can be derived by setting the three sums of the Pitch, Roll and Yaw DoF to zero. In total, the robot on a flat ground is subjected to 6 geometric constraints represented in vector format as $h(q) = 0$. The feet separation is already included in $h$. As a result, six degrees of freedom in the closed chain will become dependent on the other joints. In the case of a robot standing on a flat ground, as shown in Fig. 3.6, the nonlinear constraints can be derived by computing the cartesian coordinate of one foot and setting the height on the $Z$ axis to zero and the feet separation on the $X, Y$ axis to the appropriate value.

In general, a constrained multi-body system is described as a differential algebraic equation [144]:

\[
M(q)\ddot{q} + c(q, \dot{q}) = \tau + J_c(q)^T \lambda \\
h(q) = 0, \quad \dot{h}(q) = J_c(q)\dot{q} = 0, \quad \ddot{h}(q) = J_c(q)\ddot{q} + \dot{J}_c(q)\dot{q} = 0
\]

where $h(q)$ is a vector valued function with three elements that represents the feet separation in $XYZ$ cartesian coordinates and the feet configuration being parallel to
Figure 3.6: Model of the legs in double support with feet separation denoted by \( d \). The area enclosed in the dashed line forms a closed kinematic chain that has a constrained motion. The 3 DoF in the waist are not part of the closed kinematic chain and can have independent motion.

the ground, \( J_c(q) \) is the Jacobian of the constraint and \( \lambda \) is the constraint forces-torques (lagrange multipliers). In the double support phase, when \( \lambda \) is non-zero the robot states are confined within a constrained subspace and when \( \lambda \) is zero Equation (3.3) represents the single support model (unconstrained case). Equation (3.3) can be represented as an ordinary differential equation by deriving the constraint forces-torques \( \lambda \). The constraint forces-torque vector \( \lambda \) is derived explicitly as

\[
\lambda = (J_c M^{-1} J_c^T)^{-1} (J_c M^{-1} c(q, \dot{q}) - J_c M^{-1} \tau - \dot{J}_c \dot{q}) \tag{3.4}
\]

substituting the lagrange multiplier vector into the Equation (3.3) gives

\[
M(q)\ddot{q} + P c(q, \dot{q}) + J_c^T (J_c M^{-1} J_c^T)^{-1} \dot{J}_c \dot{q} = P \tau \tag{3.5}
\]

where \( P = I_n - J_c^T (J_c M^{-1} J_c^T)^{-1} J_c M^{-1} \) is a non-orthogonal projection. Equation (3.5) can be used for the nonlinear representation of the double support model in the form of an ordinary differential equation.

### 3.4.1 Linearized Model

Not surprisingly, the linearization of Equation (3.5) has uncontrollable modes that corresponds to the geometric constraints in double support phase. The linearization
of the double support model can not be used for controller design due to lack of controllability but this linear double support model will be used in the simulation section to illustrate the double support controller’s numerical results,

\[ M_L \ddot{q} + P_L C \dot{q} + P_L G q = P_L \tau \]  \hspace{1cm} (3.6)

where, \( P_L \) is obtained by linearizing \( P \) about the upright posture (zero joint angles and velocities). Further modelling details are given in [160].

In the following section, the compliant actuator dynamics is coupled with the mechanical equations of motion derived in (3.2) to design state-space based control systems in chapter 4.

### 3.5 Compliance and Actuator Dynamics

CoMan uses series elastic actuators in order to try to improve the locomotion energy efficiency, absorb shocks from the ground impacts and also interact with the environment via better force control at the contact points. As a result of this, the drive train has a reduced stiffness and the motor inertia’s motion becomes independent of the joint motion. This adds more state variables to the state space model. In addition, the reflected motor inertias are of the same order of magnitude as the inertia of CoMan’s links. Hence, the actuator dynamics have a significant effect on the overall robot motion. In this section, the electrical and mechanical equations of the actuators are derived and their corresponding state space equations are presented. It is shown that the electrical dynamics are fast and the actuator dynamics are dominated by the drive train stiffness dynamics. Further details about the effects of actuator dynamics on the robot’s motion control system is provided in [59].

#### 3.5.1 Electrical Dynamics

The principle idea in DC motor dynamics is that a current carrying conductor (stator in this case) creates a magnetic field that can exert an electromagnetic torque on the rotor. This torque is proportional to the electrical current that is expressed as
\[ \tau_m = K_t I \]

where \( K_t \) is the motor’s torque constant. Hence, the problem of deriving the motor torque \( \tau_m \), reduces to deriving the current \( I \) in motor the electrical circuit as shown in Fig. 3.7. The current dynamics can be derived by using the Kirchhoff’s voltage law as

\[
L \frac{dI}{dt} + RI + K_b \dot{q}_m = u(t) \tag{3.7}
\]

where \( V_b = K_b \dot{q}_m \) is the back EMF voltage that is proportional to the motor’s angular velocity. Often the current dynamics are considered to be fast and the inductance \( L \approx 0 \). For instance, the Kollmorgen RBE-01211 motor used in CoMan has \( L = 8e - 4 \) Henry and \( R = 1.75 \) Ohms. Therefore, the time constant in Equation (3.7) is \( \frac{L}{R} = 4.5714 \times 10^{-4} \) that in comparison to the motor mechanical time constant (0.69496), derived in section 3.5.2, is negligible. Moreover, the sampling time used in the simulations and experiments are often 1 ms or 2 ms that in comparison to the dominant time constant of the system provides a good resolution on the signal. The sampling at 2 ms proved to be more regular in the experiments. The fast dynamics will die out in the first sampling time (1 ms is about two time constants of the fast dynamics) and hence they do not impose any limitation on our sampling rate. The only limitation on the upper-bound of sampling rate is the quantization noise that will affect the estimated velocity signal. This is further discussed in section 4.5.

Hence, the motor torque reduces to

\[
\tau_m = K_t R^{-1} u(t) - K_t R^{-1} K_b \dot{q}_m \tag{3.8}
\]
The term due to the back EMF constant introduces considerable damping into the system and improves the stability of the joint motion. The motor torque derived in Equation (3.8) is used in the next section to provide the motor input to the drive train and eventually the joint motion.

### 3.5.2 Drive Train Dynamics

In this section, the issue of joint flexibility due to the harmonic drive stiffness and the series elastic element is discussed. A diagram of the compliant joint connected to a load through the series elastic element is shown in Fig. 3.8. A matrix array of the actuators is considered with the correct joint order in the following equations.

![Diagram of a compliant joint driving a load inertia.](image)

Figure 3.8: Mechanical diagram of a compliant joint driving a load inertia. \( N \) represents the gearbox reduction ratio.

In order to couple Equation (3.2) with the actuator dynamics, the joint torque \( \tau \) must be replaced by two coupling terms that indirectly apply torque to the joint. In other words, each joint is driven by the torque produced from the difference in motor and joint angular positions and velocities as

\[
M\ddot{q} + C\dot{q} + Gq = \tau_L
\]

\[
J\ddot{q}_m + B_m\dot{q}_m + \tau_L = \tau_m
\]

where the coupling torque is

\[
\tau_L = B_s(q_m - \dot{q}) + K_s(q_m - q)
\]

and \( J = \text{diag}\{N^2J_{mi}\}, B_m = \text{diag}\{N^2d_{mi}\}, B_s = \text{diag}\{d_{Li}\}, K_s = \text{diag}\{k_{si}\}, \tau_m \) is the torque vector with its elements consisting of \( \tau_{mi} \) given in Equation (3.8) and
\( i = \{1, ..., n\} \). Note that the motor inertia and damping are reflected to the output of the gearbox. Also \( J_{mi}, d_{mi}, d_{Li} \) and \( k_{si} \) are motor inertia, motor damping, link damping and series spring coefficient in joint \( i \), respectively. Moreover, the stiffness of the gear box is included in series with the elastic element. In the simplest case, when the drive stiffness is infinite, the time constant of the actuator is given by \( \frac{J}{B_m} \), that is \( \frac{13.87e-6}{19958e-5} = 0.69496 \) at the input of the gearbox for RBE1211 motor and CSD17 gearbox parameters, as provided in Appendix C. Hence, it is clear that the mechanical time constant dominates the actuator dynamics.

The state space equations to design the optimal stabilizing feedback can be derived by substituting Equation (3.8) in Equation (3.9)

\[
\dot{\tilde{x}} = \tilde{A}\tilde{x} + \tilde{B}u
\]  

(3.11)

where \( \tilde{x} = (q, \dot{q}, q_m, \dot{q}_m)^T \) is the state of the system and \( u \) is the vector valued input voltage to the system. The joint ordering used in the models is ankle lateral, ankle sagittal, knee, hip yaw, hip lateral, hip sagittal of the support leg and mirror of this ordering is used for the swing leg. Equation (3.11) is used in chapter 4 to design the single support and double support controllers.

Having derived the detailed continuous compliant dynamics of CoMan in single support and double support phases, the transitions between the two phases is considered next.

### 3.6 Contact Model

The transition from the single support phase to the double support phase is done via a heel strike with the ground. The heel strike in this study is modelled as a flat foot contact. This part of the walking cycle can be modelled by a rigid body impact or a compliant impact. Since CoMan uses series elastic actuators in the sagittal plane and has plastic shoes that contact with the ground, the compliant contact model is used in this thesis for walking simulation.

It should be noted that the impact model is primarily used as an event detection
mechanism to provide feedback to the robot to switch its trajectory generation phase from single support to double support rather than to control the complex impact dynamics. Hence, in this study the contact model is merely used for simulation. In the following sections, a brief overview of the rigid body impacts is given and the compliant impacts together with the assumptions are explained and finally the ground model is discussed.

### 3.6.1 Rigid Body Impacts

A rigid body impact is a discrete map that maps the rigid body velocity before the impact to the velocity after the impact via a coefficient of restitution. In the case of multibody systems, the impact map is derived via the conservation of angular momentum principle before and after the impact. It is often assumed that the impact is instantaneous and the positions do not change during the impact. In particular, the dynamics and impact map of a compass gait walker has been widely studied in the literature [161, 162, 163].

### 3.6.2 Compliant Impacts and the Ground Model

Compliant impacts are used in this thesis due to the compliant dynamics of the robot. There are different types of ground contact models in the literature which are mainly based on nonlinear spring-damper models [164, 165, 166, 167] that result in a realistic contact force via the deformation and deformation rate of the bodies in contact. The actual contact model of a foot with the ground is quite complex since each foot has an area in contact with the ground. However, this can be approximated by four contact points at each corner of the foot, as shown in Fig. 3.9, to simulate the compliant contacts. It should be noted that the vertical ground reaction force $F_z$ is unilateral, that is the ground can only push the robot but it cannot pull the robot towards itself. Therefore, forward kinematics is used to compute the absolute position of the contact points together with piecewise linear functions to apply the vertical ground reaction force only in one direction depending on the height of the contact point with
Table 3.2: Ground model coefficients.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
<th>Value</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>(K_G)</td>
<td>Vertical spring coefficient</td>
<td>(8 \times 10^4)</td>
<td>N/m</td>
</tr>
<tr>
<td>(D_G)</td>
<td>Vertical damping coefficient</td>
<td>(2 \times 10^4)</td>
<td>N.sec/m</td>
</tr>
<tr>
<td>(K_F)</td>
<td>Friction stiffness coefficient</td>
<td>(4 \times 10^6)</td>
<td>N/m</td>
</tr>
<tr>
<td>(D_F)</td>
<td>Friction damping coefficient</td>
<td>(2 \times 10^6)</td>
<td>N.sec/m</td>
</tr>
<tr>
<td>(\mu)</td>
<td>Damping coefficient</td>
<td>0.9</td>
<td>[-]</td>
</tr>
</tbody>
</table>

respect to the ground level

\[
F_z = \begin{cases} 
0 & z \geq z_0 \\
-K_G \Delta z - D_G \Delta \dot{z} & z < z_0
\end{cases}
\]

where \(\Delta z = z - z_0 < 0\), \(z_0\) is the ground level height which is zero on a flat ground and \(z\) is derived by a sensor measuring the absolute position of the contact point. \(\Delta z\) provides the relative ground deformation with respect to initial ground level \(z_0\). Only positive forces are applied, i.e. \(F_z > 0\) to implement unilateral vertical ground reaction forces. The horizontal ground reaction forces are either sliding or stiction forces. However, the horizontal static and sliding friction forces are bilateral and they are applied in the opposite direction of the motion (in the sliding case) or the force (in the static case). The horizontal force is

\[
F_{x,\text{slide}} = -\text{sgn}(\Delta \dot{x}) \mu F_z \quad (3.12)
\]

\[
F_{x,\text{stick}} = -K_F \Delta x - D_F \Delta \dot{x} \quad (3.13)
\]

where \(\mu\) is the coefficient of friction, \(\Delta x = x - x_0\), \(x_0\) is the location that each corner of the foot contacts the ground and \(x\) is the amount of horizontal displacement from the original contact point \(x_0\) which is derived by an absolute position sensor. \(\Delta x\) indicates the relative motion with respect to the initial point of contact \(x_0\). If \(|F_x| > \mu F_z\) the foot will slide on the ground and vice versa.

In addition, the type of the ground plays an important role in quantifying the spring-damper coefficients. For instance, if the ground is made of sand or rocks or carpet, different coefficients should be used to obtain a realistic ground reaction force model. Hence, each type of ground can be simulated by changing the spring-damper
3.7 Conclusions

In this chapter, detailed models of walking were discussed and linearized models with actuator dynamics in single support and double support were derived to be used for controller design. The contact model was then introduced to simulate the ground reaction forces and implement the hybrid model of walking in a unified simulation environment.

It was pointed out that multivariable models of bipedal robots are often not used for controller design and one of the contributions of this study is to benefit from these extra information in designing a better walking control system. This issue is considered in the following chapter where three novel algorithms for centralized LQR and decentralized feedback design are proposed. In addition, a novel constrained feedback design is also proposed for the double support phase that will respect the double support dynamics.
Chapter 4

Tracking Control System

The joint trajectory tracking control design (inner layer of the walking control system) is presented in this section. The trajectory generator, which is responsible for generating dynamically feasible and stable trajectories, is presented in chapter 6. Fig. 4.1 illustrates this hierarchial architecture. These two layers have to be consistent in terms of performance requirements such as walking speed and tracking bandwidth in order to realize a fast and dynamic walking gait. The humanoid multibody model is an unstable system. Hence, the tracking controller is responsible for stabilization of the robot’s upright posture and tracking the trajectory generator’s commanded joint motions. Due to the walking control system architecture, the tracking controllers play a significant role in realizing the desired dynamic motion. In particular, due to the compliant joints used in CoMan, to enhance its walking robustness, shock tolerance, safer interaction and potentially energy efficiency, designing a reliable controller is important. Although compliance can be used to store and release energy and to absorb shocks from the ground impacts to protect the actuators, control of a compliant robot becomes more complex due to the increased under-actuated DoF introduced by the springs. Hence, realizing a better performance in such robotic systems requires sophisticated state space controllers that utilize feedback from all the state variables of a compliant joint.

This chapter is organized as follows. In section 4.1, the most common servo control method, independent PID joint servo control is reviewed. In section 4.2, an
LQR based centralized feedback controller is formulated. In sections 4.3 and 4.4 two decentralized control design methods are proposed. In section 4.5, Luenberger reduced order observers are used to estimate the velocity from the position encoders. This method is chosen due to simplicity of implementation as discussed in chapter 8 and good stability margins when considered together with the output feedback which is explained in section 4.5.4. In this chapter, all control and observer designs are formulated in discrete time, based on the compliant single support model.

4.1 Independent PID Joint Control

Independent PID joint control is the simplest and the most widely used solution that treats each joint as an independent, SISO system. The coupling and interactions with other joints are treated as disturbances, as shown in Fig. 4.2. This approximation is more accurate when high ratio gear reducers (for instance 100:1) are used to reduce the reflected link inertia in comparison with the motor and gearbox inertia [59]. Independent PID joint control has been widely studied and several tuning techniques can be found in [59, 168]. The overall performance of a multi degree of freedom robot under this control strategy is adequate for low speeds but in order to achieve higher performance and bandwidth, multivariable design methods should be used that utilize the interactions in the feedback design process.
CoMan also uses independent joint PID controllers to track the desired reference position and velocity on the motor side while the link is left to oscillate with the natural frequency of the spring as shown in Fig. 4.3. Moreover, the current PID gains that are used in the joints are derived by trial and error for stable position (when the robot is fixed to a frame at the hip) i.e. the gains have to be re-tuned since the loading of the joints will change while robot is standing or walking. Table 4.1 shows the PID gains used in CoMan with feedback from the motor encoder. It should be noted that these values have opposite sign for the left leg due to the input-output polarity as described in Appendix E. The hip yaw joint is not considered for the controller design and experiments at this stage of development.

So far, a systematic way of tuning PID gains and studying the stability for CoMan...
Table 4.1: CoMan’s independent joint PID gains (right leg).

<table>
<thead>
<tr>
<th></th>
<th>Ankle roll</th>
<th>Ankle pitch</th>
<th>Knee</th>
<th>Hip roll</th>
<th>Hip pitch</th>
</tr>
</thead>
<tbody>
<tr>
<td>$K_p \ (\text{volts/rad})$</td>
<td>-186</td>
<td>-372</td>
<td>-168</td>
<td>-149</td>
<td>-223</td>
</tr>
<tr>
<td>$K_d \ (\text{volts.s/rad})$</td>
<td>-0.28</td>
<td>-0.56</td>
<td>-0.28</td>
<td>-0.28</td>
<td>-0.418</td>
</tr>
<tr>
<td>$K_i \ (\text{volts/(rad.s)})$</td>
<td>0.1865</td>
<td>0.373</td>
<td>0.1865</td>
<td>0.1865</td>
<td>0.2797</td>
</tr>
</tbody>
</table>

bipedal locomotion has not been carried out. Hence, one of the contributions of this work is to use the state space model of CoMan and the LQR optimal control solution to re-tune the PID gains for walking. Another challenge of this control problem is due to the joints’ series elasticity that requires state space methods to derive the optimal PID gains on the joint side and the PD gains on the motor side (as shown in Fig. 4.3) while the multivariable nature of the robot is taken into account.

In the next section, the detailed state space model of the robot with 10 DoF is used to design the optimal LQR feedback for joint tracking. At this stage, the upper body electronics have just been installed on the robot, and the two yaw DoF in each hip are not considered for simplicity. Therefore, the controller designs in the next section assumes the 10 DoF model that was available for experiments. The proposed LQR solution lies at the heart of the novel feedback methods that are presented in the next sections.

### 4.2 Centralized LQR Joint Control

In order to systematically design multivariable controllers which consider the coupling effects between the robot joints, the compliance at each joint and all the sensory information (in particular before and after the spring) while keeping the controller relatively simple, LQR optimal control is proposed for CoMan’s tracking control system. LQR is designed using detailed state space models that include the actuator dynamics and joint compliance. This design method has the advantage of a wide range of available linear control design tools and relative simplicity for digital implementation.
The choice of discrete time observer based linear optimal control (LQR) is motivated by its simplicity and yet optimal performance with respect to the conventional independent PID joint control and its good stability margins. Optimal in this context refers to the optimal formulation of the LQR problem that can optimally place the closed loop poles via minimizing a quadratic cost function of the states and the control inputs. However, in order to have energy optimality suitable walking trajectories will still have to be designed. The observer designs and velocity estimation theory implemented on CoMan is presented in section 4.5. LQR multivariable controller is implemented successfully on CoMan as described in chapter 8. The simulation results show that the centralized LQR can achieve higher bandwidth (1.4 Hz compared to the existing 1 Hz with the gains in Table 4.1). Once the values of the springs are known accurately, the reference trajectories can then be specified directly for the link positions instead of the motor positions which is currently practiced on CoMan. Also the peak torques of the actuators can be penalized in the LQR formulation that can provide a more energy efficient walking gait. Hence, applying a centralized LQR optimal control to CoMan as a more promising feedback control in comparison to the existing independent PID joint control, is one of the contributions of this study.

The LQR approach relies on the linearization of the multi-body system about the upright unstable equilibrium. The steps for deriving the linearized model in Robotran and Matlab are described in section 7.1.6. The state-space model derived in section 3.5 is used to design the controller.

4.2.1 Problem Formulation

Consider the discrete time, state-space model obtained from the discretization of Equation (3.11)

\[
\tilde{x}(k + 1) = \tilde{A}\tilde{x}(k) + \tilde{B}u(k) \\
y(k) = C\tilde{x}(k)
\]

where, \( \tilde{x} = (q, \dot{q}, q_m, \dot{q}_m)^T \) is the discrete time state of CoMan and \( u \) is the vector valued input voltage. In order to allow reference tracking, integral action is added to
the system
\[ z(k + 1) = z(k) + r(k) - y(k), \] (4.2)
where \( r(k) \) is the reference input vector. Therefore, the discrete tracking system is described by
\[ x(k + 1) = Ax(k) + Bu(k) + Br_r(k) \] (4.3)
where \( x(k) = \begin{bmatrix} \tilde{x}(k) \\ z(k) \end{bmatrix}, A = \begin{bmatrix} \hat{A} & 0 \\ -C & I \end{bmatrix}, B = \begin{bmatrix} \hat{B} \\ 0 \end{bmatrix} \) and \( Br_r(k) = \begin{bmatrix} \hat{G}_{ff} \\ I \end{bmatrix} \).

The state vector \( x(k) \) consists of \( n \) link positions, velocities, motor positions, velocities and integrators.

In order to obtain the discrete time optimal feedback gain \( K \), the state feedback law \( u = -Kx \) must minimize the cost function \( J \) subject to dynamics of the single support model. The optimisation problem is
\[
\begin{align*}
\min J = & \frac{1}{2} \sum_{k=0}^{\infty} (x(k)^T Q x(k) + u(k)^T R u(k)) \\
\text{subject to} & \quad x(k + 1) = Ax(k) + Bu(k)
\end{align*}
\] (4.4)
where \( Q \) is a Hermitian, positive semi-definite matrix and \( R \) is Hermitian positive definite matrix. It is well known that the state feedback gain \( K \) that minimizes \( J \) is
\[
K = (R + B^T PB)^{-1} B^T PA
\] (4.5)
where \( P \) is the solution to the following algebraic Riccati equation
\[
P = Q + A^T PA - A^T PB (R + B^T PB)^{-1} B^T PA.
\] (4.6)
Hence, the closed loop system can be formulated by
\[
x(k + 1) = A_d x(k) + Br_r(k)
\] (4.7)
where \( A_d = A - BK \), \( Br_r = [BG_{ff} \ 0]^T \) and \( G_{ff} = K[I_n \ 0]^T \). \( G_{ff} \) is referred to as the feed-forward gain and is derived from the first block of the feedback gain \( K \) that corresponds to the joint positions. In practice, a scaling factor is used to tune the feed-forward gain to avoid large overshoots in the transients. The feed-forward gain \( G_{ff} \) is used in a number of simulations in the decentralized designs, whenever
the feedback gains did not provide a fast transient response. Fig. 4.1 illustrates how this feed-forward gain is added to the control system.

Remark 1: In this chapter, the formulation is presented as a state feedback, hence it is assumed that the robot has sufficient resolution in the sensors to obtain the state variables directly from the sensor measurements. The velocity estimation is discussed in section 4.5.

4.2.2 Selection of Weighting Matrices: $Q$ and $R$

Tuning the closed loop time response is done by choosing the state penalty matrix $Q$ and the input penalty matrix $R$. Generally, choosing a small value for entries of $R$ in relation to the robot’s states leads to a faster response at the cost of higher actuator peaks. Moreover, $x(k)$ consists of motor and link positions and velocities as well as integrators. Hence, by penalizing the positions and integrators a good response can be derived. Often the motor states are not penalized, since from a kinematic perspective, the links motion is more important than the internal motor states. Also increasing the penalty on the integrators provides a faster response due to higher gains in the integrators dynamics.

4.2.3 Performance and Robustness of LQR

The models used for control design are always approximations of the real physical system. While the feedback is designed for the nominal model, the stability and sensitivity of the real physical closed loop system should be evaluated against the uncertainties such as un-modelled dynamics and parameter variations. The stability robustness in face of parameter uncertainties, quantization noise and load disturbances can be investigated via different methods. In the case of linear systems, frequency response measures, gain margin (GM) and phase margin (PM) are often used for measuring the robustness of the system to such uncertainties. Intuitively, GM describes how much the controller gain can be increased before the system go unstable while the PM describes how much phase lag or time delay can be tolerated
in the closed loop system before it goes unstable. The GM and PM provide a measure on how close the frequency response plot is to the critical point (-1 in SISO and the origin in MIMO systems) and they can be derived from the Bode or Nyquist diagrams. The further the frequency plot from the critical point, the higher the GM and PM and hence the more robust the closed loop system will be to the uncertainties.

In linear multivariable feedback design, the Nyquist diagram of determinant of the return difference at the input and output of the plant is utilized to serve as a crude estimate for relative stability. The return difference at the input and output of the plant are \( F_i(s) = (I_n + K(sI_p - A)^{-1}B) \) and \( F_o(s) = (I_p + (sI_p - A)^{-1}BK) \) respectively, where \( p \) is the dimension of \( A \). However, one of the advantages of using linear optimal control is the guaranteed good gain and phase margins. It is shown in [169, 170, 171] that

\[
F_i^H(j\omega)RF_i(j\omega) \geq R, \tag{4.8}
\]

and if \( R = \rho I \), where the scalar \( \rho > 0 \), Equation (4.8) reduces to

\[
F_i^H(j\omega)F_i(j\omega) \geq I,
\]

and PM would be at least 60 degrees with infinite GM (the gain in each loop can be increased without loosing stability) and 50% (6 dB) gain reduction tolerance. Further results on robustness of LQR is given in [172].

In general, GM and PM define an upper bound on the magnitude of possible perturbations but they do not track the origin of uncertainty to the exact parts of a plant. This crude approximation for such perturbations is referred to as unstructured uncertainty. However, there are more systematic methods for assessing uncertainty in a plant in form of an additive or input/output multiplicative uncertainty that can be used to analyse structured uncertainty if some information about the uncertain parts of a plant is available. Further details on frequency analysis of uncertainty is given in [173].
4.2.4 LQR Simulation for Compliant Model of CoMan

The linear compliant model of CoMan with 10 DoF, actuator dynamics and integrator dynamics is simulated. The simulation setup described in the following section serves as a standard for the next simulations presented in this chapter.

4.2.4.1 Simulation Set-up

The dynamic compliant model provided in Appendix B.2, is discretized with 1 ms. The same sampling is used for the decentralized feedback simulations to allow a comparison between the numerical results. The LQR penalties used in this design are $Q = \text{diag}\{2500 I_n, 0, 0, 0, I_n\}$ and $R = 0.5 I_n$. The integrators are placed on the link positions. The feed-forward gain $G_{ff}$ is not used in this simulation, because the LQR gains provide a desired speed for the transient response with no overshoot. The reference position is

$$r = [0.1, 0.2, -0.2, -0.1, 0.2, -0.2, 0.1, 0.2, -0.2, -0.1]^T$$

The reference position used corresponds to a lateral sway and a squat motion as shown in Fig. 4.4. The joints are ordered from the ankle of the support leg to the ankle of the swing leg as shown in Fig. 4.4. The numerical results are discussed next.

4.2.4.2 Discussion

The lateral and sagittal step response of the closed loop system and their corresponding control voltages are shown in Figs. 4.5 and 4.6. It is evident from the step response that the lateral plane is using half of the sagittal plane control input which is mainly due to the larger sagittal reference angles (for instance 0.35 rad for the knees) that requires higher voltages. The settling time of the step response is less than 0.3 sec. In this simulation an ideal case is considered where the quantization and stiction are not included. The optimal gain $K$ serves as the basis for design of the decentralized controller using an iterative scheme which is presented in the next
Figure 4.4: The white configuration of the robot legs represents the initial condition (that is the upright position) and the coloured configuration of the legs represents the desired and final posture of the legs that corresponds to $r$.

It is shown that when the structural constraints are imposed on the gains, a lower bandwidth is achieved in the decentralized designs.

Figure 4.5: LQR closed loop step response.
4.3 Decentralized LQR with Sparse Gradient Descent Feedback Design

Most multivariable control methods assume the notion of centrality, that is all the information about a system is available at a central location where the control calculations are performed. However, in many physical systems there are restrictions on the rate of information transfer among a group of sensors and actuators. In the case of CoMan, achieving faster sampling rates (1 kHz) and the ability to process information locally at the DSP level is the first motivation for decentralized feedback design. The second motivation is to derive model based algorithms for decentralized feedback design to replace the conventional trial and error approach to this problem in humanoid robots, as discussed in section 4.1. As it was mentioned, currently CoMan uses a PID on the motor side without any feedback from the links that can lead to uncontrolled (open-loop) oscillations during walking.
4.3.1 Background

Decentralized control has been studied for more than half a century. Early results were reported in 1962 by Radner, inspired by team decision theory [174]. In 1968, an important example was published by Witsenhausen that showed under some constraints for linear quadratic stochastic optimal control, a nonlinear controller can achieve greater performance than any linear controller [175]. This was important to illustrate that even for a linear plant the optimal solution to the decentralized problem could be a nonlinear controller. Later, Witsenhausen presented sufficient conditions under which a decentralized optimal control problem could be solved by a linear decentralized feedback [176]. Decentralized pole placement algorithms have also been derived in [70]. The computational complexity of this problem via convex optimisation was studied in [69]. A comprehensive survey on several decentralized control algorithms and theory of decentralized feedback control can be found in [71, 177].

4.3.2 Contribution

There is a gap in robotics literature on systematic decentralized feedback designs and the notion of decentralized has often been used in a different way with ad-hoc approaches to decentralized control [63]. In this section, an optimal decentralized gradient descent algorithm proposed by Lunze [177] in continuous time is converted to discrete time and applied to the rigid and compliant models of CoMan.

4.3.3 Problem Formulation

The general decentralized control problem can be formulated as the following optimisation problem

\[
\begin{align*}
\text{minimize} & \quad \|f(P, K)\| \\
\text{s.t.} & \quad K \text{ stabilizes } P \\
& \quad K \in S.
\end{align*}
\]
where \(\|f(P, K)\|\) is the norm of a closed loop map, \(K\) is the feedback that stabilizes the plant \(P\) and the subspace \(S\) defines the structure of \(K\). The general solution to problem (4.9) is currently being studied in control systems community where complex mathematical ideas have been proposed [178]. A necessary and sufficient condition called the quadratic invariance was recently proposed in [67, 68] which unifies the previous results. An important consequence of quadratic invariance property is to show that if \(K\) is block diagonal but the plant \(P\) is not, the quadratic invariance property does not hold and therefore the optimisation problem is not convex in feedback variables. This result is significant in LMI formulation of such problem.

In the proposed method, the problem is simplified by limiting the solution of (4.9) to linear, block diagonal structure gains as shown in (4.10). In addition, due to the excellent sensitivity and robustness properties of the LQR formulation (as discussed in section 4.2.3), the two decentralized algorithms proposed in this section and section 4.4 are based on the LQR formulation.

State ordering is an important point to be considered in the formulation. The ordering which is considered is joint positions, velocities and integrators for rigid models and joint positions, velocities, motor positions, velocities and integrators for compliant models with actuator dynamics. Therefore, in the rigid case the feedback gain is \(K = [K_P \ K_D \ K_I]\) and in the compliant case \(K = [K_P \ K_D \ K_{Pm} \ K_{Dm} \ K_I]\) where a block diagonal structure is imposed on the each block of feedback gain matrix \(K\). For instance

\[
K_P = \begin{bmatrix}
k_1 & 0 & 0 & \cdots & 0 \\
0 & k_2 & 0 & \cdots & 0 \\
0 & 0 & \ddots & \cdots & 0 \\
\vdots & \vdots & \ddots & k_{n-1} & 0 \\
0 & 0 & \cdots & 0 & k_n
\end{bmatrix}_{n \times n}
\]

(4.10)

where \(n\) is the number of joints and the diagonal elements \(k_i\) are scalars which represent the proportional, derivative or integral gain of the decentralized controller.
The LQR based decentralized feedback optimisation problem is

$$\min_{u = -\text{diag}(K_i)x} J$$

subject to $$x(k+1) = Ax(k) + Bu(k)$$

where $$J$$ is defined in Equation (4.4) and the discrete dynamics are derived in Equation (4.3). However, the optimal feedback gains “$$\text{diag}\{K_i\}$$” obtained from this optimisation depend on the initial condition $$x(0)$$. Therefore, to remove this dependency on the initial condition, an average optimal cost is used in [177, 179] that approximates the optimal cost $$J$$ by taking its average over a linearly independent set of initial conditions $$x(0)$$. It is assumed that $$x(0)$$ is a random variable with $$E[x(0)] = 0$$ and $$E[x(0)x(0)^T] = I$$, where $$E[.]$$ denotes the expected value. It is worth noting that all the states of CoMan are normalized and reflected to the joint side. Hence, the expected value of the cost function (4.4) is

$$\hat{J} = \frac{1}{2} \sum_{k=0}^{\infty} tr[(A_d^T)^k(Q + K^T R K)(A_d)^k]$$

(4.12)

where $$tr[.]$$ denotes the trace of a matrix and $$R, Q$$ are the LQR penalties. Let $$V(K)$$ be

$$V(K) = \sum_{k=0}^{\infty} (A_d^T)^k(Q + K^T R K)(A_d)^k$$

(4.13)

then the averaged cost $$\hat{J}$$ is

$$\hat{J} = \frac{1}{2} tr[V(K)]$$

(4.14)

It is shown in [179] that if $$A_d$$ is stable, $$V(K)$$ is solution to the following Lyapunov equation

$$A_d^T V A_d - V + Q + K^T R K = 0.$$  

(4.15)

An iterative scheme is proposed to solve the optimal control problem using the following gradient descent approach

$$K_{i+1} = K_i - \alpha \frac{d}{dK} tr[V(K_i)]$$

(4.16)

where $$\alpha > 0$$ is a fixed step size used to achieve convergence to the optimal value. The value of the gradient is given as

$$\frac{d}{dK} tr[V(K_i)] = ((R + B^T V_i B)K_i - B^T V_i A)W_i$$

(4.17)
where \( i \) is the iteration number, \( W \) is the solution to the following equation and it only exists if and only if the closed loop system \( A_{cl} \) is stable

\[
A_{cl} W A_{cl}^T - W + I = 0. \tag{4.18}
\]

Note that the transpose has changed in Equation (4.18) in comparison to Equation (4.15). In the design of decentralized controllers, the actual optimal control solution has to be projected on to a sparse matrix \( K \) which will be used by the local controllers. The following updating policy is proposed by [177]

\[
K_{i+1} = K_i - \alpha \left( \frac{d}{dK} \text{tr}[V(K_i)] \ast E_a \right) \tag{4.19}
\]

where the operator “ \( \ast \) ” denotes the element-wise matrix multiplication as used in Matlab.

This theory is applied to the rigid system without actuator dynamics just to compare the performance of the centralized and decentralized control strategies on the dynamics of the ideal mechanical system. In the rigid case \( E_a = [I_n I_n I_n] \) and in the compliant case with actuator dynamics \( E_a = [I_n I_n I_n I_n I_n] \). Basically \( E_a \) selects the diagonal elements of \( K = [K_P K_D K_I] \) in the rigid model or \( K = [K_P K_D K_{Pm} K_{Dm} K_I] \) in the complaint model, at each iteration and sparsifies \( K \) to keep the decentralized architecture. The algorithm presented above maintains the structure and stability of the decentralized controller, provided the initial decentralized gain \( K_0 \) stabilizes the closed loop system \( A_{cl} = (A - BK_0) \).

In order to obtain an initial \( K_0 \), the centralized LQR gain \( K \) is used in \( K_0 = K \ast E_a \), where \( K_0 \) is verified afterwards to be a stabilizing feedback gain for the multivariable closed loop system \( A_{cl} \).
Algorithm 4.1: Sparse gradient descent feedback computation (LQR-GraDe)
This algorithm takes the discrete time, state space matrices \((A, B)\), the LQR penalties \((R, Q)\), the initial stabilizing feedback \(K_0\), \(\alpha\) and computes the decentralized \(K\).

1. Compute \(V_i\) and \(W_i\) as solutions of Equations (4.15) and (4.18).
2. Compute \(\frac{d}{dK}tr[V(K_i)]\) from Equation (4.17).
3. Compute \(K_{i+1}\) from Equation (4.19).
4. Stop the iterations if \(\alpha\|\frac{d}{dK}tr[V(K_{i-1})] \ast E_{\alpha}\| < \epsilon\) is satisfied for a threshold \(\epsilon\), otherwise go to step 1.

The numerical results for the rigid and compliant robots are presented next.

4.3.4 LQR-GraDe Simulation for Rigid Model of CoMan

In this section, Algorithm 4.1 is applied to the linear model of CoMan with 10 DoF \((n = 10)\), while the compliance and actuator dynamics are not considered. The purpose of this section is to gradually build the simulation results from a simpler rigid model to a more complex compliant model. The joint angles are ordered as ankle lateral, ankle sagittal, knee, hip lateral, hip sagittal for the support leg (right leg) and the mirror of this ordering is used for the swing leg (left leg). The continuous model is discretized with 1 ms sampling time. This sampling time is chosen to allow a comparison between this algorithm and the LMI based decentralized design that is presented in the next section. The discrete time state space equations are given in (4.3) where the states are ordered as joints’ angles, joints’ angular velocities and integrators, i.e. \(x = [q, \dot{q}, z]^T\). The LQR penalty matrices used are \(Q = \text{diag}\{2500 I_n, I_n, I_n\}\), \(R = 5 I_n\). These penalties are used to derive the centralized LQR gain as well as the decentralized LQR-GraDe gains. The step size \(\alpha\) is set to \(4 \times 10^{-10}\) and after 100 iterations the cost of the decentralized LQR is minimized from \(1.435 \times 10^8\) to \(1.005 \times 10^8\) while the optimal LQR cost is \(0.764 \times 10^8\) as shown in Fig. 4.7.
The feed-forward gain $G_{ff}$ used is scaled by half in this simulation. The reason for using the feed-forward gain was because the feedback itself did not provide the desired transient speed. Hence, $G_{ff}$ can partly compensate the initial slow response. The initial condition for the simulation is $x = 0$ and the commanded reference position is

$$r = [0.1, 0.1, -0.2, -0.1, 0.1, -0.1, 0.1, 0.2, -0.1, -0.1]^T$$

The joint angles ordering used in this simulation is ankle lateral, ankle sagittal, knee, hip lateral, hip sagittal for the support leg (right leg) and mirror of this ordering is used for the swing leg (left leg). CoMan's final posture corresponds to bending the ankles, knees and hips in the sagittal plane and a sway in the lateral plane. The joints' positions for the LQR and the decentralized gains are shown in Fig. 4.8 and the corresponding torques are shown in Fig. 4.9. A comparison between the centralized and decentralized joints' positions shows that, as expected, the centralized LQR is performing better and has a faster settling time. In particular, the ankle and knee joints of the support leg in sagittal plane (pitch angles) under decentralized control have a slower response and their corresponding transient and the steady states torques are larger than the centralized LQR joint torques. The bandwidth of this closed loop system is approximately 1 Hz. Reducing the bandwidth of the system by increasing the input penalty $R$ results in a poorer decentralized closed loop performance. On the other hand, increasing the bandwidth by reducing the penalty $R$ has the problem that the initial sparsified feedback $K_0 = K_* E_a$ is not stabilizing. It should be noted that this deterioration in performance (speed of response) and energy efficiency (joint torque) is far less apparent in continuous time as reported by the author in [73], but the bandwidth limitation for obtaining the initial stabilizing gain is still the main limitation of this algorithm in continuous time.

The $Q$ and $R$ penalties chosen for this design correspond to the highest bandwidth that would result in an initial stabilizing feedback gain to start the gradient descent iteration. Considering the discrete time numerical results of this algorithm, the LQR provides a better performance in terms of bandwidth and also smaller transient and steady state joints' (ankle and knee) torques (the only limitation would be
Figure 4.7: Cost of the LQR objective function $\hat{J}(K_i)$ shown against the number of iterations.

Figure 4.8: A comparison between CoMan’s decentralized (dashed line) and centralized LQR (solid line) joints’ positions step response.
the actuators saturation levels). The next section applies this method to the compliant case with actuator dynamics and provide further insights into the limitations of this algorithm.

4.3.5 LQR-GraDe Simulation for Compliant Model of Co-Man

In this section, Algorithm 4.1 is applied to the linear model of CoMan with 10 actuated DoF ($n = 10$) with compliance and actuator dynamics. It should be noted that Algorithm 4.1 is a solution to the problem of decentralized LQR design, but more general constrained optimisation algorithms such as Linear Matrix Inequality (LMI) can be used to design the decentralized controllers. This is discussed in the section 4.4. The same joint ordering (rigid case as described in the previous section) is used in this simulation. All 10 DoF joints have finite stiffness, but ankle pitch, knee and hip pitch have lower stiffness with 188 ($Nm/rad$), 185 ($Nm/rad$)
and 982 (Nm/rad) spring constants, respectively. The continuous model is discretized with 1 ms sampling time. The discrete time state space model given in Equation (4.3) is used, where the states are ordered as joints’ angles, velocities, motors’ angles, velocities and integrators, i.e. \( x = [q, \dot{q}, q_m, \dot{q}_m, z]^T \). The matrix \( E_a = \begin{bmatrix} I_n & I_n & I_n & I_n \end{bmatrix} \). The LQR penalties are given below where \( q_m \) and \( \dot{q}_m \) are not penalized, \( Q = \text{diag}\{2500 I_n, 20 I_n, 0, 0, 2 I_n\} \), \( R = 20 I_n \). These penalties are used to derive the centralized LQR gain as well as the decentralized LQR-GraDe gains for comparison. The step size \( \alpha \) is set to \( 2e^{-12} \) and after 140 iterations the cost of the decentralized LQR is minimized from the initial value of \( 7.4284 \times 10^9 \) to \( 1.6201 \times 10^9 \) while the optimal LQR cost is \( 0.2781 \times 10^9 \) as shown in Fig. 4.10. The ankle pitch and knee (support leg) integrator gains derived from the gradient descent approach are small \((0.0068, 0.099)\) in comparison with the other joints integrator gains that is about \( 0.18 \). Increasing these gains manually results in joint oscillations. In this simulation both these gains are set to 0.1 to improve the step response settling time. However, this is one of the drawbacks of using gradient descent approach that the optimisation might not lead to the global optimum and the response might not be desirable.

The feed-forward gain \( G_{ff} \) is not used in this simulation. The initial condition
and commanded reference position are the same as the previous simulation. The lateral and sagittal joints’ positions for the LQR and the decentralized gains are shown in Fig. 4.11 and the corresponding motor voltages are presented in Fig. 4.12. In terms of speed of response, the centralized LQR, as expected, has a better settling time (about 0.8 sec) while the decentralized response settles after 2 seconds. On the other hand comparing the control signals (motor voltages) the decentralized controller is resembling the centralized voltages more closely than in the previously considered rigid case where both the decentralized transient and steady state control inputs were larger than the optimal LQR. However, in both cases the ankle pitch and knee of the support leg showed a slower response compared to the other joints. The decentralized closed loop bandwidth in the lateral and sagittal plane are about 1 Hz and 0.5 Hz while the centralized LQR has about 1 Hz bandwidth in both planes.

By integrating the actuator and compliance dynamics to the rigid body model, the result of the decentralized controller provided a PD on the motor side and a PID on the link side, since the integrators are closed on the link angles. The continuous
results of this section for rigid robots are published by the author in [73]. This algorithm has a better performance in continuous time and the comparison between the centralized LQR and decentralized LQR shows a more comparable bandwidth. A major drawback of this method is the low bandwidth required for finding the first decentralized stabilizing gain $K_0$ to start the iterations. However, it is shown in the next section that the LMI approach can improve on this limitation, provided the LMI problem is feasible.

Moreover, it should be noted that for bipedal robots it is often desired to have a symmetry in the gains designed for each leg. However, both the centralized LQR and the decentralized LQR-GraDe methods compute an asymmetric gain that has higher gains for the support leg and lower gains for the swing leg. The difficulty that arises in the implementation of such controllers is that the feedback must be switched when the robot changes the support leg that complicates the control system. The LMI-LQR formulation presented next is an attempt to incorporate symmetry into the decentralized gains.
4.4 Decentralized LMI-LQR Feedback Design

The general optimal decentralized control problem in Equation (4.9) (particularly feedback design with a block diagonal structure) is a non-convex optimisation problem. Hence, control problem (4.9) can be approached in two ways. The first is to approximate the solution by convex optimisation. The second is to directly formulate the global non-convex optimisation in terms of Bilinear Matrix Inequality (BMI) method [180].

However, non-convex optimisation is much harder than convex optimisation where efficient solvers are available. Therefore, this section considers the control problem (4.9) via the second approach (convex optimisation), in terms of the LMI method. The LMI method provides a more viable solution to the problem of decentralized feedback design where various type of structures including symmetry can be formulated. In addition, there are algorithms for LMI problems that:

1. are globally convergent (good initial guess not needed),
2. compute the global optimum, or find proof of infeasibility, and
3. can specify performance such as LQR cost.

LMI has been widely studied in the literature for the purpose of feedback synthesis that is discussed next.

4.4.1 Background

The study of LMI dates back to 1890 when Lyapunov published his seminal article on stability of linear dynamical systems [181]. Since then, LMI has been applied to many practical control engineering problems by Lur’e, Yakubovich, Popov, Kalman and others. In the late 1980’s interior-point algorithms were developed that together with the advances in computing power, led to a wide range of control problems being solved efficiently [181, 182].

LMI arises in many control engineering problems. In particular, the solution to the Lyapunov and Riccati equations can be obtained via the LMI approach.
proves to be valuable in cases where analytical or closed form solutions do not exist. The optimal decentralized control problem has been considered in the literature in this framework. One approach to this problem is the \( V - K \) iteration that can handle constraints on the controller structure [183]. The authors of this approach have acknowledged the difficulties in design of optimal fixed structure controllers in terms of computational complexity (classified as NP-hard problems) and proposed the heuristic \( V - K \) iteration. This method solves two convex optimisation problems based on LMI, by either fixing the Lyapunov function \( V \) or the controller \( K \) at each iteration. The algorithm does not provide a guarantee for convergence and it depends on proper initialisation of the decision variables \( V \) and \( K \). However, it finds a local optimum and it is reported to work well in practice.

In [184] the decentralized control problem was studied in the LMI framework and sufficient conditions for decentralized feedback synthesis were suggested which minimizes the \( H_2 \) norm of a linear discrete system. Similar sufficient condition was implicitly mentioned in [185]. Later, in [186] an extended LMI formulation for Lyapunov stability was proposed that provides a less conservative sufficient condition for decentralized design.

In [187] it was reported that necessary and sufficient conditions for existence of decentralized static output feedback were derived. In this method, the problem was reduced to a Quadratic Matrix Inequality (QMI) and the LMI method was used iteratively to solve the QMI numerically. An algorithm was presented on how to solve the LMI with one decision variable kept fixed at each iteration. However, no rigorous proof on the convergence of this algorithm to the global minimum was given and the results were illustrated for three and four dimensional state space systems.

Although decentralized feedback design has been widely studied in the control engineering literature and numerous algorithms exist in the LMI framework [188, 189], the robotics’ literature lacks a systematic and effective decentralized control algorithm.
4.4.2 Contribution

This section proposes an LMI design for formulating the decentralized LQR gains in discrete time similar to [184] with the difference that it is formulated as an LQR problem. This method is novel for the design of decentralized joint control systems for bipedal robots. Most of the numerical results in the literature are for low dimensional systems but the proposed method is applied to a 50 dimensional state space model of CoMan. It is shown that this algorithm can be used to achieve the desired bandwidth (provided the LMI feasibility) with less limitation which was faced in the previous gradient descent design. Numerical simulations for a 10 DoF model of CoMan and the compass gait model (Appendix D) are provided to illustrate the use of this method. Both rigid and compliant cases are considered.

4.4.3 Problem Formulation

In order to benefit from the excellent stability margins of the LQR formulation (as discussed in section 4.2.3), the LMI formulation is proposed with the performance specified in terms of the LQR performance index. The proposed discrete time, LMI-LQR formulation is given in Algorithm 4.2, where the discrete dynamics are given in Equation (4.3) with noise added to the system, $Q$ and $R$ are the LQR penalties and $P$ is the solution of the Lyapunov equation.

**Algorithm 4.2**: Decentralized LMI-LQR feedback computation.
This algorithm takes the discrete time, state space matrices $(A, B)$, the LQR penalties $(R, Q)$, the noise variance $\beta$, and computes the decentralized $K = YP^{-1}$. The LMI variables are $P, Y$ and $X$ with block diagonal structure.

$$
\min_{(P,Y,X)} \quad tr[QP] + tr[X] \quad \text{subject to:} \quad (4.20)
\begin{bmatrix}
(P - \beta I) & (AP - BY) \\
(AP - BY)^T & P
\end{bmatrix} > 0
$$

$$
\begin{bmatrix}
X & R^\frac{1}{2}Y \\
Y^T R^\frac{1}{2} & P
\end{bmatrix} > 0
$$


The LMI Algorithm 4.2 is developed in two parts. The first part formulates the stability of the stochastic linear system in terms of the state covariance matrix and the corresponding Lyapunov equation. The second part chooses the quadratic LQR performance index to formulate the optimal feedback gain as the solution to the LMI convex optimization problem.

Consider the discrete time system in Equation (4.3) with noise $\eta$ and zero reference inputs

$$x(k+1) = Ax(k) + Bu(k) + \eta.$$ \hspace{1cm} (4.21)

Assuming that the state $x$ is available for measurement and the pair $(A, B)$ are controllable, the feedback can be expressed as $u(k) = -Kx(k)$ and the closed loop system is

$$x(k+1) = (A - BK)x(k) + \eta$$ \hspace{1cm} (4.22)

where $(A - BK)$ is asymptotically stable. The steady-state state covariance matrix $P = E[xx^T]$ is the solution to Lyapunov equation

$$P - (A - BK)P(A - BK)^T - \hat{Q} = 0 \hspace{1cm} (4.23)$$

where the noise covariance matrix is $E[\eta\eta^T] = \hat{Q}$. If every entry in the noise vector has the same variance $\beta$ and the entries are all statistically independent or uncorrelated then the noise covariance is $E[\eta\eta^T] = \beta I$. Hence,

$$P - \beta I - (A - BK)P(A - BK)^T > 0.$$ \hspace{1cm} (4.24)

Since this inequality is nonlinear in $K$ and $P$, a change of variable is necessary. Equation (4.24) can be written as

$$P - \beta I - (AP - BY)P^{-1}(AP - BY)^T > 0$$ \hspace{1cm} (4.25)

where $Y = KP$. Equation (4.25) can be expressed as the Schur complement of $P$

$$\begin{bmatrix} (P - \beta I) & (AP - BY) \\ (AP - BY)^T & P \end{bmatrix} > 0 \hspace{1cm} (4.26)$$
The above LMI equation represents the set of feedback gains $K$ that stabilize the linear system. In order to find the optimum gain, the LQR cost function is derived in terms of the penalized state and control input as reported in [171]

$$y(k) = \begin{bmatrix} Q^\frac{1}{2} & 0 \\ 0 & R^\frac{1}{2} \end{bmatrix} \begin{bmatrix} \tilde{x}(k) \\ u(k) \end{bmatrix} = \begin{bmatrix} Q^\frac{1}{2} \\ -R^\frac{1}{2}K \end{bmatrix} x(k)$$  \hspace{1cm} (4.27)

where $Q = Q^T \geq 0$ (positive semi-definite and symmetric) and $R = R^T > 0$ (positive definite and symmetric). Then the expected LQR cost function can be defined as

$$\bar{J} = E[tr[y^T y]] = E[tr[y y^T]].$$  \hspace{1cm} (4.28)

The simplified cost function in terms of the LQR penalty matrices is

$$\bar{J} = tr[QP] + tr[R^\frac{1}{2}KPK^T R^\frac{1}{2}].$$  \hspace{1cm} (4.29)

However, the second term of the cost above is nonlinear in $K$ and $P$ and must be redefined for the LMI representation. Let

$$X = R^\frac{1}{2}KPK^T R^\frac{1}{2}$$  \hspace{1cm} (4.30)

where $X$ is a symmetric LMI variable. Equation (4.30) can be written as a Schur complement

$$\begin{bmatrix} X & R^\frac{1}{2}Y \\ Y^T R^\frac{1}{2} & P \end{bmatrix} > 0.$$

(4.31)

The LMI optimization problem can be written as

$$\min_{(P,Y,X)} tr[QP] + tr[X]$$

subject to (4.26) and (4.31). The centralized feedback is obtained by

$$K = YP^{-1}.$$  \hspace{1cm} (4.33)

In general, the feedback gain in (4.33) has a centralized structure. That is the feedback for each joint depends on states of the joint itself and other joints. In fact, without imposing constraints on $K$ the solution of (4.32) subject to (4.26) and (4.31)
is the same as the discrete time LQR solution. In other words, without the constraints, this method is a computationally expensive way of computing the LQR feedback. However, for practical implementation on local joint DSP controllers, it is desirable to have the decentralized structure.

4.4.4 Decentralized and Symmetric Gains

Ideally, the decentralized structure must be directly imposed on the feedback gain $K$ as in (4.34). However, the product between $K$ and $P$ in (4.24) causes a nonlinearity, so the decentralized structure is indirectly imposed on the LMI variables $P$ and $Y$. It can be easily shown that if these variables have a block diagonal structure the resulting feedback gain in (4.33) will have the decentralized structure. Hence, in this paper, $P$ and $Y$ are defined to have similar structure as in (4.34). The only difference is that diagonal blocks of $P$ are square matrices of dimension 5 (in the case of a compliant model) and diagonal blocks of $Y$ have the same dimension as blocks of $K$.

Moreover, symmetric decentralized gains are desirable in bipedal walking to avoid switching the gains while the robot changes the supporting leg. These gains are derived by providing a mirror reflection of the blocks with respect to the central block. It is important to point out that this is only a sufficient condition and the decentralized LMI problem may or may not be feasible. The symmetry will decrease the number of LMI decision variables and the optimisation problem has less degrees of freedom.

The state ordering in this formulation is different from Algorithm 4.1, in the sense that a permutation matrix is used to group the variables of each joint together, i.e. the link position, velocity and motor position, velocity and integrator state of each joint is grouped together via a similarity transformation on the state space matrices, where the transformation matrix is a permutation. The diagonal feedback structure
is

\[
K = \begin{bmatrix}
K_1 & 0 & 0 & \cdots & 0 \\
0 & K_2 & 0 & \cdots & 0 \\
0 & 0 & \ddots & \cdots & 0 \\
\vdots & \vdots & \ddots & \cdots & K_{n-1} \\
0 & 0 & \cdots & 0 & K_n
\end{bmatrix}
\]  \tag{4.34}

(4.34)

where \( n \) is the number of joints and \( K_i \) blocks are row vectors of dimension 1 by 3 for rigid models and 1 by 5 for compliant models. Each block \( K_i \) contains three PID gains for rigid robots or five PD-PID gains for compliant robots that provide feedback from the \( i \)th joint position, velocity, motor position, motor velocity and integrator states as illustrated in Fig. 4.3. In order to solve the LMI problem, the formulation must be well posed which is discussed in the next section.

### 4.4.5 LMI Well-Posedness

The LMI solvers in Matlab are based on interior-point optimization techniques and they require the feasible set to have a nonempty interior, which means the linear matrix inequality \( L(x) \) must be strictly feasible. However, in some cases \( L(x) \leq 0 \) can be feasible while \( L(x) < 0 \) might not hold, which means \( L(x) \) is not strictly feasible. This issue is addressed by reformulating the LMI problem as

\[
\min t \text{ subject to } L(x) < tI.
\]

The LMI constraint \( (L(x) < t I) \) is always strictly feasible in \( x \) and \( t \) if and only if the global minimum of \( L(x) \leq 0 \) satisfies \( t_{\min} \leq 0 \). In case the LMI is not strictly feasible the Matlab implementation of this formulation \textit{feasp} tries to reach the global minimum \( t_{\min} = 0 \).

In next sections, numerical results of this formulation is shown on a 10 DoF model of CoMan, both when the joints are assumed to be rigid and when the compliance and actuator dynamics are included in the model. The 10 DoF results are compared with the gradient descent results. Appendix D provides a lower dimensional simulation of applying the LMI-LQR algorithm to rigid and compliant compass gait models.
4.4.6 LMI-LQR Simulation for the Rigid Model of CoMan

The LQR based decentralized LMI algorithm is applied to the 10 DoF model of CoMan without the actuator dynamics. The model is discretized with 1 ms which was also used for the gradient descent feedback design. The input penalty matrix $R = I_n$ and $Q = \text{diag}\{\text{angles, velocities, integrators}\}$, where

\[
\text{angles} = \text{diag}\{200, 2000, 200, 2000, 400, 200, 400, 200\}
\]

\[
\text{integrators} = \text{diag}\{100, 5000, 6000, 100, 2000, 2000, 100, 2000, 1000, 100\}
\]

The velocities are not penalized ($\text{velocities}=0$). The LMI formulation has 145 decision variables and the final feedback gains are derived after 136 iterations using Matlab LMI toolbox. The positions and control inputs corresponding to the step response of the decentralized closed loop system are shown in Figs. 4.13 and 4.14. The approximate tracking bandwidth of the closed loop system is about 1 Hz.

The largest closed loop eigenvalue is 0.998 while the largest eigenvalue for the closed loop system with the centralized LQR gain designed with the same penalties is 0.992. Hence, the centralized LQR has the advantage of a faster settling time with the same penalties. As mentioned earlier, one of the main advantages of this algorithm

![Figure 4.13: Closed loop step response of CoMan’s lateral (top) and sagittal (bottom) joint angles using the decentralized gains.](image-url)
CHAPTER 4. TRACKING CONTROL SYSTEM

Figure 4.14: The control input (torques) of CoMan’s lateral (top) and sagittal (bottom) joints using the decentralized gains.

is the ability to put constraints on the feedback gain structure while minimizing the LQR cost function.

The decentralized and symmetric gains could not be derived using the rigid model discretized at 1ms sampling time, since the LMI formulation was found to be not feasible. However, discretizing the model with 0.1 ms sampling time (10 kHz) provided a feasible solution for decentralized and symmetric gains. The same penalties can be used as mention earlier with the only difference that the integrators’ penalties is multiplied by 0.02 to avoid aggressive behaviour of the controller and actuator saturation. The resulting positions and torque inputs corresponding to the step response of the decentralized closed loop system are shown in Figs. 4.15 and 4.16. It can be seen from the closed loop step response that the bandwidth has improved considerably which is mainly due to faster sampling. Although the derived symmetric gains are decentralized and can be implemented on the DSP on CoMan, the DSPs does not have the required speed to sample the joint at 10 kHz so future work will consider other software such as BMI solvers to design the symmetric feedback gains at 1 kHz sampling time to implement at the DSP level. Another disadvantage of sampling too fast is amplification of the quantization noise. Therefore, the DSP speed should be
consistent with the sensor resolution (encoders) used in the joints to obtain a suitable velocity estimate at high sampling rates.

Figure 4.15: Closed loop step response of CoMan’s lateral (top) and sagittal (bottom) joint angles using the symmetric and decentralized gains.

In section 4.4.7, a 10 DoF model of CoMan with compliance and actuator dynamics is used to design the decentralized gains and comparisons are made with the gradient descent results.

4.4.7 LMI-LQR Simulation for the Compliant Model of Co-Man

This algorithm is applied to the compliant model of CoMan with 10 joints coupled with actuator dynamics via springs. The model is discretized with 1 ms and the augmented state space model is formed as in (4.3). The input penalty matrix $R = 0.0001 I_n$. The $Q$ penalty matrix is formed as $\text{diag}\{\text{links}, 0, 0, 0, \text{integrators}\}$, where the links penalties are $\text{links} = \text{diag}\{10, 100, 10, 100, 10, 100, 10\}$ and integrators $= \text{diag}\{50, 1000, 1000, 100, 100, 200, 200, 200, 200, 50\}$. The link velocities and motor states are not penalized.

The sparse LQR feedback gain using the penalties mentioned is unstable and the gradient descent approach in section 4.3 cannot be used. However, the LMI based
approach does not have that limitation and it is applied to this model. This LMI formulation has 255 decision variables that are used in the optimisation and the LMI solution is computed over 140 iterations using the Matlab LMI toolbox.

The step responses of the decentralized closed loop system in lateral and sagittal planes are shown in Fig. 4.17 and Fig. 4.18 where the joint and motor angles are plotted with solid and dashed lines. The corresponding controller voltages are shown in Fig. 4.19. The approximate tracking bandwidth of the closed loop system in lateral plane is about 1 Hz and in the sagittal plane it is about 2 Hz.

Similar issue with the sampling time is observed in this simulation, since the decentralized and symmetric feedback design was not feasible at 1ms sampling time while reducing the sampling time to 0.1 ms leads to the desired structure on the feedback gains.

As mentioned earlier, the LMI formulation proposed in Algorithm 4.2 is based on a sufficient condition that if satisfied results in a stable decentralized feedback gain. However, there are cases where the LMI constraints are not feasible and extended methods can be used to add additional degrees of freedom to the LMI optimisation and enlarge the set of feasible gains. The next section discusses this issue.
Figure 4.17: Closed loop step response of CoMan’s lateral joint and motor angles.

Figure 4.18: Closed loop step response of CoMan’s sagittal joint and motor angles.
4.4.8 Future Work

The discrete time LQR based LMI formulation proposed in section 4.4 is based on the classical LMI representation of Lyapunov stability

\[
\begin{bmatrix}
P & AP \\
P A^T & P
\end{bmatrix} > 0.
\]  

(4.35)

It requires the decentralized structure to be imposed directly on \( P \) that limits the number of decision variables used in the optimisation. An extension to this method can be used by utilizing the new condition that is recently proposed in the literature [186]. Based on this new condition (4.35) is modified to include a new instrumental variable \( G \) that is a general matrix with arbitrary structure

\[
\begin{bmatrix}
P & AG \\
G A^T & G + G^T - P
\end{bmatrix} > 0.
\]  

(4.36)

Introducing the new instrumental variable \( G \) allows the optimisation variables associated with controller parameters to be independent of the symmetric matrix \( P \) that is used for stability. Future work will consider the new conditions to extend the proposed method. In addition, \( V - K \) iteration [183], iterative LMI [187] need to be investigated to verify their efficiency in decentralized design for high dimensional...
problems that arise in humanoid robotics.

Moreover, the problem can be formulated as a BMI [180] to provide a necessary and sufficient condition for the decentralized feedback design. However, the solvers for BMI problems are only available commercially [190].

The controllers presented in sections 4.2, 4.3 and 4.4 are all based on the single support model of CoMan which is an open kinematic chain. The next section, uses the single support model to design linear Luenberger observers.

4.5 Velocity Observer

CoMan uses 12 bit encoders to measure the joint positions and 17 bit encoders for the motor positions. However, PID or LQR feedback require velocity signal to compute the control inputs. Since CoMan has position encoders the velocity information must be estimated from the position measurements. In the case of full state LQR feedback the joints’ velocities and motors’ velocities should be estimated. The linear system is fully observable. The diagram of the observer together with the feedback is shown in Fig. 4.20 where the LQR feedback is divided into $F_1$, $F_2$ and $F_3$ which are corresponding gains for joint and motor positions, joint and motor velocities and integrators. The observer block shown in the figure is the general case discussed in section 4.5.2, but it can also be replaced by the unknown input observer [191] if the dotted line is removed or the difference and averaging scheme which is discussed in section 4.5.1. All the three methods are implemented in the experiments on CoMan as presented in chapter 8. Among these methods, the Luenberger reduced order observer and the difference and averaging scheme provided stable and reliable results.
Figure 4.20: Control system diagram of observer based LQR. The mark “x” is used to show where the loop is broken to investigate the relative stability.

The relative stability of each observer is discussed in section 4.5.4 where the loop is broken at the input of the plant as shown with the mark “x” in Fig. 4.20 and the frequency response of the loop gain is used to study the stability margins.

4.5.1 Difference and Averaging

The simplest method for estimating the velocities is to use the first order difference scheme. This method provides a good estimate for the motor velocities but due to quantization noise dynamic observers must be used to derive the joint velocities.

In order to compute the velocity estimate difference and averaging scheme is used that with the current sensor resolution (12 bits) provides a noisy velocity signal, particularly at low speeds. A simple calculation shows that the minimum speed that can be detected with 12 bits resolution with 1 ms sampling time is $1.5 \, \text{mrad/1 ms} = 1.5 \, (\text{rad/s})$. Averaging this signal over the last $l$ samples provide a smaller velocity spike $(1.5/l)$ but leads to a delay in the control system’s feedback loop.

The effect of first order differentiation can be studied by the following transfer function matrix

$$G_c(z) = \begin{pmatrix} (1 - z^{-1}) & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & (1 - z^{-1}) \end{pmatrix}$$ (4.37)
which requires one state variable per joint to store the previous position state. The transfer function matrix for differentiation and averaging over the last $l$ samples is

$$G_c(z) = \frac{1}{l} \begin{pmatrix} (1 - z^{-l}) & \ldots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \ldots & (1 - z^{-l}) \end{pmatrix}$$ (4.38)

which requires $l$ state variables per joint to store the last $l$ samples of the position. In practice, the state variables (observer’s memory) can be initialized with the first position reading from the encoder (instead of a memory filled with zeros) to avoid undesired transient behaviour in experiments. Linear reduced order observers are proposed in section 4.5.2 to improve the estimated velocity signal and the relative stability of the system.

### 4.5.2 Luenberger Observer

In this section, the design of the Luenberger discrete time reduced order observer is briefly discussed. Hence, the overall control system uses position measurements of $n$ joints and $n$ motors and estimates the remaining $2n$ velocities by means of the reduced order observer.

Let the discretization of the state space model given in (3.11) be

$$\hat{x}(k+1) = \tilde{A}\hat{x}(k) + \tilde{B}u(k)$$ (4.39)

where $\hat{x} = (q, \dot{q}, q_m, \dot{q}_m)^T$ is the state of the system with dimension $4n$ and $u$ is the vector valued input voltage to the system. The output is

$$y_m(k) = C\hat{x}(k) = \begin{pmatrix} I_n & 0 & 0 \\ 0 & I_n & 0 \end{pmatrix} \hat{x}(k)$$ (4.40)

and the observer’s equations are

$$w(k+1) = Ew(k) + H_o u(k) + G_o y_m(k)$$

$$\hat{x}(k) = w(k) + K_o y_m(k)$$ (4.41)
where \( \mathbf{w}(k) \) is the observer state vector, \( \mathbf{y}_m(k) \) is the measurement vector, \( \hat{\mathbf{x}}(k) = [\hat{\mathbf{q}}^T, \hat{\mathbf{q}}_m^T]^T \) is the estimated states (velocities) with dimension \( m = 2n \) and the matrix \( P \) is a permutation matrix which satisfies

\[
\bar{C} = CP^{-1} = \begin{bmatrix} I_m & 0 \end{bmatrix}_{m \times 2n}.
\]

(4.42)

The first step toward computing the observer matrices \( (K_o, E, H_o, G_o) \) is to do the similarity transformation on \( (A, B) \) using the permutation matrix \( P \) in order to satisfy the format of the measurements to be \( \bar{C} = \begin{bmatrix} I_m & 0_{m \times 2n} \end{bmatrix} \). Subsequently the state space can be partitioned as

\[
\bar{A} = P\tilde{A}P^{-1} = \begin{bmatrix} \bar{A}_{11} & \bar{A}_{12} \\ \bar{A}_{21} & \bar{A}_{22} \end{bmatrix}, \quad \bar{B} = PB = \begin{bmatrix} \bar{B}_1 \\ \bar{B}_2 \end{bmatrix}.
\]

(4.43)

The eigenvalues of \( E = (\bar{A}_{22} - K_o\bar{A}_{12}) \) are the observer poles which can be assigned using pole placement or LQR algorithm depending on the numerical robustness of each place or \( dlqr \) algorithm in Matlab. Therefore, the result of pole placement determines the gain \( K_o \) which in turn determines \( E \).

The remaining matrices \( H_o \) and \( G_o \) are then computed as

\[
H_o = \bar{B}_2 - K_o\bar{B}_1 \quad \text{and} \quad G_o = EK_o + \bar{A}_{21} - K_o\bar{A}_{11}.
\]

Further details on derivations of the matrices are provided in [192]. The relative stability of the observer based LQR feedback is discussed in section 4.5.4.

4.5.3 Unknown Input Observer

In this section, the theory behind an observer for linear systems with unknown inputs proposed in [191] is briefly described. This observer was implemented on CoMan. The motivation of implementing this observer on the robot was to provide a more robust control system in face of unknown stiction effects in the joints. Design of this type of observer was also motivated by the challenges that were faced during the experiments as explained in section 8.9.4. The observer equations are similar to the Luenberger observer with \( H_o = 0 \),

\[
\begin{align*}
\mathbf{w}(k+1) &= E\mathbf{w}(k) + G_o\mathbf{y}_m(k) \\
\hat{\mathbf{x}}(k) &= \mathbf{w}(k) + K_o\mathbf{y}_m(k).
\end{align*}
\]

(4.44)
The key idea behind this design is to choose the gain \( K_o \) such that \( H_o = \bar{B}_2 - K_o \bar{B}_1 = 0 \) which makes the observer estimates independent of the control input signal. The existence conditions of such a design were reported in [191] as

1. The number of measurements \( (2n) \) must be larger than the number of inputs \( (n) \).

2. Rank \( CB = \text{Rank} B = \text{Rank} \bar{B}_1 \).

3. The system must not have any transmission zeros, i.e.

\[
\text{Rank} \begin{pmatrix} zI - \bar{A} \bar{B} \\ C & 0 \end{pmatrix} = 5n
\]

(4.45)

where \( z \) is the discrete time frequency variable.

Provided all the three existence conditions are satisfied, the design can be carried out. Assuming that the similarity transformation described in (4.43) is performed, let

\[
\bar{B}_1^+ = (\bar{B}_1^T \bar{B}_1)^{-1} \bar{B}_1^T 
\]

(4.46)

\[
\mathcal{A}_{22} = \bar{A}_{22} - \bar{B}_2 \bar{B}_1^+ \bar{A}_{12} 
\]

(4.47)

\[
\mathcal{A}_{12} = (I_m - \bar{B}_1 \bar{B}_1^+) \bar{A}_{12}.
\]

(4.48)

The multivariable pole placement for the pair \( (\mathcal{A}_{22}, \mathcal{A}_{12}) \) can be performed to derive an intermediate gain \( K_1 \). Once \( K_1 \) is computed,

\[
E = \mathcal{A}_{22} - K_1 \mathcal{A}_{12}
\]

(4.49)

and

\[
K_o = \bar{B}_2 \bar{B}_1^+ K_1 (I_m - \bar{B}_1 \bar{B}_1^+).
\]

(4.50)

The remaining matrix is \( G_o = EK_o + \bar{A}_{21} - K_o \bar{A}_{11} \) and the norm of \( H_o \) should be verified numerically to be zero. The experimental results of this implementation are discussed in section 8.3.4. However, the experiments using the unknown input observer did not provide an improvement on the velocity estimation (in fact, oscillations were increased).
The experimental results of this implementation are discussed in section 8.3. In the next section the formulation of relative stability is provided.

### 4.5.4 Relative Stability

The observer pole placement must be considered in parallel with relative stability. A fast observer does not necessarily improves the performance of the control system since it affects the stability margins. Therefore, the relative stability must be checked before implementation to have enough robustness to deal with disturbances.

Consider the diagram shown in Fig. 4.20 with the loop broken at the input of the plant as shown with mark “x”. In order to investigate the stability margins of the feedback with the observer the determinant of the frequency response of $F_o = I + G(z)G_c(z)$ at the output of the plant is plotted where $G(z) = C(zI - A)^{-1}B$ is the plant’s discrete time transfer function and $G_c(z)$ is the controllers transfer function which is computed in two cases. In the first case the controller is considered without the observer which is the constant LQR feedback gain. In the second case the observer’s dynamics are included with the LQR feedback gain. The overall controller state space model for case 2 is defined by $(A_c, B_c, C_c, D_c)$, where

\[
A_c = \begin{pmatrix}
E - HF_{L_2} & -HF_3 \\
0 & I_{10}
\end{pmatrix}, \quad B_c = \begin{pmatrix}
G_o - HF_{L_1} \\
[0, I_{10}]
\end{pmatrix}, \quad C_c = -[F_{L_2}, F_3], \quad D_c = -F_{L_1}
\]

(4.51)

and $F_{L_1} = [F_1, F_2] L_1$, $F_{L_2} = [F_1, F_2] L_2$, $F_3$ is the integrators feedback gain, $L_1 = [I_{20}, K_o^T]^T$ and $L_2 = [0, I_{20}]^T$. The symbol 0 in Equation (8.2.2) and the corresponding definitions refers to a zero matrix of suitable dimension. Therefore, in the second case the controllers transfer function is $G_c(z) = C_c(zI - A_c)^{-1}B_c + D_c$. The relative stability of different schemes are plotted in Fig. 4.21.
Figure 4.21: The determinant of the return difference at the input of the plant as a measure of relative stability margins among different observers.

The solid line corresponds to the relative stability of the plot without any observer (merely LQR state feedback) and also the unknown input observer with $H_0 = 0$. Because of this reason the unknown input observer was implemented on the robot to improve the performance but the results were oscillatory. The dashed line in Fig. 4.21 correspond to the fast Luenberger observer which has some degradation from the pure LQR state feedback. The dashed-dotted line correspond to the slow Luenberger observer which indicates a slightly better margins compared to the fast observer. Both these observers were implemented on the robot and the slow observer provided the best results. Experiments with the fast observer resulted in an oscillatory response. This suggests that the oscillation issue seen in the experiments (explained in chapter 8) is mainly due to amplification of noise (quantization) rather than issues with relative stability. Because the unknown input observer has the same stability margins as the LQR state feedback but it tends to behave as a differentiator which amplifies the noise.
4.6 Conclusions

In this chapter, three novel state space feedback design methods were proposed for the stabilization and joint tracking of CoMan. The most common method used in many humanoid robots is the independent PID joint control that treats each link as an independent system and ignores the coupling between the links. A multivariable LQR optimal control method was proposed for joint tracking. This method is a centralized feedback method that has been successfully implemented on CoMan.

In addition, two decentralized feedback design methods based on sparse, gradient descent method and LQR based LMI method were proposed to derive the decentralized feedback gain from the LQR optimal solution. The advantage of these methods is that the derived controller gains can then be directly implemented in the local DSPs at each joint.

Finally, the formulations of the reduced order observers were presented which has been applied to the robot to estimate the motors and joints velocities.
Chapter 5

Double Support Geometric Control

This chapter studies the problem of controlling a compliant robot in double support phase. The motion of a humanoid robot in double support is subject to geometric constraints which result from the fact that the motion in double support requires both feet to be on the ground. Therefore, a controller can be designed to dynamically maintain these constraints while the robot is tracking a reference trajectory that is consistent with the constraints. The main challenges of this work are due to lack of controllability of the double support model and the presence of under-actuated DoF due to the compliance. However, the single support model of the robot can be used to design a constrained feedback controller. The main advantage of this approach is benefiting from the controllability of the single support model for the feedback design. In addition, this method provides a full mathematical proof of the closed loop stability.

The structure of this chapter is as following. Initially, in section 5.1 the background of this problem is reviewed where the gaps in the literature are pointed out. This leads to the contributions of this chapter which are summarized in section 5.2. In section 5.3, the formal definition of the dynamical system and the constraint subspace as well several standard geometric definitions are provided. In section 5.4 key points of the method are presented in several theorems. An algorithm to compute the linear feedback is presented under the assumption that the geometric constraint subspaces are computed beforehand. In section 5.5, the non-trivial computation
for the basis of the constraint subspace is discussed afterwards in two cases. In the first case, the rigid mechanical model is considered and for the purpose of reference tracking, additional integrator dynamics are added to the mechanical model. In the second case the compliant electro-mechanical model with integrators is considered. Finally, in section 5.6, the numerical results of the 10 DoF compliant model of CoMan under the constrained feedback control are presented followed by a discussion about the future work.

5.1 Background

The control of bipedal robots under constraints is a challenging control problem. In an early attempt, a reduced system was derived by solving for the lagrange multipliers and using pole placement to design the feedback gains. However, it was reported that unsuitable assignment for pole locations can lead to violation of the constraints [193]. In addition, other methods partitioned the constrained dynamic model into dependent and independent coordinates and derived a reduced model [194]. The demonstrated results were in 2D (a lateral sway motion) and separate feedbacks were designed for the independent and dependent variables. The issue of choosing pole location was also addressed via linear optimal control. However, the proposed method did not provide an overall guarantee for the stability of the full order linear system where the two separately designed feedback gains are used.

Others have approached this problem from a geometric control perspective with an overall proof of stability for the full order system [195]. The constraint forces were derived as explicit functions of the state and the input. The general model can represent the constrained case whenever the constraint forces are nonzero and the unconstrained case when the constraint forces are zero. Algorithms were proposed for the computation of the feedback gains in both the constrained and the unconstrained cases based on the linearized model for the planar bipedal robots. An interesting point about this method is that the original dimensionality of the system is retained in both the constrained and the unconstrained cases and the robot can deliberately
break or hold the constraints. Further developments on this work were reported [196, 197]. Nevertheless this method does not cater for the dynamics associated with compliant actuators.

There is a gap in the literature for feedback control designs that explicitly take the dynamics of compliant actuation and the constrained motion into account. Hence, the main purpose of this section is to formulate a linear control design methodology based on the theory of \((A, B)\) invariant subspaces for the constrained motion of a compliant humanoid robot in double support. The dynamic model used for the control design is the linearized single support phase model given in Equation (3.11) in continuous time with actuator dynamics and integral reference tracking.

### 5.2 Contribution

The contributions of this work are as follows. A geometric control design method for planar models [195] is extended for the general constrained 3D models of compliant humanoids. A step-wise algorithm is proposed to facilitate the feedback design process using matrix calculations. Then, the geometric constraint subspace required for the controller design is studied in the cases of rigid and compliant robots. In case of compliant robots a nontrivial difficulty arises in design of constraint subspace that is addressed in this section and some physical insights are provided. All formulations are presented in continuous time.

The novelties of this section are two fold. Firstly, Algorithm 5.1 is provided for 3D compliant robots that given the state space model and the suitable constraint subspace, computes the constrained feedback. Secondly, a non trivial solution is provided to compute the constraint subspace for compliant robots.

### 5.3 Preliminary Definitions

In this section, the idea behind control of a humanoid robot in double support phase using the geometric control theory is explained. Consider a humanoid robot that is
moving its upper body to different directions while standing with both feet on the ground. The geometric constraints of maintaining both feet on the ground, boils down to three constraints regarding the feet separation in X, Y and Z directions (i.e. $h_X, h_Y, h_Z$) and two constraints regarding keeping the feet parallel to the ground in lateral and sagittal planes, as shown in Fig. 5.1. The nonlinear dynamics of the robot and the nonlinear constraints can be linearized about the upright posture, as discussed in chapter 3, which provides the setting to apply the geometric control theory to this problem.

![CoMan's Animation](image)

Figure 5.1: Constrained motion of CoMan in the double support phase.

Let the linearized single support model be described by $\Sigma = (X, U, A, B)$ where $X = \mathbb{R}^n$, $U = \mathbb{R}^m$, $A : X \to X$ and $B : U \to X$ are the state set, the input set and linear transformations on these sets, respectively. The linear dynamics are

$$\dot{x} = Ax + Bu \quad (5.1)$$

The system $\Sigma$ is subject to a set of linear constraints represented by $Hx = 0$. The basis for the constraint’s subspace $V$ can then be computed as $V = \text{Ker } H$. The linearized phase portrait of the constrained and unconstrained dynamics are visualized
in Fig. 5.2. This figure is a conceptual figure, since the actual dimension of CoMan’s state space is 50 while the constrained subspace has a dimension of 30. Nevertheless, Fig. 5.2 serves the purpose of illustrating the concept of this method on a 3 dimensional phase portrait. The key idea is that the double support motion can be represented by the constraint’s subspace $V$, which is of a lower dimension. The main aim is to design a feedback such that if the initial condition is within $V$, the robot can continue to move while satisfying the constraints.

Figure 5.2: A conceptual illustration of the constrained dynamics flow in the double support phase.

The standard definitions and theorems that are required for the controller design are presented as follows.

**Definition 1. $(A, B)$- Invariant Subspaces**

Consider the dynamical system $\Sigma$. A subspace $\mathcal{V} \subseteq \mathcal{X}$ is $(A, B)$- invariant if there exist a map $K : \mathcal{X} \to \mathcal{U}$ such that $(A - BK)\mathcal{V} \subseteq \mathcal{X}$.

**Definition 2. $A$- Invariant Subspaces**

Consider the dynamical system $\Sigma$ and let $\mathcal{V} \subseteq \mathcal{X}$ have the property $A \mathcal{V} \subseteq \mathcal{V}$, then $\mathcal{V}$ is said to be $A$-invariant.
Any $A$-invariant subspace is also $(A, B)$-invariant that can be shown by simply choosing the feedback $K$ in Definition 1 to be zero.

**Definition 3. Equivalence Relation**

Vectors $x, y \in \mathcal{X}$ are equivalent if $(x - y \in \mathcal{V})$ and this is denoted by $\sim$ as $(x \sim y)$.

**Definition 4. Factor Space**

The factor space (or quotient space) $\mathcal{X}/\mathcal{V}$ is defined as the set of all equivalence classes

$$\bar{x} := \{y : y \in \mathcal{X}, y - x \in \mathcal{V}\}, \ x \in \mathcal{X}$$

**Definition 5. $\mathcal{V}$-constrained Feedback**

A feedback map $K : \mathcal{X} \to \mathcal{U}$ is a $\mathcal{V}$-constrained feedback if $(A - BK)\mathcal{V} \subseteq \mathcal{V}$.

**Lemma 1.** Suppose $\mathcal{V}$ is an $(A, B)$-invariant subspace and $K$ is chosen according to the Definition 1. If $x(0) \in \mathcal{V}$ then $x(t) = e^{t(A-BK)}x_0 \in \mathcal{V}$ for all $t$, [198].

**Lemma 2.** The set of all $\mathcal{V}$-constrained feedbacks is nonempty if and only if $A(\mathcal{V}) \subseteq \mathcal{V} + \text{Im } B$, [195].

Lemma 2 simply means that subspace of $A(\mathcal{V})$ must be contained in the sum of two subspaces, namely $\mathcal{V}$ itself and $\text{Im } B$. That is there exists input vectors $u$ in the input set $\mathcal{U}$ that can be used to confine $A(\mathcal{V})$ to the subspace $\mathcal{V}$.

**Theorem 1.** Let $V$ be a basis for the subspace $\mathcal{V}$. $V$ is $A$-invariant if and only if there exists a matrix $E$ such that $AV = VE$.

In other words, $A(\mathcal{V}) \subseteq \mathcal{V}$ can be formulated in terms of A-invariance as rank test $\text{Rank } [V] = \text{Rank } [V \ AV]$. The result of this theorem is well-known and the proof can be found in [199].

**Theorem 2.** Consider the linear system $\Sigma$ and the $A$-invariant, constrained subspace $\mathcal{V} \subseteq \mathcal{X}$. There exist a similarity transformation $T$ such that

$$A_T = T^{-1}AT = \begin{bmatrix} A_{11} & A_{12} \\ 0 & A_{22} \end{bmatrix} \quad \text{and} \quad B_T = T^{-1}B = \begin{bmatrix} B_1 \\ B_2 \end{bmatrix} \quad (5.2)$$

where $\text{dim } \mathcal{V} = s$, $A_{11}$ is matrix of size $s \times s$ and corresponds to the dynamics on $\mathcal{V}$. 
Proof:

Consider the transformation $T = [V \ T_2]$ where $T_2$ can be chosen arbitrarily such that $T$ is invertible. It is clear that $T^{-1}V = \begin{bmatrix} I_s \\ 0 \end{bmatrix}$ where $I_s$ is the identity matrix of size $s = \text{Rank} \ [V]$.

In addition, from Theorem 1, the following relations hold

$$AV = VE, \quad \rightarrow \quad T^{-1}ATT^{-1}V = T^{-1}VE, \quad \rightarrow \quad T^{-1}AT \begin{bmatrix} I_s \\ 0 \end{bmatrix} = \begin{bmatrix} E \\ 0 \end{bmatrix},$$

hence $T^{-1}AT$ must have the structure, [199].

\[ \blacksquare \]

### 5.4 Constrained Control Design

In this section, the formulation of the linear feedback is presented that is based on the classic theory of $(A, B)$ invariant subspaces [195, 198, 199].

Consider the system $\Sigma$ with linear dynamics given in Equation (5.1). $\Sigma$ is derived from the the open chain model that is fully controllable, where $x = (q, \dot{q}, q_m, \dot{q}_m, z)^T$ is the state of the system that evolves on $\mathcal{X}$.

The double support geometric constraints $h(q) = 0$, given in Equation (3.3), is used to derive the linear constraints on $x$, that is represented as $Hx = 0$. The detailed computations for deriving $H$ are presented in section 5.5. The main results of this section is presented in Theorem 3 and Algorithm 5.1.

Consider the system described by Equation (5.1). In most cases the subspace $\mathcal{V}$ is not $A$-invariant and a feedback $K_0$ is required to yield a closed loop system that is $(A, B)$-invariant. This feedback is proposed in Theorem 3.

**Theorem 3.** Consider the dynamical system described by $\Sigma = (\mathcal{X}, \mathcal{U}, A, B)$ where $(A, B)$ are controllable. Let $\mathcal{V} \subseteq \mathcal{X}$ be the constraint subspace and let $W = \text{Ker} \ V$ where the columns of $V$ provide the basis for $\mathcal{V}$. If $\mathcal{V}$ is not $A$-invariant, then the feedback $K_0 = (W^TB)^+W^TAVV^+$ satisfies $(A - BK_0) \subseteq \mathcal{V}$ if and only if $K_0$ is an exact solution to $(A - BK_0)V = VE$ and $K_0$ is called a $\mathcal{V}$-constrained feedback.
CHAPTER 5. DOUBLE SUPPORT GEOMETRIC CONTROL

Proof:

Based on the result of Theorem 1, the feedback $K_0$ is obtained as

$$(A - BK_0)V = VE \quad (5.3)$$

Let $W^T V = 0$ where $W = \text{Ker} \ V$ is the orthogonal complement of $V$. Then, multiplying Equation (5.3) by $W^T$ and solving for $K_0$ gives

$$W^T AV - W^T BK_0 V = W^T VE = 0$$

Let $W^T AV - W^T B \tilde{K} = 0$ where $\tilde{K} = K_0 V$, then $\tilde{K} = (W^T B)^+ W^T AV$ and

$$K_0 = (W^T B)^+ W^T AV \ V^+$$

where the superscript $+$ denotes the pseudo inverse of a matrix.

■

From the numerical aspect, it is important to verify that $K_0$ is an exact solution otherwise the condition $(A - BK_0) \subseteq V$ is not satisfied.

5.4.1 Control Design Method

Initially, it must be ensured that the condition of Lemma 2 holds and the set of $V$-constrained feedback is nonempty. The condition of Lemma 2 is translated to a matrix rank test as

$$\text{Rank} \begin{bmatrix} B & V \end{bmatrix} = \text{Rank} \begin{bmatrix} B & V & AV \end{bmatrix} . \quad (5.4)$$

The matrix rank test in Equation (5.5) is used to verify whether the constrained subspace $V$ is $A$-invariant,

$$\text{Rank} \begin{bmatrix} V \end{bmatrix} = \text{Rank} \begin{bmatrix} V & AV \end{bmatrix} . \quad (5.5)$$

In most cases the subspace $V$ is not $A$-invariant, and a state feedback $K_0$ must be obtained to provide an $(A, B)$ invariant subspace. According to Theorem 3, an initial $V$-constrained feedback, $K_0$ is computed and $A$ is replaced by $(A - BK_0)$. A second stage feedback is used for pole placement.
Nevertheless an arbitrary linear feedback that corresponds to a desired pole location is not necessarily a $\mathcal{V}$-constrained feedback. This issue is resolved by partitioning the state space into the constraint subspace dynamics and its factor space [195].

Fig. 5.3 illustrates the partitioning of the matrix $A$ with a similarity transformation $T = [V \ V^\perp]$ into the constrained subspace and its orthogonal complement. It is also possible to choose any arbitrary direction such as $W$ for the similarity transformation $T = [V \ W]$ such that $T$ is invertible. However, considering the high dimension of the state space model (50 dimensional), it is more convenient to choose the orthogonal complement direction $V^\perp$ for the partitioning. This idea is depicted in $\mathbb{R}^2$ in Fig. 5.3. Moreover, it is shown that the four partitions created have the structure given in Theorem 2, where the maps $A_{VV} : \mathcal{V} \to \mathcal{V}$, $A_{V^\perp V} : \mathcal{V}^\perp \to \mathcal{V}$, $A_{VV^\perp} : \mathcal{V} \to \mathcal{V}^\perp$ and $A_{V^\perp V^\perp} : \mathcal{V}^\perp \to \mathcal{V}^\perp$ define the dynamics in the partitioned state space $\mathcal{X}$. According to Theorem 2 and Definition 2, the map $A_{VV^\perp}$ is zero by definition since no trajectory leaves the $A$-invariant constrained subspace $\mathcal{V}$.

In other words, let $B_{V} : \mathcal{U}_{V} \to \mathcal{V}$ and $A_{VV} : \mathcal{V} \to \mathcal{V}$ where $\mathcal{U}_{V} = \{u : Bu \in \mathcal{V}\}$, $Im\ B_{V}$ is a restricted $Im\ B$ to the set $\mathcal{U}_{V}$ and $A_{VV}$ is the restricted dynamics of $\Sigma$ to $\mathcal{V}$. In a similar way, let the sets $\mathcal{U}_{V^\perp}$ and $A_{V^\perp V}$ be defined on the factor space $\mathcal{X}/\mathcal{V}$ as $\mathcal{U}_{V^\perp} = \{u : Bu \in \mathcal{X}/\mathcal{V}\}$ and $A_{V^\perp V} : \mathcal{X}/\mathcal{V} \to \mathcal{X}/\mathcal{V}$. Then pole assignment can be attempted independently in both subspaces via a feedback $K_{V} : \mathcal{V} \to \mathcal{U}_{V}$ and a feedback $K_{V^\perp} : \mathcal{X}/\mathcal{V} \to \mathcal{U}_{V^\perp}$ provided the pair $(A_{VV}, B_{V})$ is controllable (as a result the pair $(A_{V^\perp V^\perp}, B_{V^\perp})$ will also be controllable). The original system is partitioned
into the two subspaces by a linear transformation.

It should be noted that for an input vector $\mathbf{u}$ in $\mathcal{U}$, $B_{\mathcal{V}^\perp}\mathcal{U} = 0$ if and only if $Bu$ lies in the constraint subspace $\mathcal{V}$. This set of inputs $\mathcal{U}_\mathcal{V}$, is spanned by $M$ where

$$M = \text{Ker}B_{\mathcal{V}^\perp}$$

hence,

$$B^T M = \begin{bmatrix} \hat{B}_V M \\ 0 \end{bmatrix} = \begin{bmatrix} B_V \\ 0 \end{bmatrix}. \quad (5.7)$$

LQR optimal gains are preferred over pole placement methods due to their optimality and numerical robustness. Suppose the LQR feedback gain for this subsystem is $K_V$ then the other subsystem $(A_{V^\perp V}, B_{V^\perp})$ can be used to design the feedback $K_{V^\perp}$ outside $V$. Hence, the overall feedback gain for the full system would be $K_t = [MK_V \ K_{V^\perp}]$ and the closed loop system under the feedback $K_t$ is given by

$$A - BK_t = \begin{bmatrix} A_{VV} & A_{V^\perp V} \\ 0 & A_{V^\perp V^\perp} \end{bmatrix} - \begin{bmatrix} \hat{B}_V \\ \hat{B}_{V^\perp} \end{bmatrix} \begin{bmatrix} MK_V \\ K_{V^\perp} \end{bmatrix} \quad (5.8)$$

$$= \begin{bmatrix} A_{VV} - B_V K_V & A_{V^\perp V} - \hat{B}_V K_{V^\perp} \\ 0 & A_{V^\perp V^\perp} - B_{V^\perp} K_{V^\perp} \end{bmatrix}.$$ 

Since the feedback $K_t$ is designed for the transformed system $(A_T, B_T)$ the transformation must be applied to the feedback to transform $K_t$ to the original coordinates $(A, B)$. The gain for the original system is $K = K_0 + K_t T^{-1}$.

Algorithm 5.1 summarizes these ideas into a computational algorithm in terms of matrix operations.
Algorithm 5.1: Constrained feedback computation

This algorithm takes the state space matrices of the single support model, \((A, B)\) and the constrained subspace basis \(V\) to compute the \(V\)-constrained feedback \(K\).

1. Ensure the computation is feasible by the rank test in Equation (5.4).

2. Verify \(V\) is \(A\)-invariant by rank test in Equation (5.5). If the condition is satisfied go to step 4, otherwise proceed to step 3.

3. Compute a preliminary feedback \(K_0\) as in Theorem 3. Replace \(A\) by \((A - BK_0)\) and proceed.

\[
K_0 = (W^T B)^+ W^T A V V^+ \tag{5.9}
\]

4. Partition the state space system \(\Sigma\) by \(T = [V \ \text{Ker} V]\) and derive the transformed system as

\[
A_T = T^{-1} A T = \begin{bmatrix} A_{VV} & A_{V\perp V} \\ 0 & A_{V\perp V\perp} \end{bmatrix} \quad \text{and} \quad B_T = T^{-1} B = \begin{bmatrix} \hat{B}_V \\ B_{V\perp} \end{bmatrix}. \tag{5.10}
\]

5. Compute the matrix \(B_V = \hat{B}_V M\) where \(M = \text{Ker} B_{V\perp}\) and proceed.

6. Compute an LQR feedback \(K_V\) for the constraint subspace represented by the pair \((A_{VV}, B_V)\) and an LQR feedback \(K_{V\perp}\) for the factor space represented by the pair \((A_{V\perp V\perp}, B_{V\perp})\) and proceed.

7. Compute the total feedback in the original coordinates as

\[
K = K_0 + [MK_V \ K_{V\perp}] T^{-1}.
\]

It should be noted that the pseudo inverse operator used in Algorithm 5.1 computes a least square solution to the linear system and for this reason the computed \(K_0\) must be checked numerically to be an exact solution. In this thesis the Moore-Penrose pseudo inverse is used which exists and is unique for any matrix \(A\), where \(A^+\) satisfies
the following four properties \[200]:

1. \(AA^+ A = A\), 2. \(A^+ AA^+ = A^+\), 3. \((AA^+)^* = AA^+\), 4. \((A^+ A)^* = A^+ A\).

The control design algorithm described above assumes that the constraint subspace is calculated beforehand. However, when the compliant actuator dynamics are included in the model, the form of this constraint subspace is non-trivial. The next section discusses the different forms that this subspace takes as additional dynamics are gradually added to the system’s dynamics.

### 5.5 The Geometric Constrained Subspace

The geometric constraint subspace takes different forms depending on the definition of the state space equations. In other words, depending on the type of dynamics present in the linear model, different constraints must be considered for the additional dynamic variables. Two rigid and compliant cases are presented in this section. In both cases the linear constraints on the state variable vector is presented as an equality constraint

\[
H \mathbf{x} = 0 \tag{5.11}
\]

where \(\mathbf{x}\) has different dynamic variables present depending on each case and the constrained subspace is spanned by the columns of \(V = \text{Ker} \ H\). As mentioned in the control design Algorithm 5.1, the condition in Equation (5.4) must hold in both cases. Moreover, since the tracking problem is considered in this section, the integrator dynamics are included in the constrained subspace computation which is presented next.

#### 5.5.1 The Constrained Subspace for the Rigid Model

The geometric constraints of the robot are expressed by a nonlinear function \(h(q) = 0\) can be linearized about the operating point to provide linear equations of constraint, where \(J_c(q)\) is obtained by considering the incremental motion on the surface of the
constraints $h(q)$ and $J_c = J_c(q_0)$ where $q_0$ is the operating point.

$$J_c q = 0$$  \hspace{1cm} (5.12)

It should be noted that the two constraints of keeping the feet parallel to the ground are already included in the Jacobian $J_c$. These two constraints are derived by setting the sum of the angles in lateral and sagittal plane equal to zero. Moreover, a constraint on the position, as expressed in Equation (5.12), imposes a constraint on the velocity as well. In this case $H$ has the following structure

$$Hx = \begin{bmatrix} J_c & 0 \\ 0 & J_c \end{bmatrix} x = 0$$ \hspace{1cm} (5.13)

where $x = (q, \dot{q})^T$. This case is the simplest form of constraints.

In presence of integrators a constraint must be imposed on the integrator incremental variable $z$. This is due to the fact that $\dot{z} = r - q$, where $r$ is defined to be consistent with the constraint and $q$ is constrained as in Equation (5.12). Hence, $H$ has the following structure

$$Hx = \begin{bmatrix} J_c & 0 & 0 \\ 0 & J_c & 0 \\ 0 & 0 & J_c \end{bmatrix} x = 0$$ \hspace{1cm} (5.14)

where $x = (q, \dot{q}, z)^T$. The addition of the non-ideal actuators with finite stiffness and damping requires a more elaborate formulation of the constraints which is considered in the following section.

### 5.5.2 The Constrained Subspace for the Compliant Model

In the case of having compliant actuator dynamics an interesting issue arises on structure of $H$. In fact, the structure of $H$ becomes non-trivial and a suitable method of obtaining invariant subspaces must be investigated.

In the practical implementation of a feedback controller, the viscous friction and finite stiffness must be taken into account. CoMan’s mechanical joints that are coupled to compact electric actuators are rather difficult to back drive. Hence, all of the
joints have to be driven appropriately to ensure that the geometric constraints are satisfied. In other words, the inclusion of actuator dynamics and compliant elements are inevitable. However, an interesting problem arises when one tries to design $\mathcal{V}$-constrained feedbacks for the overall model. The rank test in Equation (5.4) does not hold if $H$ is simply chosen as $H = \text{diag}\{J_c, J_c, J_c, J_c\}$. This leads to a physical insight into the relationship between the actuators variables $q_m, \dot{q}_m$ and the joint variables $q, \dot{q}$. In this thesis, a general solution is derived for this problem when the joint accelerations are constrained as well as the joint positions and joint velocities. Since the joint accelerations, $\ddot{q}$, do not have a direct input but instead they are driven indirectly via the coupling to actuators, they must also satisfy the linearized constraints equation, i.e. $J_c \ddot{q} = 0$. Hence,

$$J_c \ddot{q} = J_c \begin{bmatrix} -M_L^{-1}(G + K_s) & -M_L^{-1}(C + B_s) & M_L^{-1}K_s & M_L^{-1}B_s \end{bmatrix} x = 0 \quad (5.15)$$

where $x = (q, \dot{q}, q_m, \dot{q}_m)^T$. In addition, it is worth noting that since the joint positions $q$ and velocities $\dot{q}$ have to satisfy $J_c q = 0$ and $J_c \dot{q} = 0$, the vectors $q$ and $\dot{q}$ have to be projected into the null space of the Jacobian $J_c$

$$P_J = I - J_c^T (J_c J_c^T)^{-1} J_c \quad (5.16)$$

where $I$ is an identity matrix of the appropriate size. Therefore, the constraints on the joints’ accelerations can be defined as

$$J_c \ddot{q} = J_c \begin{bmatrix} -M_L^{-1}(G + K_s)P_J & -M_L^{-1}(C + B_s)P_J & M_L^{-1}K_s & M_L^{-1}B_s \end{bmatrix} x = 0 \quad (5.17)$$

hence $Hx$ has the following structure

$$\begin{bmatrix} J_c & 0 & 0 & 0 \\ 0 & J_c & 0 & 0 \\ -J_c M_L^{-1}(G + K_s)P_J & -J_c M_L^{-1}(C + B_s)P_J & J_c M_L^{-1}K_s & J_c M_L^{-1}B_s \end{bmatrix} x \quad (5.18)$$

This non-trivial equality constraint is consistent with the physical relationship between the actuators and the joints that are coupled by springs and dampers. This coupling results in the control input entering the system at the actuator acceleration and as a result the joints’ accelerations can only be changed indirectly by the coupling
torque between the actuators and the joints. This issue is not present in the rigid case, due to the reason that the control input enters the dynamics on the acceleration terms of the joints.

In addition, when integral action is considered, the integrators incremental value must also satisfy the $J_c z = 0$ relationship. Hence, the linear equality constraints $H x$ in this case has the following structure

$$
\begin{bmatrix}
J_c & 0 & 0 & 0 & 0 \\
0 & J_c & 0 & 0 & 0 \\
-J_c M^{-1}_L (G + K_s) P_J & -J_c M^{-1}_L (C + B_s) P_J & J_c M^{-1}_L K_s & J_c M^{-1}_L B_s & 0 \\
0 & 0 & 0 & 0 & J_c
\end{bmatrix} x
$$

where $x = (q, \dot{q}, q_m, \dot{q}_m, z)^T$. Hence, having computed the correct constrained subspace, the rank condition in Equation (5.4) is satisfied and the constrained feedback design can be carried out.

The next section presents a numerical simulation of CoMan with 10 DoF to demonstrate the results of applying a constrained feedback to the single support (open chain) and double support (closed chain) models.

### 5.6 Simulation of Double Support Control of the Complaint Model of CoMan

In this section, a 10 DoF linear model of CoMan with compliance and actuator dynamics in single support given by Equation (5.1) is used to design the constraint feedback based on the algorithm presented in section 5.4 and the corresponding linear double support model given by Equation (3.6) is used to investigate the numerical results of applying the feedback to the constrained model. The model parameters including DC motors, gearboxes (harmonic drives), compliance and the bodies’ mechanical parameters such as mass, CoM and inertia are given in Appendix C.

The main purpose of this simulation is to demonstrate that the robot can track a desired reference trajectory in 3D while maintaining the constraints during the entire
dynamic response. Furthermore, this simulation does not consider the full walking cycle that consists of single support, heel strike, double support and push-off at this stage and it is simply meant to illustrate the double support dynamic motion. This section begins with the formulation of the reference tracking control problem. The reference angles used in the simulation corresponds to a lateral sway, while bending the knees, the ankles and the hips in the sagittal plane. Then the frequency response of the determinant of the return difference at the input of the plant is used as a rough qualitative measure on the relative stability. Finally, some remarks on the relation between use of ankle torque and robustness are presented.

5.6.1 Controller Design

The linear model in Equation (5.1) is derived by using the model parameters given in Appendix C and introducing 10 integrators on the joint positions that are of interest to control. The pair \((A, B)\) are fully controllable. Hence, the state vector \(x = (q, \dot{q}, q_m, \dot{q}_m, z)^T\) is a 50 \(\times\) 1 vector where the size of each of its components is 10. The constraints' Jacobian \(J_c\) is

\[
\begin{bmatrix}
0, & -0.0603, & -0.2616, & 0, & -0.4882, & -0.4882, & 0, & -0.2616, & -0.0603, & 0 \\
0.0603, & 0, & 0, & 0.4882, & 0, & 0, & 0.4882, & 0, & 0, & 0.0603 \\
0.1642, & 0, & 0, & 0.1642, & 0.0008, & 0.0008, & 0, & 0, & 0, & 0 \\
1.0, & 0, & 0, & 1.0, & 0, & 0, & 1.0, & 0, & 0, & 1.0 \\
0, & 1.0, & 1.0, & 0, & 1.0, & 1.0, & 0, & 1.0, & 1.0, & 0 \\
\end{bmatrix}
\]

where \(J_c\ q = 0\) is the equality constraint on the joint angles and the elements of the joint vector \(q = (q_1, ..., q_{10})\) in this model are ordered as ankle lateral, ankle sagittal, knee, hip lateral, hip sagittal on the right leg and hip sagittal, hip lateral, knee, ankle sagittal and ankle lateral on the left leg, respectively. The first three rows of \(J_c\) represent the XYZ distance between the two feet and the last two rows of \(J_c\) are due to the constraint of keeping the feet parallel to the ground. As it was
mentioned in the previous section, the total equality constraint is derived by applying this constraint on the incremental value of the each component of the state space vector $x$, that is the total equality constraint is computed by deriving the $H$ matrix given by Equation (5.19). Hence, the constraint subspace basis that is spanned by columns of $V$ is derived by computing $V = \text{Ker} \ H$. The size of the matrix $V$ is $50 \times 30$ and hence $s = \text{Rank} \ [V] = 30$. Once $V$ is computed the rank condition stated in step 1 of the algorithm is verified. However, the condition in step 2 does not hold and a preliminary feedback $K_0$ as given by Equation (5.9) in step 3 is computed such that $\text{Rank} \ [V] = \text{Rank} \ [V (A - BK_0)V]$ and $A$ is replaced by $A - BK_0$. In step 4 and 5, the transformation $T = [V \text{Ker} V]$ is computed and its invertibility is verified via calculating its determinant where in this case $\text{det} \ T = 1$. Then the pair $(A, B)$ are partitioned into the constraint subspace and its factor space denoted by the pairs $(A_{VV}, B_V)$ and $(A_{V\perp V\perp}, B_{V\perp})$, respectively.

In step 6 the LQR feedback design is used to place the poles of the two subsystems while minimizing a quadratic cost on the state and input vectors. The LQR penalties used for the feedback design are $Q_0 = \text{diag} \{100I_n, \ 0.01I_n, \ 10I_n, \ 10I_n, \ 3000I_n\}$ where $I_n$ is the identity matrix of size $n$ and $n = 10$ is the number of joints. The penalties are applied to CoMan’s model joint positions, joint velocities, motor positions, motor velocities and the added integrators. This penalties are then transformed to the new coordinates using the transformation $T$ and the first block of $T^TQ_0T$ with the size of $A_{VV}$ ($30 \times 30$) is used for the $(A_{VV}, B_V)$ LQR feedback design and the next diagonal block with the size of $A_{V\perp V\perp}$ ($20 \times 20$) is used for the LQR feedback design of the $(A_{22}, B_{V\perp})$ subsystem. The positive semi definite property of the state penalties are checked and the off diagonal blocks of $T^TQ_0T$ are discarded. The control input penalty $R$ is chosen as identity matrices of suitable size for both subsystems in the transformed coordinates. Matlab’s LQR algorithm is used to compute the feedback gains $K_V$ and $K_{V\perp}$ for the two subsystems. In step 7 the total feedback in the original coordinates is computed as $K = K_0 + [MK_V \ K_{V\perp}]T^{-1}$. 
5.6.2 Simulation Results

Having computed the total feedback $K$, the numerical result of this feedback is illustrated by the following simulation. In the simulation, the closed loop linear double support model given by Equation (3.6) with the addition of compliant actuator dynamics is used to show reference tracking of a desired trajectory. A simple step reference is used for the joint angles as

$$r = (0.1, 0.2, -0.2, -0.1, 0.2, -0.2, 0.1, 0.2, -0.2, -0.1)^T.$$ 

Fig. 4.4 shows the initial and final configuration corresponding to the vector $r$. In Fig. 5.4 the step response of the joint angles in lateral and sagittal planes are demonstrated. As it can be seen from this figure the bandwidth of the system is about 1 Hz. This is one of the important points that to achieve a high bandwidth on the servo control system of a compliant humanoid, explicit models of compliance and actuator inertias must be considered. Otherwise increasing the bandwidth using ad-hoc methods or by using models that ignore the compliance and the actuator dynamics can lead to an unstable system. Fig. 5.5 illustrates the joint velocities in the lateral and sagittal planes. In Figs. 5.6 and 5.7 the corresponding motor joint angles and velocities are shown. It can be seen that motor positions and velocities do not fully resemble the joints positions and velocities as is the case in rigid robots. This is due to passive compliance and damping between the actuators and the joints.

In Fig. 5.8 the control input, that is the voltages sent to each motor is illustrated for the lateral and sagittal planes. The magnitude of the these control voltages are relatively small when compared to the maximum actuator voltage that is 24 volts. This observation, roughly indicates that with the addition of a 10kg upper-body mass the actuators will have enough capacity to maintain the dynamic balance during walking.

In addition, an important property of this feedback is that it maintains the full dynamic response with in the constraint subspace. This is verified by figure 5.9 that illustrates the maximum violation of the nonlinear constraint derived by computing $h(q)$ over the entire system response trajectories. As shown in the simulation, the
Figure 5.4: Joint angles in lateral (left) and sagittal (right) planes in double support.

Figure 5.5: Joint velocities in lateral (left) and sagittal (right) planes in double support.
Figure 5.6: Motor angles in lateral (left) and sagittal (right) planes in double support.

Figure 5.7: Motor velocities in lateral (left) and sagittal (right) planes in double support.
amount of constraint violation for the foot separation is less than $10^{-3}$ which shows the agreement between the feedback and the constraints.

Remark 1: In computation of the null space or kernel two methods can be used. The null function in Matlab uses two methods to obtain the null space of a given matrix. The first method returns an orthonormal basis for the null space using the singular value decomposition and second method returns a rational basis for the null space using the reduced row echelon form which is basically a Gauss Jordan elimination with partial pivoting. However, both methods were numerically ill conditioned for the controller design. Hence, a third method of calculating the null space proposed in [199] was used which is based on QR decomposition which proved to be numerically more efficient.

Remark 2: Regarding the choice for the similarity transformation $T$, the following choice is numerically better conditioned compared to numerical algorithms of Matlab for computation of the null space. $T = [V \ Ker V]$ and this will result in $|det(T)| = 1$ where computation of $Ker V$ is provided by [199] based on QR decomposition.

Remark 3: In design of the LQR feedback gains the penalties can be set on the
The nonlinear constraint violation (meter)

Figure 5.9: The constraints $h(q)$ violation during the dynamic response of the system in X, Y and Z directions.

positions and velocities and integrators that have physical meanings by using the similarity transformation relationship. i.e. if the penalties on the state in the original coordinates are $Q_0$ then the penalties in the new coordinates would be $Q_n = T^TQ_0T$ and the first diagonal block that corresponds to the constraint subspace states and the second diagonal block corresponds to the rest of the states that can be used for the LQR controller design.

5.7 Conclusions

A systematic approach for design of constrained feedback gains for a general 3D robot model was proposed that provides reference tracking while taking into account the constraints, effects of actuator dynamics, compliance (springs) and damping in the robot joints. The approach is given as a step wise algorithm in 5.1.

It was shown that once the assumption of rigid joints is removed, the computing the constraint subspace is non trivial due to the under actuated degrees of freedom associated with the compliance. This issue was studied in detail for the rigid and
compliant model with additional integrator dynamics required for reference tracking. A realistic model of CoMan with 10 DoF and significant compliance in the ankles, knees and hips (in sagittal plane) was simulated to illustrate the application of this approach.

Moreover, the closed loop system is stable for both the single and the double support phases. The closed loop stability of the single support phase is a direct consequence of designing the feedback using the single support model, therefore, by design, the resulting single support closed loop system is stable. For the double support the additional requirement, besides closed loop stability, is that the feedback has to be constrained so that the state evolves within the constrained subspace that is consistent with the constraints $h(q)$. In this case also the reference trajectories have to be consistent with $h(q)$.

The approach presented here can also be applied to a robot with upper body. The only difference is that the upper body is not part of the closed kinematic chain and it is part of the unconstrained subspace. Future work will consider the design of constrained feedbacks in discrete time for the purpose of implementation on CoMan.
Chapter 6

Walking Trajectory Generation

A trajectory generator is a control system that computes the reference trajectory for the robot’s joints. This control system translates the desired walking parameters such as foot locations, step length, walking speed and walking direction into the feasible and stable joints’ trajectories. Ideally, the trajectory generator should work in real-time (also referred to as online) and in closed loop, in order to respond appropriately to disturbances acting on the joints (such as stiction). In addition, the trajectory generator must provide walking trajectories with reasonable torques. Various methods for bipedal walking were reviewed in chapter 2. Zero Moment Point (ZMP) is the most commonly used criteria in bipedal walking which is used in this chapter to propose an online and more robust trajectory generation method.

The contributions of this chapter are two fold. Firstly, an online trajectory generation method based on preview control of the extended cart table model is proposed in section 6.1 that utilizes the upper body for improving walking robustness. The simple inverted pendulum models of walking are useful because of their low dimensions which facilitate the analysis to gain further insight into the problem as well as low computational cost which make them ideal for online implementation. Secondly, new results on the selection of bipedal walking parameters, such as the tracking bandwidth, walking speed and step length are presented in section 6.2. Numerical results are provided using the CoMan’s parameters to identify the limitations which serves as a guidance for selection of feasible walking parameters. Moreover, the torques in
the static case is studied while a step is taken.

6.1 Trajectory Generation Using Simple Models

The key contribution of this section is to formulate the upper body motion in the preview control trajectory design framework, in order to improve the walking robustness. In this section, an overview of the cart-table model and its relation to the constrained linear inverted pendulum are discussed, including analysis of the unstable zero of the model that causes its non-minimum phase characteristic which makes the ZMP stabilization challenging. Then a two mass inverted pendulum that was proposed in [201], is reviewed and its benefits and limitations are mentioned. In sections 6.1.1, 6.1.2 and 6.1.3, a single pendulum, a double pendulum and the extended pendulum cart-table model are discussed, respectively. In section 6.1.4, the idea of formulating walking pattern generation as a servo control problem [202], which can be applied to the formulated inverted pendulum type dynamic models, is briefly reviewed.

6.1.1 Linear Single Inverted Pendulum and the Cart-Table Model

In [203] a three dimensional linear inverted pendulum model, with application to biped walking pattern generation, was proposed. The linear and decoupled dynamics of the 3D inverted pendulum in sagittal and lateral planes are derived by constraining the pendulum’s CoM to move along a plane which simplifies the inverted pendulum equations as shown in Fig. 6.1. In [202] it is shown that the 3D inverted pendulum dynamics directly corresponds to the dynamics of a running cart on a pedestal table. The cart-table dynamics can be expressed as:

\[
\begin{align*}
p_x &= x - \frac{z_c}{g} \ddot{x} \\
p_y &= y - \frac{z_c}{g} \ddot{y}
\end{align*}
\]

where \(p_x\) and \(p_y\) are the zero moment points in the \(X\) and \(Y\) directions, \(z_c\) is the height of the cart and \(g\) is the gravity coefficient. As the formulations of the problem in the
Figure 6.1: In (a) constrained linear inverted pendulum is shown that assumes that ZMP is in the origin \((0,0,0)\) (origin corresponds to the ankle of robot) and torque in the ankle is zero. In (b) the cart table model is shown for Z-X plane where the ZMP can move in the foot (table’s contact with the ground) and the torque in the ankle does not need to be zero.

\(X\) and \(Y\) directions are similar, the problem is studied in the sagittal plane (in the \(X\) direction). In other words, under certain assumptions proposed in [203] the motion in sagittal and lateral planes can be decoupled. Therefore, the models in this section and the proposed model in section 6.1.3 are also valid for the three dimensional case. In order to represent Equation (6.1) in the state space form, a new variable \(u_x\) is defined to be the time derivative of the horizontal acceleration of CoM. That is, \(u_x = \frac{d}{dt} \dddot{x}\), is considered to be the jerk motion of the CoM. Therefore, the state space equations are:

\[
\begin{align*}
\dot{x} &= \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} x + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} u_x, \\
p_x &= \begin{bmatrix} 1 & 0 & \frac{z_c}{g} \end{bmatrix} x
\end{align*}
\]

(6.2)

where \(x = (x, \dot{x}, \dddot{x})^T\). The cart table model provides a convenient framework to directly control the ZMP of a bipedal robot using optimal finite horizon LQR control that is also referred to as preview control in the robotics’ literature, which is discussed in section 6.1.4.

However, a problem arises from the non-minimum phase characteristic of Equation (6.2). The transfer function in Equation (6.2) can be obtained as

\[
G(s) = \frac{p_x(s)}{u_x(s)} = \frac{1 - (\frac{z_c}{g})s^2}{s^3} = \left(\frac{z_c}{g}\right) \frac{\frac{g}{z_c} - s^2}{s^3}.
\]

(6.3)

The transfer function (6.3) is unstable since it has three poles at the origin but this
can be solved using pole placement. However, in Equation (6.3), it is shown that
the transfer function of the cart table model has two zeros. Feedback will not affect
the positions of the zeros and this will cause certain problems in the control of such
systems. The left half plane zero will have a scaling effect on the steady state response
which can be solved by a feed-forward gain, but the right half plane zero will limit the
achievable bandwidth of the system and produces an undesirable undershoot [204].
In other words, consider the case where the desired ZMP is in front of the actual
ZMP of the robot. In order to correct the error between the desired and the actual
ZMP, the hip must accelerate in the positive direction and according to Equation
(6.2) the ZMP will initially move backward and diverges from the desired position
as illustrated in Fig. 6.2. Therefore, the task of the stabilizer, that is discussed in
section 6.1.5, is to minimize this influence or ideally try to avoid it.

In general, perfect tracking control of non-minimum phase systems without future
information of the tracking signal is not possible and one solution is to approximate
the non-minimum phase system with a minimum phase system to design controllers
with bounded tracking error [205]. In addition, design of controllers based on right
half plane pole-zero cancellation is fundamentally flawed, due to the loss of internal
stability. That is such designs rely on unstable pre-filters that can produce excessive
control inputs to the system. Therefore, the performance limitations due to the right
half plane zeros, will be present in any design leading to a trade off between the speed
of response and the amount of undershoot in the system’s step response. In Fig. 6.2
this idea is shown for the cart table model which is stabilized using pole-placement.
In the first case (dotted line) the ZMP rise time is 1.04 sec and ZMP undershoot
is 27% (notice the slow response of center of mass) while in the second case (solid
line) the ZMP rise time is 0.275 sec and ZMP undershoot is 84% (in this case the
center of mass is quickly approaching the steady state). This shows the trade off in
bandwidth and undershoot criteria. The next section, provides a partial solution to
the non-minimum phase problem of cart table model which forms the basis of the
proposed extended cart-table model in section 6.1.3.
CHAPTER 6. WALKING TRAJECTORY GENERATION

6.1.2 Double Inverted Pendulum Model

The non-minimum phase problem of one-mass inverted pendulum model which represents the CoM of a robot was first studied in [201], where a two mass inverted pendulum was proposed to form a multivariable minimum phase system. However, it should be noted that the concept of multivariable zeros are much more complex than the concept of zero in a scalar transfer function. The picture of the two mass inverted pendulum is shown in Fig. 6.3. The idea of using the upper body in ZMP walking is not new and it was first used in Waseda University to improve ZMP walking on Waseda leg called WL-12. However, the method that was used in the robot was based on FFT [28, 29] rather than preview control of ZMP that is proposed in section 6.1.5.
After linearization about the vertical, the following state space equations can be derived, where \( \mathbf{x} = (\theta_1, \theta_2, \dot{\theta}_1, \dot{\theta}_2)^T \):

\[
\dot{\mathbf{x}} = \begin{bmatrix} 0 & I_{2 \times 2} \\ 0 & 0 \end{bmatrix} \mathbf{x} + \begin{bmatrix} 0 \\ I_{2 \times 2} \end{bmatrix} \mathbf{u}, \quad p_x = [c_1 \ c_2 \ 0 \ 0] \mathbf{x} + [d_1 \ d_2] \mathbf{u}
\]

\( c_1 = \frac{(m_1 + m_2)l_1 + m_2l_2}{m_1 + m_2}, \ c_2 = \frac{m_2l_2}{m_1 + m_2}, \ d_1 = -\frac{m_1l_2^2 + m_2(l_1 + l_2)^2}{(m_1 + m_2)g}, \ d_2 = -\frac{m_2(l_1 + l_2)l_2}{(m_1 + m_2)g} \) (6.4)

where \( \mathbf{u} = (u_1, u_2)^T = (\dot{\theta}_1, \dot{\theta}_2)^T \) and \( p_x \) are the input vector and scalar output of the two mass inverted pendulum. It can be shown that system has no multivariable transmission right half plane zeros, provided the second link’s length is non-zero. The transfer function matrix of two mass inverted pendulum can be obtained as following:

\[
G(s) = C(sI - A)^{-1}B + D = \begin{bmatrix} c_1 + d_1s^2 & c_2 + d_2s^2 \\ s^2 & s^2 \end{bmatrix}.
\]  

(6.5)

It should be pointed out that the zeros in multivariable systems have different concept and they are associated with the directions of the input-output of the system. To obtain the zero of Equation (6.5), \( G(s) \) can be written in the following form:

\[
G(s) = \frac{\Pi(s)}{D_G(s)}, \quad \Pi(s) = [c_1 + d_1s^2 \ c_2 + d_2s^2], \quad D_G(s) = s^2.
\]  

(6.6)

The roots of greatest common divisor of \( \Pi(s) \) determines the transmission zeros of the transfer function matrix \( G(s) \), which in this case does not have a root.

This confirms that for a double inverted pendulum the transfer function does not have any transmission zeros, and hence the system is minimum phase. However, as mentioned earlier, this scheme has limitations due to the linearizations of the two masses around the vertical line. Trajectory generation using this model can produce large errors, since the robot’s hip is required to move in a large operating range [206]. In addition, inverted pendulum models assume the ZMP to be in the ankle and ankle torque is zero, while in table cart model ZMP is located in the foot and ankle torque is not necessarily zero. In section 6.1.3, a new model is proposed that combines the advantages of cart table model and two mass inverted pendulum.
6.1.3 Extended Two Link Inverted Pendulum Model

Given the limitations of the two mass inverted pendulum due to the linearization of both pendulum links, the following model, which is a combination of the cart table model and two mass inverted pendulum, is proposed. This model does not assume linearization of the lower mass that corresponds to the robot’s hip motion. However, the upper body angle with respect to the vertical line does not have a large deviation from the linearization point which is acceptable in most cases. It is shown that this model is minimum phase and can be used for trajectory generation and stabilizer design to correct fast deviations in ZMP tracking which results from disturbances.

Consider the proposed pendulum cart table model shown in Fig. 6.4. Writing the equation for torques around the ZMP results in

\[
\tau_{zmp} = -Mg(x - p_x) + M \ddot{x}_c - (mg + m \ddot{z}_2)(x_2 - p_x) + m \ddot{x}_2(z_c + z_2) = 0 \quad (6.7)
\]

where \(z_2 = l \cos(\theta), \ x_2 = x + l \sin(\theta), \) and \(l\) is the length of pendulum. Assuming a small deviation of the torso, the pendulum equation can be linearized around the vertical axis. Therefore, we get \(z_2 = l\) and \(x_2 = x + l\theta\), and substituting this into Equation (6.7) results in

\[
p_x = x + \frac{ml}{(M + m)} \dot{\theta} - \frac{M z_c + m(z_c + l)}{(M + m)g} \ddot{x} - \frac{ml(z_c + l)}{(M + m)g} \ddot{\theta} = (1, c_1, 0, 0, c_2, c_3)x \quad (6.8)
\]

where \(x = (x, \theta, \dot{x}, \dot{\theta}, \ddot{x}, \ddot{\theta})^T\) is defined as the state vector, and consequently the control input vector is defined as \(u = (u_1, u_2)^T\), \(u_1 = \frac{d}{dt} \ddot{y}\) and \(u_2 = \frac{d}{dt} \dot{\theta}\). Therefore, the
system for control of ZMP can be defined as follows, where the output is derived from Equation (6.8).

\[
\dot{x} = \begin{bmatrix} 0 & I_{4 \times 4} \\ 0 & 0 \end{bmatrix} x + \begin{bmatrix} 0 \\ I_{2 \times 2} \end{bmatrix} u,
\]
\[y = p_x = \begin{bmatrix} 1 & c_1 & 0 & 0 & c_2 & c_3 \end{bmatrix} x. \quad (6.9)
\]

The transfer function matrix for this system is

\[G(s) = C(sI - A)^{-1}B = \frac{1}{s^3} \begin{bmatrix} 1 + c_1 s^2 & c_2 + c_3 s^2 \end{bmatrix}. \quad (6.10)
\]

Similar to section 6.1.2 the system does not have any transmission zeros. Therefore, one can design a multivariable controller to achieve the desired bandwidth required for the purpose of feedback stabilization. The next section, briefly discusses the idea of using finite horizon LQR feedback control (preview control) for ZMP trajectory generation.

### 6.1.4 Pattern generation by ZMP preview control

Consider the dynamics of pendulum cart table model given by Equation (6.9). The discrete time dynamics of this system can be expressed by:

\[
x(k+1) = Ax(k) + Bu(k), \quad y(k) = Cx(k) \quad (6.11)
\]

where the pair \((A, B)\) corresponds to the discrete time versions of Equation (6.9) with sampling time \(T\) and \(u(k) = (u_1(k) \ u_2(k))^T\). Integral action is introduced to the system by using incremental control and state \(\Delta u(k) = u(k) - u(k - 1)\), \(\Delta x(k) = x(k) - x(k - 1)\), respectively. The state vector is augmented as \(\tilde{x}(k) = \)
\( \begin{bmatrix} I & CA \\ 0 & A \end{bmatrix} \begin{bmatrix} \tilde{A} \\ \tilde{B} \end{bmatrix} \begin{bmatrix} \tilde{C} \end{bmatrix} \begin{bmatrix} 1 & 0_{1 \times 6} \end{bmatrix} \). (6.12)

An optimal control problem is formulated by minimizing

\[
J = \sum_{i=0}^{\infty} \left[ e^T(i)Q_e(i) + \Delta x^T(i)Q_x\Delta x(i) + \Delta u^T(i)R\Delta u(i) \right]
\] (6.13)

where \( e(i) = p_x(i) - p_{\text{ref}}(i) \), and the optimal control is given by

\[
u_o(k) = -G_I \sum_{i=0}^{k} e(i) - G_x x(k) - \sum_{i=1}^{N_L} G_d(i)p_{\text{ref}}(k+i)
\] (6.14)

where \( p_{\text{ref}} \) is the reference ZMP trajectory in \( x \) direction. The parameter \( N_L \) determines the horizon of the future desired ZMP. The optimal gain is determined by solving the discrete time algebraic Riccati equation,

\[
\tilde{P} = \tilde{A}^T \tilde{P} \tilde{A} - \tilde{A}^T \tilde{P} \tilde{B}(R + \tilde{B}^T \tilde{P} \tilde{B})^{-1} \tilde{B}^T \tilde{P} \tilde{A} + \tilde{Q}
\] (6.15)

where \( \tilde{Q} = \text{diag}\{Q_e, Q_x\} \). Hence, the optimal gain is defined by

\[
\tilde{K} = [G_I \ G_x] = (R + \tilde{B}^T \tilde{P} \tilde{B})^{-1} \tilde{B}^T \tilde{P} \tilde{A}
\] (6.16)

and the optimal preview gain is calculated by the following recursive formula:

\[
\begin{align*}
G_d(l) &= (R + \tilde{B}^T \tilde{P} \tilde{B})^{-1} \tilde{B}^T \tilde{X}(l-1), \\
\tilde{X}(l) &= \tilde{A}_{c}^T \tilde{X}(l-1), \quad l = 2, ..., N_L,
\end{align*}
\] (6.17)

where \( \tilde{A}_{c} = \tilde{A} - \tilde{B} \tilde{K} \), \( G_d(1) = G_I \) and \( \tilde{X}(1) = -\tilde{A}_{c}^T \tilde{P}[1 \ 0_{1 \times 6}]^T \). Further details about this method is given in [207]. A numerical example is given in the section 6.1.6 where the generator is producing the desired CoM motion. In the next section, the online stabilization problem of this method is studied.

### 6.1.5 ZMP Stabilization by Using the Torso

In section 6.1.4, a walking pattern generator based on the new pendulum cart table model is introduced. However, due to imperfect ground conditions, modelling errors...
and unknown disturbances a real time feedback stabilizer must be used to adapt the generated trajectories based on the sensor information. This section proposes a ZMP feedback stabilization scheme to achieve robustness in bipedal walking.

Consider Equation (6.9) with modified outputs as

$$\dot{x} = \begin{bmatrix} 0 & I_{4 \times 4} \\ 0 & 0 \end{bmatrix} x + \begin{bmatrix} 0 \\ I_{2 \times 2} \end{bmatrix} u \quad y = \begin{bmatrix} p_1 \\ p_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & c_2 & 0 \\ 0 & c_1 & 0 & 0 & c_3 \end{bmatrix} x \tag{6.18}$$

where $p_1$ and $p_2$ are the ZMP outputs corresponding to hip and torso, respectively. Reference tracking using feedback can be implemented as follows. Consider the system defined in Equation (6.18) in compact format as

$$\dot{x} = Ax + Bu \quad y = Cx. \tag{6.19}$$

In order to track the ZMP reference trajectory, an integral term is introduced between reference vector $r$ and the system’s output vector $y$ as a new state variable $z$, ie. $\dot{z} = r - y$. Therefore, the new state space system is given by

$$\begin{bmatrix} \dot{x} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} A & 0 \\ -C & 0 \end{bmatrix} \begin{bmatrix} x \\ z \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} u + \begin{bmatrix} 0 \\ I \end{bmatrix} r \quad y = \begin{bmatrix} C & 0 \end{bmatrix} \begin{bmatrix} x \\ z \end{bmatrix}. \tag{6.20}$$

Since the system in Equation (6.20) is controllable we can use feedback $u = -[K_1 \ K_2]x$ to stabilize the system and track the reference vector $r$. As a result of feedback the closed loop system is obtained as:

$$\begin{bmatrix} \dot{x} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} A - BK_1 & -BK_2 \\ -C & 0 \end{bmatrix} \begin{bmatrix} x \\ z \end{bmatrix} + \begin{bmatrix} 0 \\ I \end{bmatrix} r \quad y = \begin{bmatrix} C & 0 \end{bmatrix} \begin{bmatrix} x \\ z \end{bmatrix}. \tag{6.21}$$

A numerical example for the ZMP stabilization is given in section 6.1.6 and the results are compared with the old cart table model.

6.1.6 Comparison and Simulation Results

Two simulations are presented. Firstly, the extended cart table model which was proposed in section 6.1.3 is simulated using the preview control idea which was introduced in section 6.1.4. The desired ZMP trajectory is a series of steps with 0.2 m
length which corresponds to the step size and the tracked ZMP of the extended cart table model and the trajectory CoM (robot’s hip) are plotted over the desired ZMP in Fig. 6.6. The torso is kept close to the vertical, however, it can be moved depending on the ground slope. The controller and model parameters used are \( g = 9.81 \, \text{m/sec}^2 \), \( z_c = 0.6 \, \text{m} \), \( m = 8 \, \text{kg} \), \( M = 12.5 \, \text{kg} \), \( l = 0.3 \, \text{m} \), sampling time \( T = 0.005 \, \text{sec} \), preview time \( T_{\text{prev}} = 2 \, \text{sec} \) \( (NL = \frac{T_{\text{prev}}}{T}) \), \( Q_c = I_{2\times2} \), \( Q_x = I_{6\times6} \), \( R = 10^{-3} \, I_{2\times2} \) and simulation time is \( 8 \, \text{sec} \). The control inputs that corresponds to the jerk of the CoM are also shown in Fig. 6.6. The smoothness of the designed trajectories is an important characteristic that is achieved in this simulation. The smooth hip trajectory can be processed by an inverse kinematics algorithm to derive the corresponding joint angles.

Secondly, a numerical example of the real time feedback scheme that is introduced in section 6.1.5 is illustrated in Fig. 6.7. This simulation quantifies the improvements in robust ZMP tracking when the torso is used in the trajectory generation. Note that the step height is \( 0.1 \, \text{m} \) and the poles of system Equation (6.21) are placed at \([-6 - 6 - 8 - 8 - 10 - 10 - 12 - 12]\). The hip’s undershoot is reduced from \( 5.6 \, \text{cm} \) to \( 2.4 \, \text{cm} \) by using the upper body movement, which is \( 57\% \) reduction in the total ZMP of the new pendulum cart table model. At the same time the settling time in cart ZMP step response is \( 0.53 \, \text{sec} \) and in the total ZMP response the settling time is \( 0.74 \, \text{sec} \). That is the new model can achieve a reasonable tracking speed, and avoid the undesired undershoot. In addition, due to practical limitations the range of upper body movement must be as small as possible. This is illustrated in Fig. 6.8. Note that the upper body is moving by \( 0.24 \, \text{rad} \) which is \( 13.7 \) degrees.

### 6.2 Selection of Walking Parameters

Walking has been long studied in order to answer various questions regarding the speed, energy efficiency and choice of walking parameters such as step size by humans. In an early study [208], simple inverted pendulum model of walking was used to derive a theoretical upper bound on the human’s walking speed. The formula is \( v \leq \sqrt{gl} \)
Figure 6.6: The pattern generation using the extended pendulum cart table model (top) and the corresponding control inputs (bottom).

Figure 6.7: The role of torso in minimizing the amount of undershoot, hence improving the ZMP stabilization.
where $g$ is gravity and $l$ is the pendulum’s leg length. A simple calculation was shown that for an average human the maximum achievable speed is about $\sqrt{10 \times 0.9} = 3 \, \text{m/sec}$. Although this is an interesting initial result, there are other limiting factors in walking that can provide a tighter upper bound on the maximum feasible bipedal walking speed. Hence, it is of interest to study various limiting factors for humanoid robots. The energy analysis aspect of walking has also been studied in the literature [209, 210].

In this section, new results are presented which can serve as a guidance on the choice of walking parameters, such as the average walking speed, step length, tracking bandwidth and joint torque. The results are organized in terms of kinematic (bandwidth, mechanical joint limits, bent knee, use of toes and maximum angular velocity) and dynamic (ZMP criterion, capture point and ankle torque) relations. The bandwidth requirements are meant to provide a rule of thumb for specifying the required performance on the tracking control system in order to achieve a desired walking
speed. The key walking requirements for a bipedal robot’s tracking bandwidth, ankle joint torque for a given walking speed, step length and toe push-off configuration are presented.

### 6.2.1 Assumptions and Set-up

The main focus is on the maximum feasible speed and its relation to the maximum step size and the required bandwidth. The reason for choosing the compass gait model to study a high DoF pair of legs is due to the fact that a complex configuration can be represented by a simpler compass gait model where the legs of the compass gait are the effective virtual length of the original model as shown in Fig. 6.9.

![Figure 6.9: Representation of a more complex pair of legs with the compass gait model, during toe push-off.](image)

A planar model in sagittal plane is considered. Fig. 6.10 shows the relations between the inter-leg angle $\theta$, step length $y$, hip height $z$ and leg length $l$. According to trigonometric relations, the step size is $y = 2l \sin(\theta/2)$ and the hip height is $z = l \cos(\theta/2)$.

In the case where the angle $\theta$ is small ($|\theta| \leq \frac{\pi}{6}$), $y \approx l\theta$ and $z \approx l$. The compass gait model is used to derive an analytic solution for the maximum bandwidth required for a desired reference trajectory.

The instantaneous walking speed, $v$ (speed of CoM) over a single step as shown
Figure 6.10: Compass gait model of walking.

in Fig. 6.10 is

$$v(t) \approx \lim_{t \to t_0} \frac{l(\theta(t) - \theta(t_0))}{t - t_0} = \dot{l}(t)$$  \hspace{1cm} (6.22)

and the average walking speed is

$$\bar{v} = \frac{y}{T_s} = \frac{\int_{t_0}^{t} v(t) \, dt}{\Delta t} \approx \frac{l(\theta(t) - \theta(t_0))}{\Delta t} = \frac{l \Delta \theta}{\Delta t}$$  \hspace{1cm} (6.23)

where $T_s = \Delta t = t - t_0$ is the time required for taking one step. Equation (6.23) relates the forward average walking speed $\bar{v}$ to the average angular velocity at the ankle and hip joints $\frac{\Delta \theta}{\Delta t}$. It should be noted that by assuming that $\bar{v}$ does not change in the subsequent steps, $\bar{v}$ is the average walking speed over the entire walking gait.

6.2.2 Bandwidth Requirement

Bandwidth is the maximum frequency at which the output of a system will track an input sinusoid reference signal and it is used as a measure for the speed of response. The definition of bandwidth is based on the magnitude of the frequency response and phase information is not considered. In fact, a system with bandwidth $\omega_B$, tracks a reference signal often with a phase lag (delay in time domain). Therefore, for practical purposes the bandwidth is required to be ten times (a decade in bode plot) higher than the minimum requirements.

Suppose the desired reference trajectory is a sinusoidal signal, $r(t) = \frac{\theta}{2} \sin(\omega t - \frac{\pi}{2})$, where $\omega$ is the frequency of the sine wave. The use of sine wave for deriving the above
relations can be justified by considering the fact that in steady state bipedal walking, the joints’ trajectories closely resemble a periodic sine wave signal.

The bandwidth of the control system must be simply higher than \( \omega \) to be able to track the signal \( r(t) \). Moreover, the bandwidth \( \omega_B \) can be related to the average walking speed by utilizing Equation (6.23) and observing that \( \omega = \frac{\pi}{T_s} \). Hence,

\[
\omega_B > \frac{\pi \bar{v}}{y} \left( \frac{rad}{sec} \right) \rightarrow f_B > \frac{\bar{v}}{2y} \, (Hz). \tag{6.24}
\]

As mentioned earlier, in practice the bandwidth of the system must be considered to be at least ten times more than the minimum values given in Equation (6.24). The required bandwidth versus walking speed is shown in the section 6.2.11, for ASIMO and CoMan using their physical parameters, respectively.

On the other hand, considering the stiffness in the ankle \( k_a = 185 \, Nm/rad \) of CoMan a maximum feasible bandwidth \( \omega_m \) (on the link side) can be computed which is the resonant frequency of the compliant joint. This frequency is

\[
\omega_m = \sqrt{\frac{mgl + k_a}{ml^2}}. \tag{6.25}
\]

Equation (6.25) is derived from characteristic equation of a pendulum with a torsional spring and no friction that is

\[
ml^2 \ddot{\theta} + mgl\dot{\theta} = -k_a\theta. \tag{6.26}
\]

It can be seen that if \( k_a = 0 \) the pendulum will swing with the natural frequency of \( \frac{1}{2\pi} \sqrt{\frac{g}{l}} \) (Hz), while if \( k_a > 0 \) the pendulum can swing faster than the pendulum which is only under the gravity effect.

### 6.2.3 Mechanical Joint Limits

Mechanical joint limits are designed for CoMan that limit the range of motion, in particular on the ankles. The reason for adding these hard limits is to protect the robot from any damages in case of instability. Sagittal ankle joint limits on the current version of CoMan are \( +26^\circ \) and \( -13^\circ \). In order to have symmetry, suppose \( |\theta| \leq 13^\circ \), hence the maximum step size is \( y = 2 \times 0.49 \times \sin(13^\circ) = 0.22 \, m \).
A limitation on the step size would limit the average feasible walking speed. Hence, substituting $g = 9.81 \text{ m/sec}^2$, $l = 0.49 \text{ m}$, $m = 9.84 \text{ kg}$ and $k_a = 185 \text{ Nm/rad}$ in Equation (6.25), gives $\omega_m = 9.74 \text{ rad/s}$. Based on Equation (6.24), maximum step size of $y = 0.22 \text{ m}$ and maximum bandwidth of $\omega_m = 9.74 \text{ rad/s}$, the maximum walking speed is 0.68 m/sec.

### 6.2.4 Bent Knees

A common problem in humanoid robots is use of bent knees while walking was discussed in section 2.4. A drawback of this is reducing the walking speed by decreasing the effective leg length. Assuming the knee joint to be at the centre of the leg in Fig. 6.11, bending the knee by the angle $\alpha$ gives a total leg length of

$$\bar{l} = l \cos \left( \frac{\alpha}{2} \right).$$

(6.27)

and the corresponding step size of

$$y = 2l \cos \left( \frac{\alpha}{2} \right) \sin \left( \frac{\theta}{2} \right).$$

(6.28)

Hence, for instance bending the knees by $\alpha = 0.3 \text{ rad}$ and considering the mechanical joint limit on $\theta$ gives an step size of $y = 0.217 \text{ m}$. Based on Equation (6.25), the maximum walking speed is 0.67 m/sec.

Figure 6.11: The trigonometric relations of a bent knee can be used to calculate the virtual leg length.
6.2.5 Use of Toes

CoMan has passive toe joints which can improve the kinematics of walking in terms of taking larger step lengths. The triangle formed by using the toe is shown in Fig. 6.12. Since the leg length \( l \) and the foot length \( l_1 \) and the angle between them \( \beta \) are known, the virtual leg length \( l_x \) is

\[
l_x = \sqrt{l^2 + l_1^2 - 2ll_1 \cos(\beta)}
\]

where the relative angle between the toe and the foot with size \( l_1 \) does not enter at this stage. In case of CoMan where the maximum value of \( \beta = 103^\circ = 1.8 \text{ rad} \), \( l = 0.49 \text{ m} \) and \( l_1 = 0.1 \text{ m} \) the effective leg length is \( l_x = 0.52 \). Moreover, the values under the square root in Equation (6.29) are always positive.

![Figure 6.12: The trigonometric relations when using the toe joint can be used to calculate the virtual leg length.](image)

Consider Fig. 6.13 to compute the step length. In order to derive the step size when the toe joint is used, the angle \( \theta_x \) must be derived. It is assumed the the virtual leg length in both legs are equal.
The step size is
\[ y = 2l_x \sin \left( \theta_1 + \theta_2 - \frac{\pi}{2} \right) \]  
(6.30)
where \( l_x \) is derived in Equation (6.29) and \( \theta_x = 2(\theta_1 + \theta_2) - \pi \). In practice, \( \theta_1 \) is measured with a sensor and \( \theta_2 = \arcsin \left( \frac{l_x \sin(\beta)}{l_x} \right) \). The limiting factor in Equation (6.30) is the joint limit on the passive toes that varies depending on the spring. However, following the numerical example and assuming the maximum \( \theta_1 = 0.7 \text{ rad} \), the maximum \( \beta = 1.8 \text{ rad} \), the maximum \( \theta_2 = \arcsin(0.5 \sin(\beta)) = 1.2 \text{ rad} \) (due to ankle’s mechanical joint limit) by using the toes the step size would be \( y = 0.33 \text{ m} \) and based on Equation (6.25) the maximum walking speed is 1 \( \text{m/sec} \). The configuration described in this section is implemented by the Toyota’s running robot, where the robot runs on its toes, as discussed in chapter 2.

6.2.6 Use of Toes and Knees

Combined use of toes and knees cancels out the benefit of having a longer virtual leg, but as seen in bipedal walking, it can reduce the torque requirement on the leading ankle joint. This scenario is illustrated in Fig. 6.14, where the robot on the right relies on flat foot walking and hence due to kinematic constraints shown in the figure, can not move its CoM over the leading support foot. In other words, the robot is in a locked position, which requires an excessive torque on the leading’s foot ankle to take the step. However, the robot on the left benefits by using the toe-off and to move its
CoM over the leading foot area. The force $f$ shown in the figure is the toe-off force which greatly reduces the ankle torque.

![Figure 6.14: Toe-off during walking (left), and flat foot walking (right).](image)

Moreover, flat foot walking limits the step size (and hence the walking speed) due to limited torque available at the leading ankle joint. This issue is considered next.

### 6.2.7 Torque Requirement

Taking different step lengths has some implications in terms of joint torques. The ankle torque in Fig. 6.10, during single support, assuming zero initial velocity, is

$$\tau = (m_H l + m_H \frac{l}{2}) g \sin \left(\frac{\theta}{2}\right).$$

Considering small angles $\tau \approx (m_H + \frac{m_2}{2}) g (\frac{l \theta}{2})$, where $l \theta$ is the step size. Hence, the maximum torque available at the ankle is one of the limiting factors in taking larger steps. However, if toe-off is used as explained in the previous section, this limiting factor can be addressed.

### 6.2.8 Maximum Angular Velocity

Harmonic drives that are used in CoMan can tolerate a maximum angular velocity that puts a limitation on the feasible reference signal.

The rate of the reference signal is $\dot{\theta}(t) = \frac{6}{7} \omega \cos(\omega t - \frac{\pi}{2})$. Since $T_s$ (the step time) is known in terms of the given step size $y$ and average walking speed $\bar{v}$, the maximum
value of the reference signal is 

\[ \dot{r}_{\text{max}} = \frac{\theta}{2} \omega = \frac{\pi \bar{v}\theta}{2y}, \] (6.31)

The maximum allowable angular velocity at the input of the harmonic drive (available from the manufacturer catalogue) must be in agreement with the results of Equation (6.31). The calculations mentioned above are considered for the ankle joint, while the hip joint satisfies the same relations with the only difference that the amplitude of the sine wave is twice the value of the ankle joint. The formula for the hip joint can be derived by simply replacing \( \frac{\theta}{2} \) by \( \theta \) in the equations above. This point is illustrated in Fig. 6.15 which shows that the angular distance travelled by the hip joint in a single step is twice the value of the ankle joint’s angle, since the swing foot has to move twice the step size in a single step while the hip mass moves by the step size \( y \).

Figure 6.15: The linear and angular distances travelled by the ankle and the hip during a single step. (The stance leg is denoted by a solid line and the swing leg is distinguished by a dashed line).

### 6.2.9 ZMP Dynamic Limitation

This section shows a limitation on the maximum feasible bandwidth and accordingly the maximum feasible walking speed when the conservative ZMP criteria is used in trajectory design. It is assumed that the mass of each leg is negligible compared the
the mass of the hip and the foot is symmetric with respect to the ankle. Fig. 6.16 shows the LIP model of robot, in single support that is driven by the sine wave \( r(t) = A \sin(\omega t - \frac{\pi}{2}) \). Assuming that the frequency of the sine wave is within the bandwidth, \( \theta(t) = r(t) \) and for small angles \( x(t) = Al \sin(\omega t - \frac{\pi}{2}) \), and \( A > 0 \). Hence, the location of the ZMP can be derived from \( P_x = x - \frac{zc}{g} \ddot{x} \), by substituting the value of \( x(t) \) and its acceleration, as

\[-D < Al(1 + \frac{zc}{g} \omega^2) \cos(\omega t) < D \]  

(6.32)

where \( D > 0 \) is half of the foot size. Equation (6.32) is equivalent to

\[ |Al(1 + \frac{zc}{g} \omega^2) \cos(\omega t)| < D, \]  

(6.33)

where \( Al(1 + \frac{zc}{g} \omega^2) > 0 \). Setting Max \( |\cos(\omega t)| = 1 \), and rearranging Equation (6.33) in terms of \( \omega \), we get

\[ \omega < \sqrt{\frac{(D-lA)g}{zcA}} \]  

(6.34)

and

\[ \bar{v} < \frac{y}{\pi} \sqrt{\frac{(D-lA)g}{zcA}} \]  

(6.35)

where according to the CoM stability \( (D-lA) > 0 \). Equation (6.35) has a maximum point if \( A \) is chosen as the only variable \( (y \approx 2lA) \), which is \( A = D \). Therefore, in the case of CoMan with parameters \( D = 0.08 \) m, \( zc = l = 0.49 \) m, \( g = 9.81 \) m/sec\(^2\), \( A = 0.08 \) rad and \( y = 0.0784 \) m, Equation (6.34) predicts \( \omega < 4.5648 \) rad/sec and Equation (6.35) predicts \( \bar{v} < 0.114 \) m/sec to be the limit on the feasible average walking speed (which is about quarter of a leg length per second).
Figure 6.16: The ZMP of the LIP model driven by a sine wave in single support phase.

Similar analysis can be done for double support phase where the support area is larger and the speed can be increased. Assuming the maximum step size of \( y = 2 \ l \ \sin(A) \), \( A < 0.22 \ \text{rad} \), and the size of each foot to be \( F_S = 0.16 \ \text{m} \), the total support area in Fig. 6.17 will be \( F_S + y \). Therefore, \( D = \frac{F_S + y}{2} = l \ A + \frac{F_S}{2} \) and substituting the values in Equation (6.35), gives \( \bar{v} < 0.261 \ \text{m/sec} \) in the double support phase. The maximum average speed between the single support and double support is \( \bar{v} < 0.1875 \) which is approximately half leg length per second.

Figure 6.17: The ZMP of the LIP driven by a sine wave in double support phase.
6.2.10 Capture Point

Recent results that were shown in [112], can be utilized in simple models such as the model of Fig. 6.16 to estimate the limitations on balance recovery. It is assumed that the legs are massless and the hip mass is a point mass. In summary, if an external force is applied to the CoM that causes it to have a velocity $v$, the required step size or the capture point $x_{cap}$, in order to come to a complete stop is

$$x_{cap} = v \sqrt{\frac{zc}{g}}$$  \hspace{1cm} (6.36)$$

where in case of CoMan, knowing the maximum step size $x_{cap} = y = 0.22 \, m$, the maximum velocity due to a push that can be controlled by taking a single step is $v < 1 \, m/sec$. However, higher pushes can possibly be controlled by taking more than one step, or angular momentum (gyroscopic effect) of upper-body can be used to obtain a capture region (an area instead of a point).

6.2.11 Numerical Examples

In this section, Equations (6.24) is evaluated numerically for ASIMO. The step size is considered to be $y = 0.5 \, m$ (blue curve with slope of one) and $y = 0.25 \, m$ (red curve with the slope of two). The results are shown in Fig. 6.18 for the ankle joint.

![Figure 6.18: Estimated minimum required bandwidth for ASIMO as a function of the average walking speed.](image)
Table 6.1: Maximum feasible walking speed for CoMan (with flat feet).

<table>
<thead>
<tr>
<th>Speed $v$ (m/sec)</th>
<th>$\sqrt{gl}$</th>
<th>Joint Limits</th>
<th>Bent Knees</th>
<th>Use of Toes</th>
<th>ZMP</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2</td>
<td>0.68</td>
<td>0.67</td>
<td>1</td>
<td>0.19</td>
<td></td>
</tr>
</tbody>
</table>

Equation (6.24) is evaluated numerically for CoMan’s ankle joint. The step size is chosen as $y = 0.2 \, m$. The result is shown in Fig. 6.19 which is a minimum requirement for the bandwidth.

Figure 6.19: Minimum required bandwidth for CoMan as a function of the average walking speed.

A comparison between Figs. 6.18 and 6.19, shows that ASIMO has the benefit of increased leg length, which in terms means larger step size, in comparison to CoMan and hence for the same walking speed it requires less bandwidth. However, a fair comparison can be made if the walking speed is normalized in terms of leg length and froude number can be considered for comparing walking speed among different bipedal robots. Moreover, the maximum angular joint velocity is of interest since there is a limit to the maximum angular velocity at the input of the Harmonic drive.

Hence, the upper bound $\omega_m$ in Equation (6.25) together with Equation (6.24) can be used to quantify the main results on the maximum feasible walking speed for CoMan which are summarized in Table 6.1.
6.3 Conclusions and Open Problems

This chapter discussed a gap in ZMP robust bipedal walking which was use of torso in the preview control framework. An online trajectory generation method using a modified version of the cart-table model was presented that uses the upper body to improve the stability of walking. Numerical examples were used to illustrate the improvement in terms of ZMP tracking for simple models. In the section 6.1, a solution for the stabilization problem of old models due to their non-minimum phase characteristic was proposed. The new pendulum cart table model combines the advantages of existing models and makes use of the upper body to compensate the unknown ZMP deviations. The effectiveness of this model is shown by numerical examples both in the case of pattern generation and in the case of ZMP feedback stabilization. In ZMP stabilization the torso is used for reducing or ideally cancelling the undesired effects of undershoots and bandwidth limitations.

Moreover, a theoretical study was conducted on the relationship between the average walking speed, step length and the required tracking bandwidth. It was shown that how different kinematic characteristic of CoMan can limit its maximum feasible walking speed. The limitation of using ZMP for flat foot walking was also quantified using simple models. Furthermore, it was pointed out that if walking trajectories are developed with toe-off phase as opposing to flat foot walking, the robot can achieve higher walking speeds (due to larger steps) while requiring less torque on the ankle joints. CoMan has passive toe joints that can be used for this purpose.
Chapter 7

Development of a Dynamic Walking Simulator

In this chapter, a dynamic simulator is developed for CoMan to simulate the generated walking trajectories in order to evaluate different aspects of the walking trajectories such as step size, centre of gravity, joint torques, tracking performance, etc. A dynamic simulator must include the multibody model of the bipedal robot, actuator dynamics, compliance, sensor noise and the ground contact model. Numerous commercial and academic walking simulators exist in the robotics and biomechanics’ communities and they were reviewed in Chapter 2. A suitable simulator must have an easy to use GUI to update the model as the robot changes, as well as the tools to include the precise multibody and actuator dynamics together with powerful solvers and control system tools. Hence, the dynamic walking simulator is developed based on Robotran and Matlab which provide the required tools and options.

The main contribution of this chapter is to develop a dynamic walking simulator for CoMan which includes the dynamical parameters (body length, mass, CoM, inertia), actuator and compliance dynamics, ground contact model and sensor noise. In section 7.1, a brief summary of how Robotran works is provided. In the subsequent sections development of the multibody model of CoMan, actuator and compliance dynamics, ground contact model and sensor noise in Robotran and Matlab are presented and a walking trajectory is designed and simulated on the dynamic model.
Finally, conclusions and future work are discussed.

## 7.1 Development of CoMan’s Model in Robotran

Robotran [144] is a symbolic model generator tool for multi-body systems that is written in Matlab. Therefore, additional code can be written in Matlab language for further analysis. The essential building blocks of this toolbox which are used in CoMan simulator are introduced in this section.

After derivation of the dynamic parameters of CoMan from the CAD drawings and translating the parameters from the CAD conventions to the Robotran conventions, a dynamic simulation is developed as shown in Fig. 7.1. The tools available from Robotran toolbox such as linearization and computation of the overall dynamics must be combined with further programming to add the actuator and compliance dynamics, the ground reaction force, the inverse kinematics computations, main walking simulation m-file and finally the 3D visualization of the simulated walking gait. The contribution of this section is the addition of the mentioned modules to Robotran.

Next, the development of a dynamic simulation for CoMan with 10 DoF is presented. This model which is built in single support phase, includes the actuator, compliance and integrator dynamics as well as the ground contact model under the swing leg. The dynamic model is then used to simulate a walking gait in Robotran and Matlab.

### 7.1.1 The Single Support Model in Robotran’s GUI

In modelling humanoid robots, it is often required to model a robot with different degrees of freedom in an easy to use environment where it is possible to quickly edit the models to keep them up to date with the hardware changes. Writing the models by hand is a laborious and tedious task. In the Robotran GUI (MBSysPad), open and closed articulated chains of bodies can be drawn graphically. Then the software generates the multibody dynamics, given in Equation (7.1), in symbolic form as Matlab m-files.
In this section, CoMan is modelled in single support phase with the right leg as the support leg and 5 actuated DoF in each leg, as shown in Fig. 7.2. The joints ordering follows the kinematics of the hardware, that is, starting from the base the joints are ordered as ankle roll, ankle pitch, knee, hip roll and hip pitch for the right leg and the mirror of this order is applied to the left leg from the hip to the foot. The dynamic parameters of CoMan which are extracted from the CAD drawings are provided in Appendix C. The modelling convention in Robotran is based on the right handed coordinate system with relative angles considered for the joints. The generated symbolic dynamics after simplification are given in

$$M(q)\ddot{q} + c(q, \dot{q}) = \tau.$$  \hspace{1cm} (7.1)

The numerical values for the parameters are stored in the Robotran’s data structure called “mbs_data” that can be modified from the GUI or within Matlab. This model has sensors attached to the CoM of each body and also to the beginning and end of each body to compute the forward and inverse kinematics as marked with circles in Fig. 7.2 and discussed in section 7.1.2. The external force feature of Robotran
is used to simulate the ground contact model under the swing foot (double support phase) which is discussed in section 7.1.5. In addition, the simulation pseudo code that is used for simulating the nonlinear model of CoMan is provided in the pseudo code 7.1.

Pseudo code 7.1: Matlab code for simulation of CoMan.

```matlab
global MBS_user

% Project loading and initialization
[mbs_data, mbs_info] = mbs_load(Coman’s Robotran Model);
% Load the designed feedback gains and Trajectories
% Set the initial conditions
% Set the number of newly defined variables (motors, compliance, integrators)
mbs_data.Nux = n;
% Set the ode45 options
% Create a log for the states
loco.y = [];
loco.t = [];
...
for s=1:n_steps
    tsuan = tspan0:Tspan:n_steps+max_time;
    [MBS_dirdyn] = mbs_new_dirdyn(mbs_data);
yprime = str2func(MBS_dirdyn.fctDerivatives);
callfct = mbs_get_fct_handle(MBS_dirdyn.fctname);
[tspan, y] = ode45(yprime, tspan, y0, options, callfct, 0);
% Update the logs
loco.y = [loco.y; y];
loco.t = [loco.t; tspan];
...
% Clear the Robotran’s internal variables for the next step
mbs_data.q = 0;
mbs_data.qd = 0;
mbs_data.qdd = 0;
...
% Reset the initial time to the end of the step:
t0 = tspan(end);
% At the end of each step switch the legs:
y0(1:n) = -data.y(end, 1:n);
end
```
7.1.2 CoMan’s Kinematics and Sensors

The Jacobian, cartesian position, velocity and acceleration of any point on the multibody system are computed in Robotran symbolically, which proves useful for analysing the forward and inverse kinematics. This is done via a tool called a “Sensor” that can be placed at the end of an anchor point. An anchor point itself is a three dimensional vector that is used to locate various points on the model, as shown in Fig. 7.2.

The sensors are placed at the CoM and beginning and end of each body as shown in Fig. 7.2 (only the sensors for the support leg are shown for simplicity, while the other leg has a similar arrangement). The total CoM is

\[
\mathbf{p}_{\text{CoM}} = \frac{\sum_{i=1}^{n} \mathbf{p}_i m_i}{\sum_{i=1}^{n} m_i}
\]

(7.2)

where \(\mathbf{p}_i\) is the 3D cartesian position (in \(XYZ\) directions) of the CoM of \(i^{th}\) body with mass \(m_i\). \(\mathbf{p}_i\) is available as a symbolic m-file generated by the sensor, while additional programming should be done to compute the CoM or the ZMP of the total multibody system. The cartesian position of the beginning and end of each body is used to visualize the 3D model of the robot as shown in Fig. 7.3. The centre of gravity in 3D (the diamond shape) and its projection on the \(XY\) plane (the star

Figure 7.2: A model of CoMan in Robotran with 10 DoF.
shape) are also illustrated.

In addition to forward kinematics, the Jacobian \( \frac{dp}{dq} \) at different points on each body is available in a symbolic m-file that is used to compute the inverse kinematics.

### 7.1.3 Joint Torques

The joint torques \( \tau \) that are given on the right hand side of Equations (3.1) and (7.1) are accessible in Robotran using the template m-file called “userJointForces.m”. In CoMan actuators have significant dynamics and the supplied joints’ torques must be provided via a coupling with the actuators and the passive compliance. Therefore, the “userJointForces.m” m-file is used to couple the actuator and compliance dynamics to the multibody dynamics of CoMan as given in Equation (3.9). In fact, the torque accessible in this m-file is the coupling torque given in Equation (3.10).

The torque in each joint of CoMan is controlled indirectly via the motor position that deflects the passive spring and subsequently moves the joint. As mentioned in
section 3.5 the joints’ torque vector is

\[ \tau_L = B_s (\dot{q}_m - \dot{q}) + K_s (q_m - q). \] (7.3)

\( \tau_L \) is programmed in the “user_JointForces.m” m-file which couples the mechanical joint torques to the additional motor and compliance dynamics that are introduced in section 7.1.4. It can be seen that the motor position leads the joint position and the difference between the two produces the required joint torque.

In addition, \( \tau_0 \) which was derived from the inverse dynamics to keep the upright posture is added to \( \tau_L \) during the simulation. The dynamics associated with the motors and integrators are added to the simulation in the next section.

### 7.1.4 Compliance and Actuator Dynamics

In development of a simulation tool for a compliant humanoid robot it is necessary to introduce additional dynamics due to the actuators and compliance dynamics as well as the integrators for tracking. These additional dynamics can be introduced to the simulation via a built in m-file called “user_Derivatives.m”.

The LQR feedback is loaded to Matlab’s workspace to compute the control input as

\[ u(t) = -\begin{bmatrix} K_p, K_d, K_{mp}, K_{md}, K_i \end{bmatrix} \begin{bmatrix} q, \dot{q}, q_m, \dot{q}_m \end{bmatrix}^T \] (7.4)

where the feedback gain is derived in section 7.1.6. The control input is then applied to the DC motors’ dynamics as

\[ \ddot{q}_m = -J^{-1} (B_m + K_t R^{-1} K_b) \dot{q}_m - J^{-1} \tau_L + J^{-1} K_t R^{-1} u(t) \] (7.5)

and finally the integrator state are updated based on the position of the links and the value of the reference vector at the current time as

\[ \dot{z} = r - q. \] (7.6)
The code for this part of the simulator is provided in the pseudo code 7.2. This pseudo code illustrate how to implement the motors and integrators dynamics as explained in Equations (7.4), (7.5) and (7.6).

Pseudo code 7.2: Matlab code for new additional dynamics.

```matlab
function [uxd] = user_Derivatives(ux, mbs_data, tsim)
    % Unpack the states from mbs_data and MBS_user
    ...
    % Compute the control input (voltage)
    u = -K * [qd; qdm; qdmi; qi];
    % Compute the motors’ accelerations.
    qddm = (J^-1)(-(Bm + Kt*(R^-1)*Kb)*qdm - Tt + Kt*(R^-1)*q)*u;
    % Return the derivatives of the DC motor dynamics and integrators to ode45
    uxd = [qdm; qddm; r - q];
```

It should be noted that the desired output to be controlled is the link position in contrary to the current independent PID based control system which controls the motor position, and is currently implemented on CoMan as discussed in section 4.1. Further details regarding the additional dynamics are discussed in section 3.5.

### 7.1.5 External Forces and Ground Reaction Force

An important feature in Robotran is the external user defined forces that can be placed at any point on the multibody model in a similar way as placing “Sensors”, as shown in Fig. 7.2. A user defined external force can then be defined in an m-file called “user_ExtForces.m” that is integrated with the dynamics in the numerical simulation. This feature is particularly important for walking robots to introduce suitable ground contact models. The vertical ground reaction force is

\[
F_z = \begin{cases} 
0 & z \geq z_0 \\
-K_G \Delta z - D_G \Delta \dot{z} & z < z_0 
\end{cases}
\]

where \(\Delta z = z - z_0\) < 0, \(z_0\) is the ground level height which is zero on a flat ground and \(z\) is provided by the external force tool in Robotran, which measures the absolute
position of the contact point. The horizontal friction force is

\[ F_{x,\text{slide}} = -\text{sgn}(\Delta \dot{x}) \mu F_z \]  

(7.7)

\[ F_{x,\text{stick}} = -K_F \Delta x - D_F \Delta \dot{x} \]  

(7.8)

where \( \mu \) is the coefficient of friction, \( \Delta x = x - x_0 \), \( x_0 \) is the location of the foot’s contact point with the ground and \( x \) is the amount of horizontal displacement from the original contact point \( x_0 \). \( \Delta x \) indicates the relative motion with respect to the initial point of contact \( x_0 \). If \( |F_x| > \mu F_z \) the foot will slide on the ground and if \( |F_x| < \mu F_z \) the foot will stick to the ground. The vertical ground reaction force \( F_z \) is unilateral and can be only positive (the ground does not pull the robot) while the horizontal friction force can be positive or negative. Further details are provided in section 3.6. Further description of the compliant ground contact model is provided in section 3.6.

The external force is implemented at a single contact point under the swing foot and it consists of the vertical ground reaction force \( F_z \) and horizontal friction forces \( F_x \) and \( F_y \) that are identical in both directions. Moreover, \( F_z \) is a unilateral force and can only be positive. The floor has a friction coefficient that defines when the foot can slip on the floor. The spring damper and friction coefficients of the ground model are summarized in Table 7.1. The spring parameters are chosen in simulation to avoid the swing leg penetrating the ground more than 1 cm and the damper parameters are chosen to avoid oscillation once the foot hits the ground. The pseudo code 7.3 illustrate how to implement the contact model in the Robotran’s built in “user_ExtForces.m” m-file.

Table 7.1: Parameters of the ground contact model.

<table>
<thead>
<tr>
<th>Force</th>
<th>Spring Coef.</th>
<th>Damping Coef.</th>
<th>Friction Coef.</th>
<th>( \mu )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( F_x ) (N)</td>
<td>2000 (N/m)</td>
<td>40 (N.s/m)</td>
<td>0.8</td>
<td></td>
</tr>
<tr>
<td>( F_y ) (N)</td>
<td>2000 (N/m)</td>
<td>40 (N.s/m)</td>
<td>0.8</td>
<td></td>
</tr>
<tr>
<td>( F_z ) (N)</td>
<td>1000 (N/m)</td>
<td>10 (N.s/m)</td>
<td>-</td>
<td></td>
</tr>
</tbody>
</table>
CHAPTER 7. DEVELOPMENT OF A DYNAMIC WALKING SIMULATOR 195

Pseudo code 7.3: Matlab code for simulating the ground contact model.

```matlab
function [Fx,Fy,Fz] = user_ExtForces(mbs_data,varargin)

if PxF(3)<=-1e-6
    Fz=-1e3*PxF(3) - 10*VxF(3)
    % Horizontal friction force in Y direction:
    if myflagy==0
        MBS_user.temp_grf y=PxF(2);
        tempy=MBS_user.temp_grf y;
        MBS_user.flag_grf y=1;
    end

    Fy=-4e1*VxF(2)-2e3*(PxF(2)-tempy):

    if abs(Fy)>mu*Fz
        Fy=mu*Fz*sign(Fy);
        MBS_user.flag_grf y=0;
    end
    % Implement similar friction in X direction.
else
    % If the swing foot is not in contact with the ground
    Fx=0;
    Fy=0;
    Fz=0;
end
```

7.1.6 Linearization and LQR Control Design

The single support nonlinear mechanical model of the robot is linearized about the upright posture. Modal analysis is performed using the Robotran’s “mbs_exe_modal.m” m-file to compute linearized Mass-inertia and gravity matrices as described in Equation (3.2). Modal analysis is an important built in tool in Robotran that computes the equilibrium point and performs linearization at a certain operating point. Hence, the linearized mass-inertia matrix is
CHAPTER 7. DEVELOPMENT OF A DYNAMIC WALKING SIMULATOR

\[
M = \begin{bmatrix}
2.5527 & 0.0022 & 0.0022 & 0.4520 & 0.0034 & -0.0027 & -0.1011 & -0.0038 & -0.0040 & 0.0012 \\
0.0022 & 2.3381 & 1.056 & 0.0022 & 0.2373 & -0.1029 & -0.0001 & -0.0026 & 0.0029 & 0 \\
0.0022 & 1.3056 & 0.8751 & 0.0022 & 0.4030 & 0.1833 & -0.0001 & 0.0007 & 0.0071 & 0 \\
0.4520 & 0.0022 & 0.0022 & 0.8043 & 0.0034 & -0.0027 & 0.5072 & -0.0038 & -0.0040 & 0.0010 \\
0.0034 & 0.2373 & 0.4030 & 0.0034 & 0.5896 & 0.5054 & -0.0001 & 0.1958 & 0.0117 & 0 \\
-0.0027 & -0.1029 & 0.1833 & -0.0027 & 0.5054 & 0.5054 & -0.0001 & 0.1958 & 0.0117 & 0 \\
-0.1011 & -0.0001 & -0.0001 & 0.5072 & -0.0001 & 0.5043 & -0.0001 & -0.0001 & 0.0010 & 0 \\
-0.0038 & -0.0026 & 0.0007 & -0.0038 & 0.1958 & 0.1958 & -0.0001 & 0.0097 & 0.0071 & 0 \\
-0.0040 & 0.0029 & 0.0071 & -0.0040 & 0.0117 & 0.0117 & -0.0001 & 0.0071 & 0.0029 & 0 \\
0.0012 & 0 & 0 & 0.0101 & 0 & 0 & 0.0101 & 0 & 0 & 0.0012 \\
\end{bmatrix}
\]

and the linearized gravity matrix is

\[
G = \begin{bmatrix}
-50.7891 & 0 & 0 & 8.0756 & 0 & 0 & 13.9464 & 0 & 0 & 0.2042 \\
0 & -50.7891 & -20.9799 & 0 & 8.0764 & 13.9456 & 0 & 4.5486 & 0.2023 & 0 \\
0 & -20.9799 & -20.9799 & 0 & 8.0764 & 13.9456 & 0 & 4.5486 & 0.2023 & 0 \\
8.0756 & 0 & 0 & 8.0756 & 0 & 0 & 13.9464 & 0 & 0 & 0.2042 \\
0 & 8.0764 & 8.0764 & 0 & 8.0764 & 13.9456 & 0 & 4.5486 & 0.2023 & 0 \\
0 & 13.9456 & 13.9456 & 0 & 13.9456 & 13.9456 & 0 & 4.5486 & 0.2023 & 0 \\
13.9464 & 0 & 0 & 13.9464 & 0 & 0 & 13.9464 & 0 & 0 & 0.2042 \\
0 & 4.5486 & 4.5486 & 0 & 4.5486 & 4.5486 & 0 & 4.5486 & 0.2023 & 0 \\
0 & 0.2023 & 0.2023 & 0 & 0.2023 & 0.2023 & 0 & 0.2023 & 0.2023 & 0 \\
0.2042 & 0 & 0 & 0.2042 & 0 & 0 & 0.2042 & 0 & 0 & 0.2042 \\
\end{bmatrix}
\]

The linearized damping matrix \((C = 0)\) is zero since the linearization is done at zero position and velocity. The linearized matrices and the linear actuators and compliance dynamics are then used to build the state space model of Equation (4.1) as described in section 4.2. Equation (4.1) is then used to design the discrete time LQR feedback control.

In addition, the linearization of CoMan be performed in Robotran using only a certain degrees of freedom by “locking” the other joints, as shown in Fig. 7.4.
7.1.6.1 Inverse Dynamics

This option is mainly used in this chapter when a linearization at a certain operating point is required. Inverse dynamics calculations are available via the built-in Matlab function called “mbs_exec_dirdyn.m”. Any configuration of the multibody model can be translated into the corresponding joint torques using the “mbs_exec_dirdyn.m” m-file. In other words, given an upright configuration of CoMan an gravity compensation torque is required to be added to the nonlinear model to keep the posture upright. This torque vector must then be added to the “user_JointForces.m” to be included in the dynamic simulation.

In order to linearize the multibody system about the upright posture (zero angles, velocities and accelerations as shown in Fig. 7.2) inverse dynamics at the equilibrium point is used to compute the torque required to compensate for the gravity and keep
the upright posture of the robot, which is

\[ \tau_0 = [14.016, 0.502, 0.492, 14.105, 0.572, -0.160, 0.178, -0.223, -0.233, -0.001]. \]

\( \tau_0 \) is included in the “user_JointForces.m”. The next section, which linearizes the nonlinear model of CoMan requires \( \tau_0 \) to compute the equilibrium and compute the linearized model.

### 7.1.7 Effects of Quantization and Stiction

The sensor’s quantization noise and joint’s stiction are considered in the simulation. The quantization noise results from the limitation in the digital encoder resolution. In the case of CoMan, the encoder which measures the link position has 12 bits \( (2^{12} = 4096 \text{ pulse/revolution}) \). This resolution leads to the minimum angle detection of \( 2\pi/4096 = 1.5e^{-3} \text{ (rad)} \) which is illustrated in Fig. 7.5. The effect of quantization is more evident when the velocity estimates are computed using first order difference of the quantized joint angles as shown in Fig. 7.6. This effect gets worse as the sampling frequency is increased. The velocity estimates of a sinusoidal signal \( (\sin(5t)) \) are shown in Fig. 7.6 which improve when the sampling time is increased from 1 ms to 10 ms. In fact the minimum velocity that can be detected is \( \frac{\text{quantization level}}{\text{sampling time}} \). However, increasing the sample time is often not desirable and this the reason that it is desired to install high resolution sensors (more than 16 bits) to overcome the measurement limitations and to avoid introducing unnecessary delay due to filtering and averaging in the control system. The pseudo code 7.4 is used in the simulation to investigate this effect.

```matlab
% 12 bit encoder result in minimum angle detection of 1.5 mili-rad.
quantization=1.5e-3;
% The joint angles are rounded by the quantization level.
q.{quantized}=round(q/quantization)*quantization
% Pass the measurement to the observers and feedback calculations
...
```

Pseudo code 7.4: Matlab code for simulating the quantization noise.
Figure 7.5: Magnified illustration of quantization effect on position.

Figure 7.6: Estimated velocity using first order difference of the quantized position measurements.

Stiction is another disturbance that is present in all joints of CoMan. A simple dead-zone model for stiction is used in the simulation which is described in the pseudo code 7.5.
Pseudo code 7.5: Matlab code for simulating the joint’s stiction.

```matlab
deadzone = 0.6; % volts

if abs(u) < deadzone
    u = 0;
end;

if abs(u) > deadzone
    u = u - deadzone * sign(u)
end;
```

### 7.2 Generating the Walking Trajectories

In order to verify the developed dynamic simulator for CoMan, a set of walking trajectories are designed based on the centre of gravity of the robot. In section 7.2.1, the walking trajectories are computed using inverse kinematics by specifying the desired swing foot position in 3D and the desired centre of gravity in 2D (projection of the 3D centre of gravity on the XY plane). In section 7.2.2, a 5th order polynomial is used to smoothly connect the five different postures of the robot. Finally, the designed trajectories are applied to the developed dynamic simulator in section 7.3.

#### 7.2.1 Computing Posture Angles by Inverse Kinematics

Robotran does not provide any tool for the inverse kinematics calculations. Inverse kinematics is an important step toward designing walking patterns in joints space that is often formulated using the standard Newton-Raphson algorithm. However, the standard Newton-Raphson suffers from several limitations such as division by zero (loss of rank in the Jacobian matrix), oscillating around an inflection point, and requires further modification to include linear constraints. Although, it is possible to cater for the mentioned disadvantages of the standard Newton-Raphson algorithm,
that is currently implemented in CoMan’s trajectory generator software, these modifications are beyond the scope of this thesis and Matlab’s nonlinear constrained optimization tool, “fmincon” is used instead to compute the inverse kinematics. “fmincon” provides an easy way of including mechanical limits that must be considered in computing the inverse kinematics without suffering from the standard Newton-Raphson’s drawbacks. However, a disadvantage of this tool is its slower speed that makes it unattractive for online computations. Nevertheless, it serves the purpose of off-line computation of the robot postures in this thesis.

CoMan’s mechanical joints’ limits translated to the Robotran convention are

\[
LB = [-0.4, 0.69, 1, 0.3, 0.3, 1, 0.3, 0.1, 0.69, 0.4] \quad (7.9)
\]

\[
UB = [0.22, 0.69, 0.1, 0.5, 1, 0.3, 0.5, 1, 0.69, 0.22] \quad (7.10)
\]

where \(LB, UB\) denote the lower and upper bound of the mechanical joints’ limits, respectively. These limits are passed to “fmincon” during the inverse kinematic calculations.

The objective function used in the optimisation is

\[
f = \sum_{i=1}^{5} (e_i^2) \quad (7.11)
\]

where \(f\) is the objective function, \(e_1 = x(q) - x_d\), \(e_2 = y(q) - y_d\), \(e_3 = z(q) - z_d\), \(e_4 = COM_x(q) - COM_{dx}\), \(e_5 = COM_y(q) - COM_{dy}\). The first three errors are the difference between the actual and the desired swing foot position in \(X, Y\) and \(Z\) directions. The fourth and fifth errors correspond to the position errors between the actual and the desired projected CoM on the \(XY\) plane.

Finally, there are two constraints imposed on the solution that correspond to keeping the feet parallel to the ground. These two constraints are simply added to the “fmincon” constraint option by setting the sum of lateral angles and sagittal angles to zero, separately.

The walking cycle consists of five stages that are described as follows. In the first stage the robot is at its initial home posture in double support as shown in Fig 7.8. The second posture is used to shift the centre of gravity to the support leg (right leg)
Table 7.2: The joints trajectories for the 5 postures of the walking cycle.

<table>
<thead>
<tr>
<th>Posture 1</th>
<th>Posture 2</th>
<th>Posture 3</th>
<th>Posture 4</th>
<th>Posture 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>$q_1$</td>
<td>0</td>
<td>0.22</td>
<td>0.22</td>
<td>0</td>
</tr>
<tr>
<td>$q_2$</td>
<td>-0.1</td>
<td>-0.0425</td>
<td>0.0515</td>
<td>0.1</td>
</tr>
<tr>
<td>$q_3$</td>
<td>0</td>
<td>-0.0125</td>
<td>-0.0280</td>
<td>0.1795</td>
</tr>
<tr>
<td>$q_4$</td>
<td>0</td>
<td>-0.237</td>
<td>-0.1280</td>
<td>0.0223</td>
</tr>
<tr>
<td>$q_5$</td>
<td>0.1</td>
<td>1</td>
<td>0.2237</td>
<td>0.1289</td>
</tr>
<tr>
<td>$q_6$</td>
<td>0.1</td>
<td>-0.7987</td>
<td>0.0251</td>
<td>0.0193</td>
</tr>
<tr>
<td>$q_7$</td>
<td>0</td>
<td>-0.1715</td>
<td>-0.2446</td>
<td>-0.22</td>
</tr>
<tr>
<td>$q_8$</td>
<td>0</td>
<td>-0.1</td>
<td>0</td>
<td>0.1</td>
</tr>
<tr>
<td>$q_9$</td>
<td>0.22</td>
<td>0.1</td>
<td>0.2237</td>
<td>0.1212</td>
</tr>
<tr>
<td>$q_{10}$</td>
<td>0</td>
<td>0.1</td>
<td>0.0223</td>
<td>0.0193</td>
</tr>
</tbody>
</table>

Table 7.3: The swing foot and CoM positions corresponding to the 5 postures.

<table>
<thead>
<tr>
<th>Posture 1</th>
<th>Posture 2</th>
<th>Posture 3</th>
<th>Posture 4</th>
<th>Posture 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_d$</td>
<td>0.0854</td>
<td>-0.0845</td>
<td>-0.0837</td>
<td>0.0837</td>
</tr>
<tr>
<td></td>
<td>0.1642</td>
<td>0.1651</td>
<td>0.1620</td>
<td>0.1615</td>
</tr>
<tr>
<td>$z_d$</td>
<td>0.0002</td>
<td>0.0045</td>
<td>0.0263</td>
<td>0.0262</td>
</tr>
<tr>
<td>$COM_{xd}$</td>
<td>-0.0501</td>
<td>-0.0011</td>
<td>-0.0016</td>
<td>0.0017</td>
</tr>
<tr>
<td>$COM_{yd}$</td>
<td>0.0906</td>
<td>0.0269</td>
<td>0.0228</td>
<td>0.0227</td>
</tr>
</tbody>
</table>

as shown in Fig. 7.9. The robot lifts the trailing leg at the third posture as shown in Fig. 7.10. Once the trailing leg is lifted the robot swings it forward to take a step while the centre of gravity is maintained in the support foot’s area as shown in Fig. 7.11. Finally, the robot enters the double support by putting the swing leg on the ground as shown in Fig. 7.12, as a mirror reflection of the initial posture shown in Fig. 7.8. This symmetry is used to obtain a periodic walking cycle that is used once the legs are switched for the next steps.

The joints trajectories corresponding to each posture of the robot are given in Table 7.2. The positions of swing foot and the CoM for each posture are provided in Table 7.3.

### 7.2.2 Trajectory Smoothing

Moreover, the five postures defined in Table 7.2 can not be used directly as a set of step functions due to the need for smooth velocity and acceleration on the joint references. Hence, a fifth order polynomial given in Equation (7.12) is used to obtain a smooth set of trajectories that connect the discrete set of postures of the robot. In order to solve for the six unknown coefficients ($c_5, ..., c_0$) in Equation (7.12), the initial and final positions, velocities and accelerations are used. An example of the
polynomial trajectory for the second joint (ankle sagittal) is shown in Fig. 7.7.

\[ p(t) = c_5 t^5 + c_4 t^4 + c_3 t^3 + c_2 t^2 + c_1 t + c_0 \]  
(7.12)

Figure 7.7: An example of the 5th order polynomial used for the stance leg’s ankle sagittal joint.

Figure 7.8: Initial posture of the robot in the walking cycle.
Figure 7.9: In the second posture of the robot, the centre of gravity is shifted to the support leg.

Figure 7.10: The robot lifts the trailing leg.
In the next section, the generated walking trajectories are simulated using the developed dynamic walking environment in Robotran and Matlab.
7.3 Robotran’s Simulation Results

In this section, Robotran and Matlab are used to simulate walking of CoMan with 10 DoF. Sensors are placed on each body’s CoM that are used to compute the trajectory of the robot’s CoM. In addition, a sensor is placed at the tip of the swing leg to detect the cartesian position during the gait. An external force is used to introduce the ground model into the ground. The walking trajectory which was designed in the previous section is simulated. The reference positions at each stage are given in Table 7.2.

The resulting joint angles are shown in Fig. 7.13 where the robot starts the first step with the right leg as the support and during the second step the legs are switched. The periodicity of the trajectories simplifies the switching of the legs which is simply implemented by changing the sign of each angle at the end of a step. A more detailed results on the trajectory tracking is shown in Fig. 7.14, which shows that the control system is following the designed trajectories with a good performance. However, the current trajectories correspond to taking a step every 4.5 seconds. In order to track a faster walking trajectory the control system’s bandwidth must be increased. The 3D visualization of the robot is shown in Fig. 7.15 which shows the projected centre of gravity (the star sign) on the XY plane as well as the history of the robot’s posture during the first step. The corresponding joint torques are shown in Fig. 7.16. It can be seen that by including the dynamics of the full multibody systems the torques required for the walking trajectories are well within the limits of CoMan actuators.

The position of the swing foot, which is computed using the Robotran sensor tool, and the ground reaction force, which is implemented using the external force feature of Robotran are shown in Fig. 7.17. The solid curve (on the top) shows the position of the swing leg at each step. In the first step the trailing leg starts at $-0.1\, \text{m}$ with respect to the stance leg that is at the origin of the world frame. Once the legs are switched the new stance leg starts to move forward from the zero position to $0.2\, \text{m}$. The dash-dotted line is the lateral position of the swing leg that alternates when the swing leg is switched from the left leg to the right leg (with respect to the world
frame). The vertical height of the swing leg is also shown in dashed lines that has a positive value at during the swing time which can be verified also by cross checking the vertical ground reaction force \( F_z \). Moreover, it can be seen that the horizontal friction has changed due to a stick-slip motion on the ground.

Finally, the overall CoM of the robot and the position of the swing leg is shown in Fig. 7.18. It can be seen that the CoM moves constantly in the forward \( X \) direction while the lateral position of the CoM alternates when the support leg is switched between the right and the left leg. At each step the CoM is transported by 0.20 m which corresponds to the forward walking velocity of 0.044 m/sec. In order to increase the walking speed, a more dynamic trajectory must be designed that uses ideas from the ZMP criteria which was discussed in chapter 6. This will be investigated in future research.

![Figure 7.13: The joint trajectories during a full walking cycle (two steps).](image-url)
CHAPTER 7. DEVELOPMENT OF A DYNAMIC WALKING SIMULATOR

Figure 7.14: Joint tracking results on four joints of the stance leg.

Figure 7.15: Visualization of the first step with trace of the trajectories' history.
Figure 7.16: The corresponding joints’ torques during the walking cycle.

Figure 7.17: The cartesian position of the swing foot (top) and the corresponding ground reaction force (bottom).
7.4 Development of CoMan’s Model in MapleSim

An additional line of research was devoted to investigating the potential of using MapleSim multi-domain simulation software, during the later stages of this project. The purpose of this research is to develop a multi-domain walking simulation environment that is more user friendly and faster to modify models of the actuators, ground profile, sensor noise, etc. Moreover, the MapleSim environment has a set of libraries, containing models of electrical actuators and mechanical components, which can be used to build models with different levels of details. In terms of code generation the dynamic and kinematic calculations can be exported to stand alone C code which is important for integrating the results into CoMan’s software.

An initial model of CoMan is developed in MapleSim that is shown in Fig. 7.19. However, the evaluation of the controller design and the models in MapleSim are beyond the scope of this thesis and it will be investigated in more detail in future work.
7.5 Model Validation

In this section, the developed model in Robotran and Matlab is compared against experimental data. The purpose of this comparison is to provide a qualitative validation of the dynamic model. Identification tests will be carried out as future work once the robot’s torque sensors are installed and the hardware reaches the final stages of development. The test used for the comparison is a basic sway test, where the robot sways periodically to each side, while keeping the CoM inside the support polygon. Initially the experimental data is presented. The lateral joints positions are shown in Figs. 7.20. The homing of the links, which is an automatic calibration technique for providing consistent sensory reading as discussed in section 8.9.3, can be seen at the first 8 seconds of this test. The references for the lateral joints are generated using the fifth order polynomial discussed in section 7.2.2. The lateral references alternate between \( \pm 0.1 \) rad. The corresponding experimental voltages for the lateral plane are shown in Figs. 7.21. The same test is simulated in Robotran and Matlab to compare the simulation predictions against this test. The lateral joints’ tracking in Robotran is compared against the experimental data and shown in Fig. 7.22. The major differences between the simulation and the experimental data is the effects of stiction.
during the peak values of the sine wave and also the effect of lower stiffness in the lateral joints. The initial homing is removed from the experimental data to provide a more clear comparison with the simulation. The corresponding motors positions are shown in Fig. 7.23, which has a good agreement between the simulation predictions and experimental data. The largest errors can be seen on the support ankle \( q_{m1} \), which does not seem to be due to the weight of the robot since the robot shifts its weight several times to the right and left leg, while the right leg does not show this much error. Hence, this seems to be more due to a PWM bug at the DSPs which was found recently. This can be confirmed by comparing the lateral voltages for the two ankles in Fig. 7.21. The ankle with the largest mismatch between the simulation and experiment has much more noisy profile in comparison with the voltage of the other ankle. The corresponding simulation voltages are shown in Fig. 7.24. The voltages in simulation has a smaller peak to peak magnitude which is known to be due to the effects of stiction and viscose friction which are present in the robot joints. In addition, the voltages are more noisy in the experiment, as shown in Fig. 7.21, than in the simulation, which is due to the effects of quantization noise as well as noisy velocity estimates which are not present in the simulation. Nevertheless, this section provides an indication that the main features in the experimental data are captured with in the simulation and future system identification tests will further improve the quantitative comparisons. The tests presented in the experimental results chapter point out that stiction, viscose friction and a more accurate estimate of the passive compliance in the joints can significantly improve the simulation predictions.
Figure 7.20: The lateral joints positions during the sway test.

Figure 7.21: The lateral joints voltages during the sway test.
Figure 7.22: A comparison between the simulation and experimental lateral joints positions during the sway test.

Figure 7.23: A comparison between the simulation and experimental lateral motor positions during the sway test.
CHAPTER 7. DEVELOPMENT OF A DYNAMIC WALKING SIMULATOR

7.6 Conclusions and Future Work

In this chapter, a dynamic walking simulator was developed based on the multi-body software Robotran and additional programming in Matlab. Robotran is a useful tool to derive the dynamics and kinematic equations of CoMan, but further programming is needed to include the actuator and compliance dynamics, ground contact model, solving the inverse kinematics, controller design and finally visualizing the numerical results. All these additional features are added to Robotran in this thesis. In order to simulate a walking gait, the passive compass gait model of walking was used as an illustration of how the full hybrid gait can be simulated using Robotran and Matlab. A number of pseudo codes were provided to guide the user on developing a walking simulator.

Finally, a sample walking trajectory based on control of the centre of gravity of CoMan with 10 DoF was designed and simulated in this chapter. Various dynamic and kinematic measurements were done during the simulation of CoMan that was discussed in the simulation results. It was shown that the developed simulator has a good potential to be used for further research on design of compliant walking gaits, provided additional programming is done for the desired features that are not
included in Robotran. A partial validation of the parameters used in the developed model was carried out to provide an idea on how the accuracy the model parameters used. Future work will focus on system identification tests to derive more accurate estimate of the passive compliance and viscose friction in the joints.

An interesting line of research is to investigate the floating base multibody modelling of walking robots. In a floating base model the base body (that is often the robots torso) is free to move in space rather than being fixed to a point as in the case of industrial robotic arms [211]. A floating base for a walking robot is implemented in the MapleSim simulation by adding a 6-DoF joint (consisting of 3 translational DoF and 3 rotational DoF) between the world reference frame and the largest mass of the robot such as the pelvis. A schematic diagram of this idea is shown in Fig. 7.25.

However, it is important to note that there are no force/torques being applied at the 6-DoF floating base joints. Moreover, the additional 6-DoF based is used only for walking simulation, but not for the controller design since the concept of control design for a robot with a floating base is a completely new area of research that has not been studied thoroughly.

![Figure 7.25: Representation of CoMan with a 6 DoF floating base attached between the world frame and the pelvis. The ground reaction forces acting under the feet are only activated when an impact event is detected and the feet height are at the ground level.](image)
In the next chapter, the experimental results of this thesis in terms of successful implementation of the multivariable LQR and reduced order observers on CoMan and the challenges faced during the implementation are discussed. Moreover, partial validation of the model including closed loop bandwidth tests and also validation of the controller and observer results are presented.
Chapter 8

Experimental Results

In this chapter the results of experiments on CoMan are presented. These experiments were conducted during the final year of this PhD project, as the robot was being developed (since 2010), to provide a tracking control system for locomotion. There are several key questions being answered in this chapter. The first question is: given the current existing PID controllers, can the joint tracking performance be improved in terms of bandwidth, robustness and closed loop stability proofs with model based feedback designs? The second question is: How the designed centralized and decentralized LQR feedbacks and observers should be implemented, in order to validate the theoretical results and to make sure every part of the code works as expected? The third question is: given the existing encoder resolution (12 bits), can the velocity estimation be improved in comparison with the existing numerical differentiation and averaging scheme? The forth question is: can a standard set of tests be designed to quantify the tracking performance as well as the control efforts, among different controller and observer schemes? The fifth question is: can the LQR based controller track a fast walking trajectory with good accuracy? It is worth mentioning that several technical challenges were tackled to achieve the answers to these questions, such as C++ programming work for the controller development and logging the data, numerous tests using different feedback gains and observers, and peculiar oscillations caused by inconsistent sensory information, stiction, quantization and a software bug in the DSP software. In addition, during the first two years of this PhD
project a feasibility study on walking with iCub was done, which did not provide a promising outcome, although several important lessons were learnt which are briefly summarized in this chapter and also discussed in [50].

This chapter is organized as follows. In section 8.1, the first experiment of this project on a compliant joint unit is described, which is used to validate the observer based LQR design. In section 8.2, the successful implementation of the observer based LQR feedback is discussed. The experimental results of different velocity estimation schemes are presented in section 8.3. In sections 8.4 and 8.5 two standard tests, namely a sway motion (in lateral plane with high stiffness) and a squat motion (in sagittal plane with lower stiffness) are described, to validate the tracking performance and to quantify control efforts on the ground. The result of a walking trajectory tracking test on CoMan is discussed in section 8.8, which quantifies the LQR tracking accuracy while the robot is being held up on the harness. Finally, the major challenges that were faced during the experiments are discussed in section 8.9, followed by the conclusions and future work.

8.1 Validation of Observer Based LQR on a Compliant Joint Unit

The first step toward implementing the multivariable controllers on CoMan was to test and validate the LQR control design on an independent and smaller scale compliant joint unit which was built as a prototype at IIT [212]. Similar compact and compliant actuators are used in CoMan. In this experiment, the aim was to design a model based LQR feedback to control the motor position mainly to validate the LQR controller design.

The main purpose of this section is to illustrate the modelling, feedback control and observer design issues on a simple prototype. Initially, the hardware is described and the state space model is developed. The eigenvalues of the state space model are then used to choose the sampling time. Subsequently, the choice of the closed loop
poles which are related to the LQR feedback and Luenberger observer are discussed. The experimental data is then used to validate the controller design and to point out the limitations of the linear model. Moreover, several effects such as larger velocity estimates and larger control signals are pointed out that are also present in experiments on CoMan. The most important lesson learnt from these results is to include the friction term (as an addition to the linearized Coriolis and centrifugal term, $C \neq 0$) in the linear models (described in section 3.5.2) for the controller and observer designs. Because as explained further on, including this parameter provides a better model prediction for the experimental data.

8.1.1 Description of the Prototype

The experimental set-up consists of a Kollmorgen RBE1211 motor (A-windings), harmonic drives (CSD 17-2A with reduction ratio of 100), a spring (spring constant is 143 Nm/rad) in series with gearbox stiffness and a load of 1 kg mass on the link. The measurement sensor is a 12 bit encoder that is located on the motor axis. Fig. 8.1 shows the experimental unit which is set up about the vertical $Z$ axis to avoid the effect of gravity ($G = 0$). The electronics for implementing the controller and power driver are based on the Motorola DSP 56F8000 chip with a CAN communication interface. Further details about this prototype unit are provided in [212].

Figure 8.1: Close-up view of the compliant joint set up (left) and the final set up of the equipment in the experiment (right).
Table 8.1: Compliant joint’s model parameters (reflected to the joint side).

<table>
<thead>
<tr>
<th></th>
<th>J</th>
<th>Kg.m²</th>
<th>Ks</th>
<th>Nm/rad</th>
<th>Bm</th>
<th>Nm/sec/rad</th>
<th>Bs</th>
<th>Nm/sec/rad</th>
<th>R</th>
<th>Ohms</th>
<th>Kt</th>
<th>Nm/A</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1502</td>
<td>25.5158</td>
<td>0.0715</td>
<td>0.664</td>
<td>4.1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.0612</td>
<td>0</td>
<td>0</td>
<td>100</td>
<td>6.1747</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

8.1.2 Modelling and Controller Design

The modelling of the unit is carried out as explained in detail in chapter 3, but the equations are repeated for the reader’s convenience. The parameters of the model are provided in Table 8.1. All the model parameters are reflected to the output of the gearbox with reduction ratio of \( N = 100 \) and \( v_t \) is the voltage to torque gain that is obtained by computing \( v_t = K_t R^{-1} \) for the compliant joint’s motor parameters. The damping due to the motor’s back EMF dominates the overall internal damping of the motor, which is \( B_m = K_t K_b / R_m \), where \( K_b \) and \( K_t \) are reflected to the joint side.

\[
\tilde{A} = \begin{bmatrix}
0 & I & 0 & 0 \\
-M^{-1}(G + K_s) & -M^{-1}(C + B_s) & M^{-1}K_s & M^{-1}B_s \\
0 & 0 & 0 & I \\
J^{-1}K_s & J^{-1}B_s & -J^{-1}K_s & -J^{-1}(B_m + B_s)
\end{bmatrix},
\]

\[
\tilde{B} = \begin{bmatrix}
0 & 0 & (J^{-1}K_t R^{-1})^T
\end{bmatrix}^T.
\] (8.1)

Substituting the parameters of Table 8.1 into the Equation (8.1), gives the state space model

\[
\tilde{A} = \begin{bmatrix}
0 & 1 & 0 & 0 \\
-2336.1 & -1.168 & 2336.1 & 1.168 \\
0 & 0 & 0 & 1 \\
951.54 & 0.47577 & -951.54 & -170.35
\end{bmatrix}, \quad \tilde{B} = \begin{bmatrix}
0 \\
0 \\
41.11
\end{bmatrix}
\] (8.2)

where the state ordering is motor and link positions and motor and link velocities, ie. \( \tilde{x} = [q, \dot{q}, q_m, \dot{q}_m]^T \). The input-output polarity between the input voltage and the
output motor position which is a crucial parameter for the feedback loop’s stability is positive as can be observed from the input matrix gain of +41.11. This polarity changes in CoMan as the magnetic encoders on the right and left leg are installed differently which leads to a positive voltage causing a negative reading on one leg and a positive reading on the other leg. However, this can be easily catered for by including the polarities in the feedback design.

The eigenvalues of the open loop system are \([-164.97, -3.2748 \pm 48.937i, 0]\), with a unstable pole at the origin. The three other poles are stable. This is due to the fact the setup is mounted to rotate about the vertical Z axis and gravity does not affect it.

The controller receives the motor position as output measurement and estimates the motor velocity, link position and link velocity using a reduced order Luenberger observer. An integral action is introduced on the motor position. Once the vector of model states and integral state is constructed from estimations and measurements, it calculates the input voltage for the motor. The sampling frequency should be chosen at least 20 times faster than the bandwidth frequency of the closed loop plant. The desired bandwidth of the system is about 2 Hz. Hence, the controller and observer are designed in discrete time with sampling time of 10 ms (sampling frequency of 100 Hz).

Choosing the LQR penalties requires a number of trials and simulations. In summary, one wants a fast response (desired closed loop bandwidth) with little overshoot (less than 10-20%) and a reasonable control input to avoid actuator saturation. As explained in section 4.2.2, the penalties for this experiment are \(Q=\text{diag}[1, 1, 1, 1, 40]\) and \(R = 20\) and the LQR feedback is

\[
F = \begin{pmatrix}
-1.45 & 0.072158 & 36.683 & 0.20803 & -1.3525
\end{pmatrix}
\]

which gives the closed loop eigenvalues of \([0.1917, 0.96557, 0.96557, 0.95956, 0.95956]\) and has a rise time of 0.36 second and settling time of about 1 second. In order to implement the LQR state feedback, the link position and velocity estimates of the motor and the link are required. This problem is addressed in the next section by
designing a reduced order observer.

8.1.3 Reduced Order Observer Design

A reduced order observer is designed to estimate the link’s position, velocity and the motor’s velocity. The inputs to the observer are the motor position (measured with a 12 bit encoder) and the control input (motor voltage). The theory of Luenberger observer, discussed in section 4.5, is used to design this observer. According to the principle of separation of estimation and control [192], the observer poles can be placed once the state feedback is designed. The observer’s poles will be additional poles of the closed loop system. Fast observer dynamics are more sensitive to quantization. The important point about placing the observer’s poles is not entirely determined by the speed of convergence. In fact the relative stability of the closed loop system must be verified to ensure the control system has good stability margins (GM and PM). In this experiment, the reduced order observer’s poles are placed at $[0.7, 0.8, 0.9]$ which are chosen both to reduce the effect of quantization noise and to ensure good stability margins. The PM and GM of the closed loop system with the feedback and the observer are $76.9^\circ$ and $24.8$ dB, respectively, which are shown in Fig. 8.2.

Therefore, the observer gains are

$$E = \begin{pmatrix} 0.00074 & 16.224 & 1.0577 \\ -0.009861 & 0.95623 & -0.01203 \\ 0.039762 & -6.4533 & 0.54303 \end{pmatrix}, \quad G_o = \begin{pmatrix} -7.1542 \\ 2.8561 \\ 5.6755 \end{pmatrix}, \quad H_o = \begin{pmatrix} 0.27124 \\ -0.003124 \\ 0.29373 \end{pmatrix}, \quad K_o = \begin{pmatrix} -207.93 \\ 2.4851 \\ -79.657 \end{pmatrix}$$

and the discrete time observer equations are

$$\dot{x}(k) = w(k) + K_o q_m(k)$$

$$w(k + 1) = E w(k) + G_o q_m(k) + H_o u(k)$$

where $w$ is the observer’s internal state and $\dot{x} = [q, \dot{q}, \dot{q}_m]$. The experimental data and the corresponding simulations are presented next.
8.1.4 Simulation and Experimental Results

Simulation and experimental results of applying a sine wave reference signal to the compliant joint unit are presented in this section. All the graphs presented in this section assume that state space variables are reflected to the link side using the gearbox reduction ratio of 100 between the motor and the link.

8.1.4.1 Tracking Performance

Since this was the first experiment to validate the implementation of the observer based LQR, a slow sine wave was chosen with the period of 12.5 seconds (frequency of 0.5 rad/sec) to avoid damaging the equipment. The sine wave tracking experimental data is illustrated in Fig. 8.3. The motor position has a time lag of 0.27 second, compared to the reference position. The Bode diagram for the closed loop system is shown in Fig. 8.4 where the frequency of the input signal is depicted on the frequency response. The bode diagram predicts 7.63 deg (0.1332 rad) phase lag at the sine wave frequency of 0.5 rad/sec which corresponds to 0.2664 second time lag between the
motor position and the input reference signal. The predicted time lag of 0.2664 second, closely resembles the experimentally observed time lag of 0.27 second.

Figure 8.3: Sine wave tracking experiment on compliant joint (Experiment).

Figure 8.4: Bode diagram of the closed loop system with the reference signal as the input and the motor position as the output (Simulation).

The corresponding experimental input voltage to the motor is shown in Fig. 8.5
and the predicted input voltage from the simulation is depicted in the same figure. The first notable difference between the simulation prediction and the experimental result is the larger input voltage in the experiment compared to the corresponding simulation. It is well known that electro-mechanical systems have different amounts of stiction which is also present in the compliant joint unit used in this experiment. The amount of stiction in this joint was determined by applying an increasing voltage until the motor starts moving which was measured as 0.8 volts. Hence, by including the effects of stiction as a simple dead-zone and viscose friction (as the Coriolis damping term in the joint) the magnitude of the voltage compares better to the experiment as shown in Fig. 8.6. Similar pattern can be seen in the simulation prediction and the experimental data. The linearization of the Coriolis and centrifugal forces about the origin is zero ($C = 0$). The viscose friction can be added to the linearized Coriolis term as $C = 30$, which gives

$$\tilde{A} = \begin{pmatrix}
0 & 1 & 0 & 0 \\
-2336.1 & -501.168 & 2336.1 & 1.168 \\
0 & 0 & 0 & 1 \\
951.54 & 0.47577 & -951.54 & -170.35
\end{pmatrix}. \quad (8.4)$$

This observation is confirmed in the next section, where the same amount of damping in the joint produces a better match for the velocity estimate signals.

Figure 8.5: Comparison between the experimental and simulation input voltage corresponding to the sine wave tracking.
8.1.4.2 Validation of the Observer Estimates

The available measurement on the compliant joint prototype is an encoder which measures the motor’s position. Hence, a reduced order Luenberger observer is used in this experiment to estimate the remaining states which are the link’s position and velocity as well as the motor’s velocity. In this section, the estimated signals from the observer are verified against the expected signals.

Initially the estimated motor velocity is studied. The motor position shown in Fig. 8.3 is differentiated to obtain an estimate of the motor’s velocity. The estimated motor velocity computed by the Luenberger observer is then scaled down by a factor of 100 and compared with the estimated motor velocity (derived by numerical differentiation) in Fig. 8.7, which shows the profile of the motor velocity estimate given by the observer resembles the estimated velocity signal computed by numerical differentiation of the motor’s position. The factor of 100 is not due to the gearbox reduction ration as this is already included in the controller and observer design (everything is reflected to the link side).

The estimated link velocity, \( \hat{q} \), and estimated motor velocity, \( \hat{q}_m \), are compared in Fig. 8.8, where both variables are reflected to the link side. It is seen that the magnitude of the estimated velocities are much larger than the velocity derived...
by numerical differentiation. This effect is reproduced in simulation by including a stiction of 0.8 volt (modelled as a dead-zone) and adding viscose friction as described in Equation (8.4). The result is shown in Figs. 8.9 and 8.10.

Moreover, at the point where the estimated velocity computed by differentiation crosses zero a region can be seen where the velocity stays at zero for less than a second and then starts increasing. This point on the velocity graph correspond to the peak of sine wave which is distorted by the stiction and the velocity staying at zero for a short time confirms this effect.

Nevertheless, the observer’s estimated velocity is less noisy than the velocity estimate obtained by differentiation and it can prove important once higher bandwidths are demanded from the control system.

Figure 8.7: Comparison between estimated motor velocity derived by the observer and numerical differentiation (Experiment).
CHAPTER 8. EXPERIMENTAL RESULTS

Figure 8.8: Comparison between estimated motor and link velocities computed by the observer (Experiment).

Figure 8.9: Reproducing the experimental observer’s estimated motor’s velocity computed in simulation by including stiction and viscous friction.
Investigating the observer’s link position estimate, shows a large difference between the measured motor position and estimated link position. Although the link position measurement was not available in this unit, the reflected motor position to the output of the gearbox can provide an indication that the estimated link position is larger than the expected link position, as shown in Fig. 8.11. The expected link position should be relatively close to the motor position which is measured in the experiment (considering the small deflection of the spring).

Figure 8.10: Reproducing the experimental observer’s estimated link’s velocity computed in simulation by including stiction and viscose friction.

Figure 8.11: Comparison between estimated link position and real motor position (Experiment).
The larger position estimates are explained by including the viscous friction as given in Equation (8.4) and stiction of 0.8 volts (modeled as a dead-zone). The simulation results are shown to be closely resembling the experimental data in Fig. 8.12. Including a unique value for the stiction and viscous friction that improves all the simulation predictions to match the experimental results strengthens the idea that the effects seen in the experiment are indeed related to stiction and friction effects.

Figure 8.12: Comparison of experimental link position estimate with the simulation prediction with stiction and viscous friction included.

The effects reported in this section such as larger control signals and larger velocity estimates are also observed in the experiments on CoMan, while CoMan has extra sensors to measure both the motor and link positions. In the next section, an experiment on 10 DoF of CoMan’s legs is described, where the bandwidth of the control system is measured experimentally to validate the theoretical bandwidth predictions.

8.2 Implementation of Observer Based LQR on CoMan

In this section, the experimental data from the first successful implementation of the LQR and observers on CoMan is presented and analyzed. The main purpose of this section is to explain the details of the centralized LQR implementation as well as
quantifying the bandwidth. It is worth pointing out that the current LQR bandwidth (which is quite conservative for safety reasons) can be increased if required. The results presented in this section were achieved after several important lessons were learnt from the challenges explained in sections 8.1 and 8.9. The controller design in this section is generic and has been used throughout the rest of the centralized LQR experiments discussed.

This section is organized as follows. Initially, the significant steps required for preparing the robot for a test is discussed. Then the modelling and controller design parameters are explained. However, since the model of the robot has high dimension (50 state variables), the linear models are made available for download from [1]. Finally, the bandwidth for various LQR designs are quantified. The achieved bandwidth is related to the feasible walking speed.

8.2.1 Homing the Joints’ Positions

The first step toward implementing the controller and observer on CoMan was to identify the input-output polarities (voltage to position). The second step is to carefully calibrate the motor and joint encoders consistently. This is referred to as the homing method, which is applied in the beginning of each experiments. A feedback technique is used to drive the joints to the zero positions and then the offset between the motor positions and the link positions are removed. The feedback gains for homing are chosen to have a low bandwidth of 0.6-0.8 Hz. The corresponding LQR penalties for designing the homing feedback are $Q = \text{diag}\{2500 \times 1, 0, 1\}$ and $R = 20 I_{10}$, with $1$ being a row vector of ones with 10 elements and $0$ being a row vector of 30 zeros. The integrators are closed on the link positions.

Homing is an important step in implementing the LQR controller with feedback from multiple sensors. Several challenges were faced in the experiments (discussed in section 8.9) which led to development of this technique. The result of this work is presented in Appendix E, in Tables E.1, E.2 and E.3. Having derived the calibration data presented, a line is fitted to the data to convert the encoder counts to radians.
for the control system as

$$\theta(k) = \theta_0 + \Delta \theta \frac{c(k) - c_0}{\Delta c} \quad (8.5)$$

where $\theta_0$ is the minimum mechanical limit of a joint in radian, $\Delta \theta$ is the range of motion of the joint in radian, $c(k)$ is the sensor reading in counts at sample $k$, $c_0$ is the sensor reading in counts corresponding to the joint being at $\theta_0$ and $\Delta c$ is the total number of counts that the encoder produces for the range of motion.

The type of errors that often occurs in calibration are called the zero and span errors. The zero error is where the line in (8.5) crosses the vertical axis which is related to $\theta_0$ being different from the actual value due to various sources of error in an experiment including the human factor. The other error in calibration is referred to as the span error which is essentially the slope of the line in (8.5). The span error is affected by errors in deriving the precise $\Delta \theta$ and $\Delta c$. Both type of errors are also present in experiments carried out in this PhD project but repeated trials of deriving the calibration data is done to improve the accuracy of sensor readings. In summary, the homing technique can be thought of as having a single origin for the motor and the joint calibration lines, by removing the offset between them. Once CoMan’s torque sensors are installed the same procedure must be applied to the torque sensor data to achieve a consistent calibration among the joint position, motor position and joint torque measurements.

### 8.2.2 Modelling and Control Design Details

In this section, details of modelling and control system design (LQR feedback and observer) are provided which is used in the implementation of a sway motion as discussed in the next section. The mechanical model of CoMan is derived using Robotran and linearized about the upright position (zero angles, velocities and accelerations). The resulting linearized mass-inertia and gravity matrices are given in Appendix C in Equations (C.1) and (C.2), respectively. The linearization of the Coriolis and centrifugal forces about the origin is zero and viscose friction is not included in the model.
Table 8.2: The tracking bandwidth corresponding to various LQR input penalties.

<table>
<thead>
<tr>
<th>Input penalty $R$</th>
<th>20 $I_{10}$</th>
<th>4 $I_{10}$</th>
<th>2 $I_{10}$</th>
<th>$I_{10}$</th>
<th>0.5 $I_{10}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bandwidth (rad/sec)</td>
<td>3.8-5.5</td>
<td>5.5-7.0</td>
<td>6.3-7.5</td>
<td>6.8-8.1</td>
<td>7.1-8.8</td>
</tr>
</tbody>
</table>

In order to build the overall linear compliant model of CoMan the actuator dynamic parameters are provided in Appendix C in Table C.2. A symmetry between the first and last five joints can be seen in the table which results from the similar drives being used for the right and left legs.

The full linear model is integrated according to the details provided in section 3.5. The full dimension of the state space model without the integrators is 40 which consists of positions and velocities of 10 joints and 10 motors. The resulting model is discretized with 2 ms sampling time interval. The integrators are then included on the motor positions. Once the controllability and observability of the model is verified, the next step would be to design the LQR feedback and the reduced order observer to estimate the motor and joint velocities.

The tracking bandwidth was increased gradually, in several experiments, starting from more conservative (low) bandwidths to higher bandwidths. The main parameters that can be varied to modify the tracking bandwidth are the LQR feedback penalties $Q$ and $R$. The relative norm of the input penalty matrix with respect to the states penalty matrix $Q$ is used as the main idea for tuning the LQR feedback. The LQR feedback state penalty, which were fixed in several tests, was $Q=\text{diag}\{v_1, 0, v_2\}$, where $v_1 = 2500 \times 1$ and $v_2 = 1$. The input penalty was then chosen to be $R = a \ I_{10}$ where $a$ was subsequently chosen to be 20, 4, 2, 1 and 0.5 with $a = 0.5$ being the fastest design which is used in the section. The bandwidths corresponding to each input penalty are provided in Table 8.2. Therefore, the final LQR feedback penalties are $Q=\text{diag}\{v_1, 0, v_2\}$, $R = 0.5 \ I_{10}$. The penalties correspond to penalty of 0.5 on all control inputs, penalty of 2500 on all 10 joint positions and penalty of 1 on all integrator states.

Fig. 8.13 shows the frequency response of the closed loop tracking system (with
the reference as input and motor positions as the output). The bandwidth for the 10 joints is between 7.1-8.8 rad/sec (1.13 - 1.4 Hz). Furthermore, the LQR feedback is almost decoupled between the lateral and sagittal planes. Fig. 8.14 shows the lateral feedback gains versus the joint and motor positions and velocities (integrators are not included in these plots). The plots on the right refer to the right leg and the ones on the left refer to the left leg. It can be seen that the LQR feedback for the lateral joints are mostly acting on the lateral state variable (joints or motors). The texts presented on the plots refers to the state variable number and the corresponding feedback gain. For instance, the left lateral joint feedback gain mainly relies on the feedback from the 30\textsuperscript{th} state variable with the feedback value of 110, which is shown as (30, 110). Fig. 8.15 illustrates the feedback gains versus the states for the sagittal plane joints. Similar idea is evident in this figure that feedback for sagittal plane variable are taken mainly from the sagittal motors and joints state variables. Hence, the standard and repeatable sway and squat tests described in sections 8.4 and 8.5 can be justified to be good choices for separately examining the control performance in the two planes.

It is well known that observer design is the dual of feedback design problem. Hence, in order to avoid numerical problems discrete time LQR algorithm was used to optimally place the observer poles. The penalties used for the observer pole placement are $Q = I_{20}$, $R = 0.1 I_{20}$. Further details as well as the experimental results of this observer, which is referred to as the slow observer, are discussed in section 8.3.3.

In order to investigate the stability margins of the feedback with the observer the determinant of the frequency response of $F_o = I + G(z)G_c(z)$ at the output of the plant is plotted where $G(z) = C(zI - A)^{-1}B$ is the plant’s discrete time transfer function and $G_c(z) = G_c(z) = C_c(zI - A_c)^{-1}B_c + D_c$ is the controllers transfer function which is computed in two cases. In the first case the controller is considered without the observer which is the constant LQR feedback gain. In the second case the observer’s dynamics are included with the LQR feedback gain. The overall controller
Figure 8.13: Closed loop bode magnitude plot of the 10 joints with the horizontal axis being the frequency in (rad/sec).

Figure 8.14: Diagram of the lateral LQR Feedback Gains (FBG) versus state variables.
Figure 8.15: Diagram of the sagittal LQR Feedback Gains (FBG) versus state variables.

The state space \((A_c, B_c, C_c, D_c)\) is

\[
A_c = \begin{pmatrix}
E - H F_{L2} & -H F_3 \\
0 & I_{10}
\end{pmatrix},
B_c = \begin{pmatrix}
G_o - H F_{L1} \\
[0, I_{10}]
\end{pmatrix},
C_c = -[F_{L2}, F_3], D_c = -F_{L1}
\]

as explained in section 4.5.4.

Both cases are plotted in Fig. 8.16 where the first case is distinguished with dashed lines from the second case which has a solid line. It is evident that the solid line has less relative margin with respect to the critical point (origin) in comparison with the dashed line where the observer dynamics are not included. Nevertheless, the plot provides an indication on the relative margin of the overall closed loop multivariable system. Knowing the guaranteed margins of LQR feedback (shown with dashed lines) the observer based LQR shows good margins relative to LQR margins.

The next section discusses the experimental velocity estimation results obtained by applying various observer estimation methods to the robot.
8.3 Velocity Estimation

A set of tests was carried out on CoMan with the aim of improving the velocity estimation. These tests were motivated by the oscillatory responses observed in some of the experiments which raised the question: if improving the velocity estimation, can resolve the oscillation problem. The low and high frequency oscillation problems faced are further discussed in section 8.9. Three different types of velocity observers were implemented on CoMan. These methods include a Luenberger observer (slow and fast), an unknown input observer and the traditional difference and averaging scheme. Among these methods the slow Luenberger observer and the difference and averaging scheme provided stable and desired results, which are presented next.

8.3.1 Experimental Set-up

The robot is lifted using a harness. The LQR feedback used is discussed in section 8.2. No feed-forward gain is used in these experiments. It should be noted that the sampling time is 2 ms. The control voltages are limited to ±6 volts for safety considerations. The maximum voltage value of 6 volts is sufficient for these tests. Specific details about each scheme is given in the corresponding sections.
8.3.2 Difference and Averaging

In this section, the experimental data from numerical differentiation with averaging is presented. Averaging is done over the last 10 samples. The main purpose of this experiment is to verify the performance of the control system with velocities computed using differentiation.

The reference trajectory to the hip lateral joints are given as a series of steps and the resulting position tracking is shown in Fig. 8.17. There is a slight overshoot in the transient, which will be dealt with by generating smooth trajectories for walking tests. The corresponding estimated velocities are shown in Fig. 8.18. The effect of quantization can be seen clearly on the graph. The motor signal has a higher resolution encoder in comparison with the joint signal, which results in a more accurate velocity estimate for the motors (the resolution is 100 times better). The corresponding control signals are also shown in Fig. 8.19. In order to see the effects of quantization better, close-up view of the right hip lateral position, velocity and voltage are shown in Figs. 8.20, 8.21 and 8.22. Although during the first 6 seconds the joint is not moving and the input reference is zero, the joint encoder produces some small pulses (0.00197 rad) that when differentiated and averaged over the last 10 samples leads to several velocity spikes (0.0767 rad/sec) and subsequently a spike of 0.31 volt in the control signal, which is within the stiction level and does not produce a motion on the joint. At $t = 6$ sec, when the step signal is applied, the control signal hits the software programmed saturation level of 6 volts as shown in Fig. 8.22, which is partly due to large velocity estimate derived by numerical differentiation. In addition, averaging over the last 10 samples leads to a delay of 20 ms in the control loop, which is of course undesired. It is worth mentioning that the motor encoder does not have this feature since its encoder has the benefit of the gearbox reduction ratio. This feature is more significant in the data when the joint is moving slowly as can be seen in the portions of the time series data that correspond to steady state. The next section, presents the results of implementing a Luenberger observer on CoMan.
Figure 8.17: The hip lateral steps with LQR control and velocities computed by difference and averaging.

Figure 8.18: The hip lateral velocities computed by difference and averaging.
Figure 8.19: The hip lateral voltages.

Figure 8.20: The effects of quantization on the right hip lateral position.
Figure 8.21: The effects of quantization on the right hip lateral velocity.

Figure 8.22: The effects of quantization on the right hip lateral voltage.
8.3.3 Luenberger Observer

In this section, the experimental data from a slow and a fast Luenberger observer are presented. Initially, the results from the slow observer, which was discussed in section 8.2, are presented. Fig. 8.23 shows the motor position, link position and input reference. The homing is done during the first 2 seconds when difference and averaging is used and afterwards, the velocity estimation is switched to the slow observer. The corresponding motor and link velocities are shown in Fig. 8.24. It can be seen clearly that once the homing is done, the slow observer produces an estimate for the velocity. However, as explained in section 8.1, the velocity estimates are larger mainly to the stiction and viscose friction in the joint. The corresponding voltage is shown in Fig. 8.25. The voltage produced is driving the motor to the zero reference position and corrects the initial error can be seen in Fig. 8.23. The 10 largest eigenvalues of the slow observer in increasing order are $[0.6907, 0.7212, 0.7664, 0.8387, 0.9312, 0.9739, 0.9839, 0.9910, 0.9913, 0.9937]$. Based on the largest eigenvalue of 0.9937 the observer has an approximate time constant of 158 sample (0.3 sec).

Furthermore, the fast observer is derived using the LQR algorithms with the state penalty of $Q = I_{20}$, $R = 0.02 \ I_{20}$. The 10 largest eigenvalues of the fast observer are
CHAPTER 8. EXPERIMENTAL RESULTS

Figure 8.24: The left hip lateral velocity obtained using difference and averaging during homing and the slow observer after the homing.

Figure 8.25: The left hip lateral voltage with velocities derived using the slow observer.
[0.6906, 0.7206, 0.7659, 0.8382, 0.9310, 0.9706, 0.9793, 0.9808, 0.9840, 0.9859]. Based on the largest eigenvalue of 0.9859 the time constant of this observer is approximately 70 samples (0.14 sec). It is well-known that a fast observer is not necessarily a better velocity estimator due to degradations in the relative stability and increased sensitivity to measurement noise. In this experiment the robot must regulate its joint positions to zero. The sagittal plane joints after using this observer are shown in Fig. 8.26. Homing is done in the first 8 seconds and the robot is held up during the first 40 seconds where oscillations can be seen on both right and left knee joints. Once the robot is placed on the ground (at $t = 40$ sec) the amplitudes of the oscillations grow. The corresponding velocity estimates are shown in Fig. 8.27. Large velocity estimates during the first 40 seconds when the robot is held up in the air is evident in the graphs. The corresponding sagittal plane voltages are shown in Fig. 8.28. A similar pattern can be seen among the joints with oscillatory position and the voltages. In the case of joint 2, it can be seen that before $t = 40$ sec although the position is hardly changing, the velocity estimate and the voltages are both oscillating. The fast observer has poorer relative stability and also due to fast dynamics it amplifies the measurement noise. However, the next section describes the results of a test with an unknown input observer which has the same relative stability as the LQR state feedback, but the results are oscillatory. Therefore, the issue with relative stability seems not to be the cause for this problem and amplification of the measurement noise is more likely to cause this behaviour. Because both the fast observer and the unknown input observer behave like a differentiator. Hence, the slower observer, which provides a more reliable and stable control system, was chosen for the subsequent experiments.

8.3.4 Unknown Input Observer

In this section, the experimental data from the unknown input observer is presented. This method was tested on the robot to tackle the stiction which caused low frequency and low amplitude oscillations as discussed in section 8.9.4. In this experiment the
Figure 8.26: The sagittal plane joint positions with velocities derived with the fast observer.

Figure 8.27: The sagittal plane joint velocities derived with the fast observer.
Figure 8.28: The sagittal plane motor voltages with velocities derived using the fast observer.

robot must regulate its joint to zero initially while held up in the air and subsequently when placed on the ground. Fig. 8.29 shows the sagittal plane joints, during which the robot is held up until $t = 145$ sec and then placed on the ground, when the oscillations increase. In particular the ankles and the knees are oscillatory. The corresponding velocities are shown in Fig. 8.30. It can be seen that the estimated velocity for the right hip joint which is not moving is oscillating with a peak to peak value of 2.5 rad/sec which increases once the robot is placed on the ground. Fig. 8.28 shows the corresponding sagittal plane voltages.

The relative stability of different schemes were computed according to the theory presented in section 4.5.4 and plotted in Fig. 4.21. Although the unknown input observer recovers the state feedback stability margins, in practice it does not leads to a promising outcome. As mentioned earlier, amplification of the measurement noise by the unknown input observer can be reduced by use of a suitable filter. Kalman filters or newly revised differentiation methods [214] promise more robustness in face of quantization noise for velocity estimation. As future work these methods will be studied to further improve the velocity estimation.

As a result of the extensive tests, the slow Luenberger observer is chosen as a
Figure 8.29: The sagittal plane joint positions with velocities derived with the unknown input observer.

Figure 8.30: The sagittal plane joint velocities derived with the unknown input observer.
more promising estimation method, at this stage. In the next section, the controller is implemented successfully in an experiment on CoMan to carry out a sway motion on the ground.

### 8.4 Implementation of Sway Motions Using the LQR Control

A sway test is the simplest repeatable experiment that can be used to test the performance of different controller designs in the lateral plane. A feature of the sway test on CoMan is that it does not excite the springs, since the lateral joints, which have higher stiffness (only the gearbox stiffness), can be tested independently of the sagittal joints with lower stiffness. Hence, the sway test is chosen as a standard test for CoMan to quantify the tracking performance and the control efforts, among different controllers. Fig. 8.32 shows the snapshots of a sway experiment performed on CoMan with an addition of 4 kg mass on top.
8.4.1 Sway Motion Experimental Set-up

In this section, the designed feedback (discussed in section 8.2) is implemented on CoMan legs to track a lateral sway on the ground. No feed-forward gain is used $G_{ff} = 0$, as the transient response has a slight overshoot (less than 10%) as explained in the results section. In the first 8 seconds the homing of the links is done while the robot is lifted on a crane. The process of homing involves regulating the links to zero position using a low bandwidth feedback gain (for safety). After 8 seconds which is well within the settling time of the low gain feedback, the offset between the links and motors positions are removed from the sensor readings to improve the calibration and consistency of the sensory information. Subsequently the robot is lowered to the floor and the high gain feedback is switched on to implement the sway motion. In all plots presented in this section the right hand side graphs refer to the right leg and the left hand side graphs refer to the left leg.

8.4.2 Sway Motion Experimental Results

The joints tracking is shown in Fig. 8.33. The step references have the magnitude of 0.03, 0.05 and 0.1 for all the lateral joints. Although homing is done at the beginning of the experiment and the lateral joints have high stiffness, a slight difference can be observed between the motors and joints positions. The tracking errors of this test are shown in Fig. 8.34. The error for each joint is $e_i = q_{mi} - r_i$, where $i$ is the joint’s index. Apart from the first 8 seconds when the homing is done, the rest of the errors
only occur shortly once a step reference is applied to the system and no steady state errors are present due to the controller’s integral action.

The corresponding control inputs (motor voltages) are shown in Fig. 8.35. The fast transients in the voltages occur at the time of a step command, which also yields an overshoot of less than 10% of the step size. In addition, the magnitude of the control signals are well within 3 volts while the maximum voltage that can be applied to the motors is 24 volts. Because part of these voltages are lost in the stiction, the sway test is not a demanding task, and is hence a suitable candidate for an initial repeatable test of a new control design.

Moreover, the estimated motor and joint velocities are shown in Fig. 8.36. Both velocities are estimated by Luenberger reduced order observer. Since the friction term inside the motor is included in the state space model the motor velocity provides a better estimate of the actual velocity and it indicates that the joint velocities are relatively large in comparison with the motor velocities which is reflected to the joint side. Nevertheless, the position control task is achieved in this experiment and the control system works with good robustness to parameter uncertainties such as friction on the joints which is not included in the state space models.

During the sway motion experiment the sagittal joints are regulated to zero and
Figure 8.34: The tracking errors of the sway test.

Figure 8.35: Experimental data from the right ankle lateral.
this is verified by the experimental data as shown in Fig. 8.37. The plots on the left side illustrate the left leg’s sagittal joints and the plots on the right side illustrates the right sagittal joints. The initial transient data corresponds to the homing of the joints that is performed at the beginning of each experiment. The knee joint positions are shown to be slightly deviated from zero due to their low compliance and low friction characteristics. The next section, discusses the second repeatable test which can be done on the sagittal joints with out affecting the lateral joints.

8.5 Implementation of Squat Motions Using the LQR Control

A squat test is the simplest repeatable experiment that can be used to test the performance of different controller designs in the sagittal plane. A squat test on CoMan directly excites the springs in the sagittal joints. Hence, this test can be used to quantify the tracking performance and the control efforts, on the joints with high compliance. Fig. 8.38 shows snapshots of a squat experiment performed on CoMan.
Figure 8.37: Sagittal joints regulated to zero during the sway motion.

Figure 8.38: Snapshots of CoMan doing a squat test (from left to right).
8.5.1 Squat Motion Experimental Set-up

In this section, the feedback and observer designed in section 8.2 are used to test three squat motions on CoMan. The feed-forward gain of $0.5 G_{ff}$ is used to speed up the transient response. In the first test, a slow (in 130 seconds) squat motion is performed which consists of three steps of $r_1 = [0, 0.1, -0.2, 0, 0.1, 0, -0.2, 0.1, 0]$, $r_2 = 2 r_1$ and $r_3 = 3 r_1$ radian in forward and reverse order asking the robot to bent the knees to the maximum of $-0.6$ rad. The transition between the three steps of the trajectory $r_1$, $r_2$ and $r_3$ is done smoothly using a ramp to connect the steps. The aim of this test is mainly to investigate the steady state tracking performance and control signals during a more torque demanding task such as a squat.

In the second test, a 2 kg mass is added to the waist and robot is placed on the floor and the same process of the first test is repeated (over 130 seconds) to investigate the robustness of the control system experimentally. In the third test, the 2 kg mass is kept on the waist and the squat motion is speeded up to complete the squat motion in 5 seconds. The aim of faster squat motion is to investigate the performance of the control system under more dynamic motions. In all plots presented in this section the right hand side graphs refer to the right leg and the left hand side graphs refer to the left leg.

8.5.2 Squat Motion Experimental Results

The first set of graphs discussed are related to the slow squat experiment carried out over a period of 130 seconds without the extra 2 kg mass. The tracking performance of sagittal joints is shown in Fig. 8.39 over 130 seconds while the lateral joints are regulated to zero. The corresponding tracking errors are shown in Fig. 8.40. The largest spring deflection (difference between the link and motor positions) can be seen on the ankles which carry the most during the squat test. The errors are $e_i = q_{mi} - r_i$, where $i$ is the joint’s index. There are some oscillations on the knee and ankle joints which is due to the lower stiffness of the joint. However, the tracking errors between the knees’ motors’ positions and the input references also show some oscillations.
which are most likely due to the PWM bug in the DSP software which was recently found. This software bug causes the joint to oscillate if the loading is in the direction of the control input. If the loading is against the control input the PWM software works correctly. This software bug is now fixed on the CoMan’s DSPs.

The corresponding observers’ estimated motor velocities during the experiment are shown in Fig. 8.41 and the sagittal input voltages are shown in Fig. 8.42. The peak to peak values of the input voltages are larger in comparison with the sway test which shows that this test is more demanding.

The next experiment is similar to the previous test with the addition of 2 kg mass on the waist while this change of parameter is not modelled in the control and observer design and the same control system is used to implement this motion. The sagittal motor and joints positions are shown in Fig. 8.43. The tracking errors are shown in Fig. 8.44. The same oscillation problem on the motors’ positions can also be observed in this experiment. The corresponding motors’ velocities estimates and input voltages are shown in Figs. 8.45 and 8.46, respectively.

In the third experiment, the same squat motion is repeated with a faster trajectory and the results are shown in Figs. 8.47, 8.48, 8.49 and 8.50. The data before
Figure 8.40: Sagittal joints tracking errors of the first squat test.

Figure 8.41: Estimated motors’ velocities of the first squat test.
Figure 8.42: Input voltages of the first squat test.

Figure 8.43: Sagittal joints tracking of the second squat test with 2 kg added on the waist.
Figure 8.44: Sagittal joints tracking errors of the second squat test with 2 kg added on the waist.

Figure 8.45: Estimated velocity of motors for sagittal joints during the second squat test with 2 kg added on the waist.
Figure 8.46: Input voltages for sagittal joints during the second squat test.

$t = 100 \text{ sec}$ does not show the oscillations on the tracking errors and the velocities. However, after $t = 100 \text{ sec}$, when the robot is repeating a squat motion for the third time, the oscillations appear which points to the problem which was recently found in the DSP’s PWM software. Another point is the control signals which are less noisy in comparison with the slow squat motions and have a peak to peak value of 5 volts which is comparable to the slower tests. Addition of the 2 kg weight does not show a significant difference in the tests which in addition to showing the robustness of the control system in terms of parameter variation, it shows that this scheme can provide a good solution once the upper body is assembled. Of course, in this case the mass and inertia of the upper body will be included in the control design, while such variations are expected to be tolerated by the control system.

8.6 Sway Tests Using the PID Control

The tracking performance and control efforts of the existing independent PID control scheme is discussed in this section. In fact, these tests provided some guidelines for implementation of the centralized LQR tests which had faced the homing technical challenge. The results of these tests were used to work backward from an existing
Figure 8.47: Sagittal joints tracking of the third squat test with 2 kg added on the waist.

Figure 8.48: Sagittal joints tracking errors of the third squat test with 2 kg added on the waist.
Figure 8.49: Estimated velocity of motors for sagittal joints during the third squat test with 2 kg added on the waist.

Figure 8.50: Input voltages for sagittal joints during the third squat test with 2 kg added on the waist.
simpler controller to a more sophisticated multi-variable controller, which led to realizing that the problem lies in the offsets between the motor and link positions. Two sway tests are described in this section. The first test consist of a series of steps where homing is not done, but because the PID controller does not take any feedback from the link side the test works with no problems. The second test correspond to a tracking a sinusoidal wave, with frequency of 0.25 Hz. In this test, homing is done and the tracking performance, including the tracking errors, estimated velocities and control signals are presented.

8.6.1 PID Experimental Set-up

In both tests the robot is placed on the ground. The independent PID control which was discussed in section 4.1 is used in the tests. The PID gains are provided in Table 4.1. In the first test the feed-forward gain $G_{ff}$ is not used, which allowed us to better investigate the transient response. In the second test, once the homing is done, full feed-forward gain is used to investigate the tracking errors in response to a since wave. In both experiments the velocities are computed by differentiation and averaging over the past 10 samples. The sampling time in this experiment is 2 ms, which is consistent with the previous tests.

8.6.2 PID Experimental Results

The first set of plots presented correspond to the the step response of the robot. The lateral joints tracking is shown in Fig. 8.51 and the corresponding tracking errors are shown in Fig. 8.52. It is evident that when the the feed-forward gain $G_{ff}$ is not used the step response has a time constant of about 2.1 sec. What is more interesting in this experiment is the offset between the link and motor positions which does not lead to oscillations as opposed to the centralized LQR test. This hint was another evidence toward tackling the oscillation problem in the LQR implementation. Moreover, the motor velocities which are used by the PID controller are shown in Fig. 8.53. The velocities are computed by numerical differentiation in order to consistently control
the joints similar to the PID software in the DSP controllers. The corresponding voltages are shown in Fig. 8.54, which has a comparable magnitude to the LQR sway tests discussed in section 8.4. The ramps in the voltage signals correspond to the transient phase, when the integrators are summing up the error to decrease the error to zero. Moreover, on the support’s leg lateral ankle $q_1$ a high frequency oscillation can be seen in the plots which is due to a joint limit programmed in the software for safety considerations. The soft joint limit is -0.17 rad and although the value of reference signal at that point is -0.15 rad, due to the joint’s offset the joint position is recorded as -0.172 rad and the safety code is triggered which sets the voltage to zero to avoid any damage to the robot. Once the voltage is set to zero, gravity pulls the robot back into the safe region where the controller is activated and the same issue repeats until the reference is set to zero.

In the second test, a more dynamic set of references are chosen for the lateral sway. These references are sine waves with the frequency of 0.25 Hz. The homing is done in this test as evident in the joints tracking, shown in Fig. 8.55. The tracking errors are shown in Fig. 8.56 which are less than 10% of the reference signal’s magnitude. The velocities computed by differentiation and averaging are shown in Fig. 8.57. However, the PID controller only uses the motors’ velocities. Finally, the
Figure 8.52: Sway’s tracking errors of the PID sway’s step response.

Figure 8.53: Lateral motors velocities corresponding to the PID sway’s step response.
corresponding voltages are shown in Fig. 8.58. The peak to peak value of voltages for the ankles are slightly larger than the hip voltages which is mainly due to the ankle’s reference signals being slightly larger.

8.7 Tracking Walking Trajectories Using the PID Control

In this section, the experimental results of tracking a set of walking trajectories using the independent PID control scheme are presented. The main purpose of this experiment is to quantify the tracking performance and control efforts, when the walking trajectories, discussed in section 8.8, are applied to the robot.

8.7.1 PID Experimental Set-up

The independent PID control scheme discussed in section 4.1 is used with full feedforward gain ($G_{ff}$) to maximize the transient response without degrading the relative stability of the feedback loop. The velocities in this experiment were computed using numerical differentiation and averaging according to the details presented in section
Figure 8.55: Sine sway test by independent PID control.

Figure 8.56: Lateral tracking errors corresponding to the sine sway test.
Figure 8.57: Lateral motors velocities corresponding to the sine sway test.

Figure 8.58: Lateral input voltages corresponding to the sine sway test.
8.3.2. This setting is similar to the existing controllers programmed at the DSPs, and this tests reproduce the local controller results. The robot is held up in the air on the harness, while the walking trajectories are tracked by the control system. In all plots presented in this section the right hand side graphs refer to the right leg and the left hand side graphs refer to the left leg. Moreover, the order of the graphs from top to bottom in each figure is ankles, knees and hips for sagittal plane figures and ankles and hips for lateral plane figures.

8.7.2 PID Experimental Results

The tracking performance of sagittal and lateral joints is shown in Figs. 8.59 and 8.60. These figures confirm that the control system has the suitable bandwidth to track the walking trajectories. In order to better illustrate the tracking performance the tracking errors for the sagittal and lateral joints are shown in Figs. 8.61 and 8.62. In comparison with the LQR controller the tracking errors are about twice larger than the LQR tracking errors.

The velocities in this experiment are computed by first order differentiation on
Figure 8.60: Lateral joints tracking of the walking trajectory.

Figure 8.61: Sagittal joints tracking errors of the walking trajectory.
CHAPTER 8. EXPERIMENTAL RESULTS

Figure 8.62: Lateral joints tracking errors of the walking trajectory.

...
Figure 8.63: Sagittal joints’ velocity estimate of the walking trajectory.

Figure 8.64: Lateral joints’ velocity estimate of the walking trajectory.
Figure 8.65: Sagittal plane input voltages during the air walking experiment.

Once the robot is placed on the ground. Future research on stretched knee walking on CoMan can improve this problem during walking as has been confirmed by other research groups in Japan and Korea.

The voltages in lateral ankles are just above $\pm 5$ volts, while the voltages in the lateral hip joints are smaller. Since the robot is lifted in the air using the harness the effect of gravity is the potential reason for this observation, since the weight of the legs can assist the motion in the hip joints while the ankle joints require more torque to overcome the effects of stiction and viscose friction. Moreover, considering the fact that the peak to peak position trajectory and velocity of the ankle and hip lateral are similar, strengthens the idea of friction and stiction being the reason for larger voltages at the ankles.

### 8.8 Tracking Walking Trajectories Using the LQR Control

In this section, the experimental results of tracking a set of walking trajectories using the centralized LQR feedback are presented. The main purpose of this experiment
Figure 8.66: Lateral plane input voltages during the air walking experiment.

is to quantify the tracking performance and control efforts, when a more dynamic walking trajectory is applied to the robot. Maintaining balance of the legs during walking is not considered at this stage (the robot is held up on the harness), and the main focus is on validating the tracking system. The details of the experimental set-up and the results are discussed next.

### 8.8.1 LQR Experimental Set-up

The centralized feedback discussed in section 8.2 is used with full feed-forward gain \(G_{ff}\) to maximize the transient response without degrading the relative stability of the feedback loop. The velocities in this experiment were computed using numerical differentiation and averaging according to the details presented in section 8.3.2. The robot is held up in the air on the harness, while the walking trajectories are tracked by the control system. In all plots presented in this section the right hand side graphs refer to the right leg and the left hand side graphs refer to the left leg. Moreover, the order of the graphs from top to bottom in each figure is ankles, knees and hips for sagittal plane figures and ankles and hips for lateral plane figures.
8.8.2 LQR Experimental Results

The tracking performance of sagittal and lateral joints using the centralized LQR feedback is shown in Figs. 8.67 and 8.68. The control system has the suitable bandwidth to track the walking trajectories. An interesting feature evident in Fig. 8.67 is the offset which corresponds to bent knee walking. This offset for the ankles (joints 2 and 9) and the hips (joints 5 and 6) in the sagittal plane is 0.3 $\text{rad}$ while the offset for the knees (joints 3 and 8) is twice this value. This is due to the condition of keeping the feet parallel to the ground. The lateral joints shown in Fig. 8.68 do not have this offset and are centred about zero. The LQR tracking errors are shown in Figs. 8.69 and 8.70. In section 8.7, it is shown that PID tracking errors are about twice the value for the LQR tracking.

The resulting estimated velocities using numerical differentiation and averaging over the last 10 samples are shown in Figs. 8.71 and 8.72.

The most interesting point about this experiment is the magnitude of the control signals. The LQR sagittal and lateral control voltages are shown in Figs. 8.73 and 8.74. The peak to peak values of voltages in the sagittal plane are about 10 volts for the ankles and the knees, and 8 volts for the hips. The peak to peak values in the
Figure 8.68: Lateral joints tracking of the walking trajectory using LQR.

Figure 8.69: Sagittal joints tracking errors of the walking trajectory using LQR.
Figure 8.70: Lateral joints tracking errors of the walking trajectory using LQR.

Figure 8.71: Sagittal joints’ velocity estimate of the walking trajectory using LQR.
lateral plane are about 5 volts. In comparison with the same tracking experiment described in section 8.7, LQR controller requires 50% less control voltages and also the LQR tracking errors are about half the PID tracking errors. This is explained further in section 8.7 where the peak to peak value of the control voltages in sagittal and lateral planes are 20 volts and 10 volts, respectively. Considering the fact that the robot is held up on the harness the magnitude of control signals are expected to get larger once the robot is walking on the ground. Hence, LQR is shown to have better performance in practice, both in terms of having higher bandwidths and also producing less control signals which is important to avoid saturating the actuators.

The experimental results presented in this chapter were only achieved once several challenges were tackled. These challenges are described in the next section.

### 8.9 Technical Challenges

In this section, a set of technical challenges which were faced during this PhD project are briefly summarized. At the early stages of this project, a feasibility study about walking was done on iCub. However, several hardware limitations were faced which are explained in sections 8.9.1 and 8.9.2. These hardware limitations are related
Figure 8.73: Sagittal plane input voltages during the air walking experiment using LQR.

Figure 8.74: Lateral plane input voltages during the air walking experiment using LQR.
to low communication bandwidth and lack of robustness of the iCub’s ankles. As 
mentioned in chapter 2, the main goal of the iCub’s project was to provide a robotic 
platform to study how an infant learns the basic skills such as crawling. Because 
of the size of the actuators, the final design was realized with size of a 2-3 year old 
child. Therefore, initially iCub was not designed for walking and only later during 
the project, developing a walking system for iCub was considered.

Having identified these technical issues, a compliant humanoid robot called Co-
Man was developed later in the project (2010), which evolved from the initial iCub’s 
mechatronic design to a more robust design to carry out walking on uneven terrain as 
well as more dynamic motions such as jumping and running. CoMan is equipped with 
Ethernet communication to remove the previous communication bottle neck as well 
as more powerful and robust mechanical design for the ankles. However, a different 
set of software related challenges were faced and mostly tackled which are discussed 
in sections 8.9.3, 8.9.4 and 8.9.5. These software challenges were identified as a re-

8.9.1 Communication Bandwidth

The version of iCub which was used in 2009 for initial identification and LQR feedback 
implementation had CAN communication which has a maximum bandwidth of 1 
Mbit/sec. Considering the number of joints (at least 12 joints including 10 DoF for 
both legs and 2 DoF for the torso) that have to be controlled by the centralized LQR 
and number of measurements and control commands per joint, it became clear that 
CAN communication had a bottle neck which creates serious problems for centralized 
feedback control of humanoid robots with high DoF systems.
In the initial experiments on iCub sampling time interval proved to be non-uniform with sampling time of 10 ms as shown in Fig. 8.75. This variation is spotted by looking at the differences in the time stamp of each data processed at the controller. Moreover, time varying delays were imposed by the LAN network which connected the robot’s hardware to the PC which was used to run the LQR feedback control. For these reasons the attempts toward identification of the robot’s dynamic parameters could not be completed since the identification methods assume a constant sampling time interval and if this assumption is violated the identification data does not provide a useful outcome. However, this issue was identified and resolved in CoMan that is designed to perform more dynamic tasks such as running and jumping. Fig. 8.76 shows a typical sampling time interval in an experiment on CoMan which has proved to be very regular at 2 ms. However, a set of identification tests is yet to be performed on the robot once the robot’s hardware reaches the final development stages, with the torque sensors properly installed.

8.9.2 Cable Drives on the Ankles

iCub uses steel cables to drive the ankles and hips in the sagittal plane. These cables have an estimated theoretical stiffness of $1.2 \times 10^5 \, N/m$ for the nominal length of 0.2 m.
Figure 8.76: A plot of the regular sample times in an experiment on CoMan with sample time of 2 ms.

The cables have a structural stretch of 1% of their nominal length due to their multi core and laminated construction which can be reduced by pre-tensioning the cable. However, placing the iCub on the ground on its feet, occasionally caused the cables to slip due to the cable stretch changes as a result of the robot’s body weight. This issue is not critical for iCub since its main purpose is to apply learning algorithms which involves the robot’s upper body and hands while the body is supported on a stable frame. Nevertheless, initial tests on iCub provided an insight into designing a more robust ankle for CoMan which has replaced the cables with a lever mechanism to connect the gearbox output to the ankle joint as shown in Fig. 8.77. The next three set of challenges which were faced in the experiments can be characterised by low and high frequency oscillations, when the robot had to regulate the joints to zero reference position. Two of these challenges (as discussed in sections 8.9.3 and 8.9.4) were tackled once various possible sources of oscillation were ruled out. The cause for the remaining high frequency oscillation issue (as discussed in section 8.9.5) was identified much later in the project (during the writing of this thesis) to be due to a bug in the DSP code, which is responsible for generating the PWM and the commutation of the motors.
Figure 8.77: Robust mechanical design of CoMan’s ankle joint built at the department of advanced robotics at IIT.

8.9.3 Homing and Calibration of Multiple Sensors

An important technical challenge on implementation of the centralized multivariable LQR and observers was to uncover the source of unknown oscillations which was observed, when regulating the joints’ positions to zero. Numerous tests were carried out on CoMan and the experimental data was logged. All the plots presented in this section have 2 ms sampling time. Plots of one of the experiments are presented in this section to illustrate the problem. Figs. 8.78 and 8.79 show the oscillatory position and velocity responses as observed in the experiments. The data corresponds to the support leg’s knee joint while the robot is placed on the ground, while a similar behaviour is observed on the other knee joint and the ankle joints. After closely examining the transient phase in the logged data, a potential cause for this behaviour was suspected to be inconsistent sensory information.

This discrepancy between the initial motor and the link position, is more clear in a close-up view of Fig. 8.78, as shown in Fig. 8.80. This effect was present in all the previous experimental logs. The offsets between the sensors’ readings, for 10 DoF, in this case is

\[ [0.019, 0.116, -0.011, 0.004, 0.048, -0.061, -0.014, -0.020, -0.126, 0.002]. \tag{8.6} \]

This is confirmed by a nonlinear simulation of CoMan with offsets in Equation
Figure 8.78: Oscillations observed in an experiments on CoMan on the ground.

Figure 8.79: Estimated joint’s and motor’s velocity by numerical differentiation.
Figure 8.80: Initial transient leading to oscillations in the experiments on CoMan on the ground (close-up view of Fig. 8.78).

(8.6) added to the motor positions to resemble the calibration error as shown in Fig. 8.81. Once the robot is on the ground the first transient response is so large (in excess of $0.8 \text{ rad}$) that disturbs the static balance of the legs on the ground and the robot falls. The largest oscillations are observed on the ankles (which also have the largest offsets) as shown with dotted and dashed lines in Fig. 8.81. Hence, this discrepancy, which is a calibration issue for multiple sensors, was indeed the source of the undesired oscillations on the ground.

Once the source of this problem was identified, a feedback technique was used to drive the joints to the zero positions and then the offset between the motor positions and the link positions was removed. This technique is referred to as the homing, which is a crucial step before implementing any multivariable controller with feedback from multiple sensors. Fig. 8.82 shows the sagittal angles being controlled on the ground using the observer based LQR. The oscillation problem is solved and the controller can respond to the external disturbances applied to the legs (several pushes by hand). In the initial 8 sec the homing is applied and the offsets are removed, as can be seen from the Fig. 8.82. In majority of the tests, the largest offsets belong to the ankle sagittal joints.
Figure 8.81: Nonlinear simulation of CoMan with an offset between the motors and links positions predicts large oscillations on the ground. The dotted, dash-dotted and dashed lines represent the hip, knee and ankle of the support leg in sagittal plane.

Figure 8.82: Once homing is done, the unstable transient diminishes and the robot responds to external disturbances.
Tackling this source of oscillations on the ground, paved the way towards implementing the centralized LQR and observers on CoMan. However, oscillatory behaviour of a different nature was also observed as explained in the next section.

8.9.4 Limit Cycles due to Stiction and Quantization

A different type of low frequency and low amplitude oscillations was observed in experiments when robot was lifted in the air with a harness and integrators were closed on the links. Homing was not done in this experiment, because this issue was observed while the homing problem was being investigated. The effects of quantization and stiction in the simulation was suspected to be a potential cause for the oscillations, because the robot was held up and the large transient explained in section 8.9.3 appears once the robot is placed on the ground.

Fig. 8.83 shows the typical oscillatory behaviour that was observed, when the references were kept at zero and the robot was kept in the air with a harness. The corresponding velocity estimates for this joint are shown in Fig. 8.84 which shows large velocities and it was initially suspected that this could be the reason for the oscillations. The frequency of oscillation can be derived from the graphs to be about 0.4 Hz, which is a low frequency. Therefore, a “slow” Luenberger observer, a “fast” Luenberger observer (in terms of their eigenvalues) and an unknown input observer [191] (which did not use the input voltage to avoid stiction effects) were tested to tackle the oscillations issue which are explained in detail in sections 8.3.2 and 8.3.3. However, the problem was still persistent. The stable limit cycle oscillations persist once the robot was placed on the ground as can be seen in a video clip in Appendix F. Although the velocity estimate has improved in this experiment the oscillations were still present. Therefore, the possibility of the source of oscillations being the large velocity estimates was ruled out. This issue was found to be due to a combination of high integrators’ gains (using the penalty of 10 instead of 1 in the LQR feedback design), stiction and quantization noise. This problem was resolved by reducing the integrators’ gains (using the LQR penalties, as discussed in section 8.2). Therefore, in
the experiments presented in this chapter the integrators’ gains were lowered down to about 1.4 by adjusting their corresponding penalties in the LQR feedback design.

### 8.9.5 High Frequency Oscillatory Behaviour

A further challenge was faced once the homing issue was resolved. Undamped high frequency oscillations occurred in several tests as shown in Figs. 8.85 and 8.86 for the ankle and knee joints for instance. Similar behaviour was observed on the other joints. In this experiment the integrators were closed on the links’ positions. The
Figure 8.85: Experimental data from ankle sagittal with homing and integrators closed on the ankle encoder.

The frequency of oscillations at the ankle joint is 17 Hz while the frequency of oscillations at the knee is 10 Hz.

A similar effect was observed in another test when the controller designed in section 8.2 was applied to the robot and the integrators were closed on the motors. Initially, (until t=40 sec) the oscillations are more evident on the knee joint, while at t=40 sec, the robot is placed on the ground and the oscillations propagate to all the joints with higher amplitude as shown in Fig. 8.87. The corresponding voltages are shown in Fig. 8.88.

Initially, several possible causes for this problem were suspected. The first possibility was the lower stiffness in the actual hardware in comparison to the nominal values of the spring’s stiffness provided by the manufacturer. Much later in the project when the torque sensors were installed on the legs, the stiffness value was measured to be about 110 Nm/rad, in comparison with the nominal value of 190 Nm/rad. However, simulations do not suggest that the high frequency oscillations are due to the lower stiffness. Recently (during the writing of this thesis), the second cause was identified which is related to a bug in the DSP code. This code is responsible for generating the PWM signal and the commutation for the motors. The result of this bug is that, once the joint is disturbed against the motion the motor responds
Figure 8.86: Experimental data from knee joint with homing and integrators closed on the knee encoder.

Figure 8.87: Experimental data from sagittal joints’ positions (with homing and integrators closed on the motors’ encoders).
Figure 8.88: Sagittal joints’ voltages corresponding (with homing and integrators closed on the motors’ encoders).

correctly and pushes back, while if the disturbance is in the direction of motion the PWM is not generated correctly and the joints starts to oscillate. Therefore, the oscillations on the knee joint in Fig. 8.86, while the robot is held up by the harness can be explained by the effects of gravity which can push against the control effort or in the direction of it. Moreover, in the other figures, once the robot is placed on the ground, the effects of gravity become much more pronounced and due to couplings between the joints the oscillations propagate. However, resolving this issue is left for future work as the robot is still in the debugging process.

8.10 Conclusions

In conclusion, the centralized observer based LQR was implemented on a compliant joint unit. Several interesting effects were observed in the experiment. These effects consist of larger control voltages and larger velocity estimates (in comparison with the simulation prediction) which were shown to be mainly due to stiction and viscose friction in the prototype of CoMan’s joint.
Furthermore, the centralized observer based (slow observer) LQR was implemented on 10 DoF of CoMan. The theoretical bandwidth prediction from the simulations was about $7.1-8.8 \text{ rad/sec}$ (1.2-1.4 Hz), which can be increased further once the debugging process on the legs has finished and the hardware has reached its final specification. At this stage of the work, this bandwidth is chosen to ensure the safety of users as well as the least amount of damage to the robot in case of a failure in an experiment. This bandwidth is sufficient to achieve the walking speed of $0.5 \text{ m/sec}$, which is quite a high speed when the kinematic relations of CoMan are considered ($0.5 \text{ m/sec}$ is equivalent of the relative speed of a leg length per second).

Moreover, several velocity estimation schemes were tested on 10 DoF of CoMan’s legs. These schemes consist of a slow Luenberger observer, a fast Luenberger observer, an unknown input observer and the traditional numerical differentiation with averaging. It was shown that although the unknown input observer has the best relative stability margins, the closed loop system is unstable. This is mainly due to the fact that the unknown input observer behaves as a differentiator which amplifies the quantization noise and creates undesired oscillations. The same effect was seen on the fast Luenberger observer which also tends to act as a differentiator. The stable results were achieved using the slow observer and numerical differentiation with averaging over the last 10 samples. As a potential future work, implementing a Kalman filter will be considered to reduce the quantization noise for velocity estimation.

As part of providing standard and repeatable tests, several sway tests were carried out as the simplest tests to validate the overall control system’s performance and robustness. The experimental results of a sway test were presented to quantify the tracking performance as well as the control effort.

In order to quantify the energy and torques at the ankles and knees two slow (almost static) and fast squat motion tests were carried out where the torques at the knee and ankles are quantified. The CoM of the legs are kept within the support polygon during both tests. An interesting feature of the sway and squat tests is that the lateral (joints with high stiffness) and sagittal (joints with low stiffness) planes can be tested separately.
Finally, in order to validate the tracking performance using a more dynamic walking trajectory, two experiments were carried out using the PID and LQR control, while the robot was being held on the harness and the joint tracking data was logged. The tracking errors and control effort were shown to be 50% smaller when LQR feedback is used. This advantage is significant in fast walking since the LQR will have a larger margin before saturating the actuators. This advantage can be further investigated during future walking experiments. Both PID and LQR had small errors and the control signals and future work can focus on implementing various walking trajectories using the developed centralized control system. Therefore, as result of the experiments carried out in this PhD project, the infrastructure for centralized controller implementation was developed to pave the way toward implementing high speed trajectory tracking which is required for dynamic motions such as fast compliant walking.
Chapter 9

Conclusions and Future Work

This PhD thesis reports on the modelling and dynamic stabilisation of the novel compliant humanoid robot, CoMan, for the purpose of compliant walking. The main aim of CoMan is to utilize the natural dynamics, energy efficiency and shock tolerance benefits of passive springs in dynamic walking and subsequently in explosive tasks such as jumping and running.

Bipedal walking is inherently hybrid due to the periodic sequences of continuous motions and discrete transitions, which poses significant challenges for controller design. Control of walking becomes more challenging when passive compliance is embedded into the robot joints.

In this thesis, the stabilisation problem of a compliant humanoid robot during walking, which has unstable dynamics resembling an inverted pendulum, was studied in detail for both the single support and the double support phases of walking. In order to achieve this aim, detailed dynamic models of CoMan were developed using Robotran and Matlab, which are available for download [1]. These models were utilised to design model based centralized and decentralized LQR tracking systems with a mathematical proof of closed loop stability. The purpose of this study was to provide an improvement on the existing independent PID controllers, which did not cater for the compliance in the joints. The existing PID controllers on CoMan have been used to achieve slow (about 0.1 \( m/sec \)) and simple walking gaits. The centralized LQR feedback is shown to produce 50% less control effort and tracking
errors while tracking the same walking trajectories. Although walking on the ground is not achieved using the LQR controller at this stage, the results presented in this thesis are promising that higher bandwidths with less control efforts and tracking errors can be achieved using the LQR feedback control. Also, the existing PIDs control the motor positions, while the LQR feedback can be adjusted to control the motor or more importantly the link position to reduce the oscillations.

Two decentralized feedback design algorithms, based on the LQR formulation, were proposed in this thesis, with the following motivations. Firstly, to achieve faster sampling rates (more than 1 kHz) and to be able to process information locally at the DSP level. Secondly, to derive model based algorithms for tuning the PD-PID gains of each joint to replace the conventional trial and error approach to this problem in humanoid robots. The first algorithm is based on a sparse gradient descent feedback design, that takes the sparsified LQR feedback gain as an initial value and then uses a sparse gradient descent algorithm to iteratively minimize the original LQR cost. The effectiveness of this algorithm is shown for a rigid and compliant models of CoMan with 10 DoF. The main drawback of this method is that if the initial sparsified LQR gain is not stabilizing the iterations can not be started. This problem occurs when the feedback bandwidth is increased. Thus, a second algorithm based on LMI formulation is proposed to minimize the LQR cost function. The effectiveness of this algorithm is shown for rigid and compliant models of CoMan with 10 DoF. The main drawback of this algorithm is that in some cases the LMI formulation is not feasible. It is pointed out that the proposed LMI-LQR algorithm is an approximation of the original BMI problem (The proposed LMI formulation uses sufficient conditions to approximate the original non-convex optimisation by a convex one). Effective commercial solvers (TOMLAB/PENBMI) are identified to directly solve the BMI-LQR problem, which seems to be an interesting area to explore.

In summary, the centralized LQR tracking system was shown to have the advantage of achieving higher tracking bandwidths, while the decentralized LQR showed a degradation in the tracking bandwidth. In addition, the decentralized LQR results in a further degradation once the formulation is converted from continuous time to
discrete time. However, an important feature of both the centralized LQR and the decentralized LQR designs is accounting for the links’ interactions, which is often neglected in feedback control design for robotics. The idea of including the links’ interactions in the feedback design stems from the fact that given the knowledge of the multibody dynamics, one can improve the feedback design by including this additional information rather than treating the interactions as undesired disturbances.

In addition, the challenging problem of controlling CoMan in double support was studied, where a novel control method based on geometric control theory was proposed. The continuous time single support model was used for the feedback design which has full controllability and a full mathematical proof of the closed loop stability was derived. The effectiveness of this algorithm is shown in simulations, while implementation on CoMan will be considered in future work, once the algorithm is properly discretized.

Moreover, a dynamic walking simulator for CoMan which includes the dynamical parameters (body length, mass, CoM, inertia), actuator and compliance dynamics, ground contact model and sensor noise, was developed in Robotran and Matlab. The parameters of CoMan were extracted from the CAD software. The derived linearized models in Robotran and Matlab, were then used for centralized and decentralized controller designs and simulations. The effectiveness of this simulator was shown by simulating a walking trajectory using the dynamic model. Simple models of walking (LIP models) were used to quantify the upper bound on the maximum feasible walking speed, step size and a solutions such use of toe-off and straight knee were suggested to improve the walking parameters. In addition to kinematic constraints, the roles of bandwidth and dynamic walking stability measures such as ZMP on maximum walking speed were studied. Numerical results were provided to provide the required bandwidth given a desired walking speed. Moreover, the role of upper body in improving walking robustness was shown using an extended cart-table model of walking and an online trajectory generation method based on preview control was formulated.

A feasibility study about walking with iCub, involving numerous experiments, was
carried out during the second year of this PhD. Several issues were identified which are pointed out in section 8.9 and in the IET presentation [50]. Having learnt the limitations of iCub for walking, CoMan was developed which addressed those issues. Hence, during the final year of this PhD project, as CoMan was being developed (since 2010), several experiments were carried out to implement and validate the centralized LQR tracking control system for locomotion. Initially, a single compliant joint unit, was used to implement the observer based LQR, where the effectiveness of this control method was validated. Interesting effects were observed (larger control voltages and larger velocity estimates as opposed to the simulation predictions) which were shown to be mainly due to stiction and viscose friction. Several practical challenges were faced and mostly tackled as discussed in section 8.9. Part of these challenges were due programming work for the controller development and logging the data.

The bandwidth of the LQR tracking system was increased gradually in several experiments up to the theoretical value of 1.4 Hz, as discussed in section 8.2. The tracking bandwidth can be increased further but due to safety considerations the current design has the conservative bandwidth of 1.4 Hz. Subsequently, the results of implementing sway and squat motions were presented. These results provided an indication on the overall tracking performance as well as quantifying the control effort (joint torques) in the ankle and knee joints. Furthermore, a more dynamic walking reference tracking was also used in another experiment to quantify the errors and control signals. It was shown that both measures have good accuracy considering the conservative bandwidth of the control system.

Improving the velocity estimation was another area of study in this project. Several schemes based on a fast and a slow Luenberger observer, an unknown input observer and traditional numerical differentiation were proposed and compared in simulations and experimentally. The slow observer and numerical differentiation with averaging over the last 10 samples provided the most stable experimental results. The slow observer estimation is less noisy compared to the numerical differentiation, but an offset is present in the observer’s data which is shown to be a result of stiction and viscose friction. The fast observer resulted in an oscillatory behaviour which was
mainly due to amplifying the quantization noise. The unknown input observer provided the largest relative stability for the control system, but in practice it amplified the measurement noise and did not provide a promising outcome. Therefore, implementing a Kalman filter can potentially reduce this effect and improve the velocity estimation.

The work presented in this PhD thesis are published or submitted to well-known national and international journals and conferences. Two journal papers are accepted for publication (with the details provided in section 1.5.1) and five conference papers are also submitted with four papers already published (with the details provided in section 1.5.2).

9.1 Future Work

There are a number of areas that will be considered for future work. The dynamic simulator built in Robotran and Maple will be extended to a more user friendly environment to allow study of walking with CoMan while having access the powerful tools in Matlab. As an alternative to Robotran, the simulation environment in MapleSim will be developed further which has shown promising results in early stage evaluation of this software.

In terms of decentralized feedback design the designs presented in this thesis will be implemented on CoMan to improve the controllers for walking. The decentralized feedback design using the BMI approach will be investigated. A commercial BMI solver (TOMLAB/PENBMI) is identified which can be used to explore this area. Currently, the robot is being equipped with torque sensors to allow improved force control. The torque sensors can be incorporated in the proposed control system to add the capability of force or impedance control as well as the existing position control system. Regarding the velocity estimation, potential areas such as Kalman filtering will be explored to further improve the results. The double support controller will be discretized for implementation on CoMan to validate the proposed constrained feedback algorithm experimentally.
Further walking experiments will be carried out on CoMan to better illustrate the effectiveness of the proposed centralized LQR feedback in terms of increased walking speed and robustness (less possibility of saturating the actuators). This is tightly related to trajectory generation schemes. Thus, the proposed ZMP online trajectory generation with the upper body stabilization, will be considered initially for generating the walking trajectories. Subsequently, the toe off phase will be added to achieve faster and more robust walking.

Another line of research is to study the effect of the passive compliance on walking trajectory generation, where optimisation methods can be used to identify the stiffness values for the joints. This work leads to the idea of using actuators with adjustable stiffness capable of increasing the stiffness during fast dynamic motions while reducing the stiffness when appropriate.
Appendix A

Planar Double Pendulum

The main purpose of this appendix is to derive the equations of motion for a planar double pendulum by hand and compare the results with Robotran to verify the derived dynamic models using this software. It should be noted that a double pendulum is presented as a sample, but further tests have been carried out during this PhD project to verify models with more DoFs (four and six DoF) and also the kinematic information provided by the Robotran software.

The double pendulum consist of two masses attached to two rigid, massless, rods as shown in Fig.A.1. The double pendulum pivots do not have friction. The parameters of the double pendulum system consist of two lengths of the rods, $l_1$ and $l_2$, two masses $m_1$ and $m_2$, two absolute angles between the vertical and the rods, $\theta_1$ and $\theta_2$, and gravity, $g$. Lagrange method is used to derive the double pendulum equations of motion.

A.1 Lagrangian Equation

The general Lagrange equation of motion is

$$\frac{d}{dt} \left( \frac{\partial L}{\partial \dot{\theta}} \right) - \frac{\partial L}{\partial \theta} = \Gamma$$

(A.1)
where, $L$ is the Lagrangian, which is defined as the difference between the kinetic energy $K$ and the potential energy $P$ as

$$L = K - P$$  \hspace{1cm} (A.2)$$

and $\Gamma$ is the input torque, but for simplicity $\Gamma$ is set to zero in the following derivations. In the following sections the kinetic and potential energy of the double pendulum are derived.

### A.1.1 Kinetic Energy

The kinetic energy for a moving mass is

$$K = \frac{mv^2}{2}.$$  \hspace{1cm} (A.3)$$

In addition, from Fig. A.1 the following cartesian relationships can be derived.

$$x_1 = l_1 \sin \theta_1$$  \hspace{1cm} (A.4)$$

$$x_2 = l_2 \sin \theta_2 + l_1 \sin \theta_1$$  \hspace{1cm} (A.5)$$

$$y_2 = l_2 \cos \theta_2 + l_1 \cos \theta_1$$  \hspace{1cm} (A.6)$$
The $v^2$ can be expressed in terms of $x$ and $y$ as $v^2 = \dot{x}^2 + \dot{y}^2$ and squared derivatives of (A.3) to (A.6) are:

\[
x_1^2 = l_1^2 \dot{\theta}_1^2 \cos^2 \theta_1
\]
\[
y_1^2 = l_1^2 \dot{\theta}_1^2 \sin^2 \theta_1
\]
\[
x_2^2 = l_2^2 \dot{\theta}_2^2 \cos^2 \theta_2 + 2l_1l_2 \dot{\theta}_1 \dot{\theta}_2 \cos \theta_1 \cos \theta_2 + l_1^2 \dot{\theta}_1^2 \cos^2 \theta_1
\]
\[
y_2^2 = l_2^2 \dot{\theta}_2^2 \sin^2 \theta_2 + 2l_1l_2 \dot{\theta}_1 \dot{\theta}_2 \sin \theta_1 \sin \theta_2 + l_1^2 \dot{\theta}_1^2 \sin^2 \theta_1
\]

Hence the kinetic energy relations $K_1$ and $K_2$ for the first and second link of the pendulum are:

\[
K_1 = \frac{m_1}{2} l_1^2 \dot{\theta}_1^2
\]  
(A.7)

\[
K_2 = \frac{m_2}{2} (l_2^2 \dot{\theta}_2^2 + l_1^2 \dot{\theta}_1^2 + 2l_1l_2 \dot{\theta}_1 \dot{\theta}_2 \cos (\theta_1 - \theta_2))
\]  
(A.8)

### A.1.2 Potential Energy

The potential energy equations is the equation for all potential energy of an equation.

\[ P = mgh \]

where, $m$ is mass, $g$ is gravity and $h$ is the height. The potential energy relations for the first and second pendulums are given in (A.9) and (A.10).

\[
P_1 = m_1 gy_1 = m_1 gl_1 \cos \theta_1
\]  
(A.9)

\[
P_2 = m_2 gy_2 = m_2 g (l_2 \cos \theta_2 + l_1 \cos \theta_1)
\]  
(A.10)

According to (A.3), (A.7), (A.8), (A.9), and (A.10), the Lagrangian is $L = (K_1 + K_2) - (P_1 + P_2)$, that is

\[
L = \frac{m_1}{2} l_1^2 \dot{\theta}_1^2 + \frac{m_2}{2} (l_2^2 \dot{\theta}_2^2 + l_1^2 \dot{\theta}_1^2 + 2l_1l_2 \dot{\theta}_1 \dot{\theta}_2 \cos (\theta_1 - \theta_2)) - m_1 gl_1 \cos \theta_1 - m_2 gl_2 \cos \theta_2 - m_2 gl_1 \cos \theta_1.
\]  
(A.11)

Hence the Lagrange equations of motion can be expressed as:

\[
\frac{d}{dt} \left( \frac{\partial L}{\partial \dot{\theta}_1} \right) - \frac{\partial L}{\partial \theta_1} = 0
\]  
(A.12)
where,

\[
\frac{\partial L}{\partial \theta_1} = -m_2 l_2 \dot{\theta}_1 \dot{\theta}_2 \sin(\theta_1 - \theta_2) + m_1 g l_1 \sin \theta_1 + m_2 g l_1 \sin \theta_1
\]

and,

\[
\frac{\partial L}{\partial \dot{\theta}_1} = m_1 l_1^2 \ddot{\theta}_1 + m_2 l_1^2 \ddot{\theta}_1 + m_2 l_1 l_2 \ddot{\theta}_2 \cos(\theta_1 - \theta_2)
\]

\[
\frac{d}{dt} \left( \frac{\partial L}{\partial \dot{\theta}_1} \right) = m_1 l_1^2 \ddot{\theta}_1 + m_2 l_1^2 \ddot{\theta}_1 + m_2 l_1 l_2 \ddot{\theta}_2 \cos(\theta_1 - \theta_2) - (\dot{\theta}_1 - \dot{\theta}_2) m_2 l_1 l_2 \ddot{\theta}_2 \sin(\theta_1 - \theta_2)
\]

Similarly,

\[
\frac{d}{dt} \left( \frac{\partial L}{\partial \dot{\theta}_2} \right) - \frac{\partial L}{\partial \theta_2} = 0
\]

where,

\[
\frac{\partial L}{\partial \theta_2} = m_2 l_2 \dot{\theta}_1 \dot{\theta}_2 \sin(\theta_1 - \theta_2) + m_2 g l_2 \sin \theta_2
\]

and,

\[
\frac{\partial L}{\partial \dot{\theta}_2} = m_2 l_2^2 \ddot{\theta}_2 + m_2 l_1 l_2 \ddot{\theta}_1 \cos(\theta_1 - \theta_2)
\]

\[
\frac{d}{dt} \left( \frac{\partial L}{\partial \dot{\theta}_2} \right) = m_2 l_2^2 \ddot{\theta}_2 + m_2 l_1 l_2 \ddot{\theta}_1 \cos(\theta_1 - \theta_2) - (\dot{\theta}_1 - \dot{\theta}_2) m_2 l_1 l_2 \ddot{\theta}_1 \sin(\theta_1 - \theta_2).
\]

Equations (A.12) and (A.13) are compared against Robotran software, and after converting the joint angle convention form relative to absolute (Robotran generates the model in relative coordinates), the correctness of the equations derived by Robotran was ensured.
Appendix B

State Space Models

The detailed rigid and compliant dynamic model of CoMan is provided. These state space matrices are meant to assist in reproducing the results of this thesis.

B.1 The Rigid Model of CoMan

The state space matrices for the rigid models (without actuator dynamics) are

\[
\tilde{A} = \begin{bmatrix}
0 & I \\
-M^{-1}(G) & -M^{-1}(C)
\end{bmatrix}, \quad \tilde{B} = \begin{bmatrix}
0 \\
M^{-1}
\end{bmatrix}.
\]

B.2 The Compliant Model of CoMan

The state space matrices presented in section 3.5.2 are

\[
\tilde{A} = \begin{bmatrix}
0 & I & 0 & 0 \\
-M^{-1}(G + K_s) & -M^{-1}(C + B_s) & M^{-1}K_s & M^{-1}B_s \\
0 & 0 & 0 & I \\
J^{-1}K_s & J^{-1}B_s & -J^{-1}K_s & -J^{-1}(B_m + B_s)
\end{bmatrix},
\]

\[
\tilde{B} = \begin{bmatrix}
0 \\
0 \\
0 \\
J^{-1}K_iR^{-1}
\end{bmatrix}.
\]
Appendix C

Dynamic Parameters of CoMan

The parameters presented in this appendix are used to derive the linear state space model of CoMan with 10 DoF in Chapter 8.

C.1 Mechanical Parameters

The linear simulation presented in this paper utilizes the following parameters of CoMan to provide a realistic simulation. Note that gravity vector is defined to be in negative Z direction with the vector value of \( \mathbf{g} = (0, 0, -9.81) \). Figure C.1 illustrates diagram of the articulated multi-body model of CoMan. In this diagram, arrows represent distances in XYZ cartesian coordinates, rectangles represents bodies such as lower leg, upper leg and so on, squares represent bodies for the actuation units to denote their mass and circles represent revolute joints where \( R1 \) denotes revolution about the \( X-axis \) and \( R2 \) denotes revolution about the \( Y-axis \). The mechanical parameters of the model illustrated by figure C.1 are provided in Table C.1. There are a few points that must be considered about the values presented in this table. The vector valued CoM (Centre of Mass) for bodies \( B_1, ..., B_9 \) are presented with respect to the body’s local coordinate frame via vectors \( \mathbf{c}_1, ..., \mathbf{c}_9 \) as shown in figure C.1. However, due to the orientation of the bodies \( B_5, B_6 \& B_7 \) their CoM values are derived as \( \mathbf{c}_5 = \mathbf{d}_3 - \mathbf{c}_3 \), \( \mathbf{c}_6 = \mathbf{d}_2 - \mathbf{c}_2 \), \( \mathbf{c}_7 = \mathbf{d}_1 - \mathbf{c}_1 \). Due to symmetry between the right and left leg the distances (lengths) \( \mathbf{d}_1, \mathbf{d}_2, \mathbf{d}_3 \) are the same as \( \mathbf{d}_5, \mathbf{d}_6, \mathbf{d}_7 \).
Figure C.1: This diagram illustrate the multi-body model and its various parameters used for CoMan model where bodies are denoted by $B_1, ..., B_9$, lengths (distances) are denoted by $d_0, ..., d_9$ and centers of mass are denoted by $c_1, ..., c_9$.

respectively and if presented as vectors, their sign can be derived from the direction of the $d_i$ arrows in the figure.

**Compliance:** In the current version of CoMan all sagittal plane revolute joints $R2$ have passive compliance. This is illustrated by a different color in the figure C.1. The stiffness value for ankle sagittal and knee joints of each leg is 190 ($N.m/rad$) and the stiffness value for hip sagittal (that is due to a cable drive) is 460.2356 ($N.m/rad$).

The value for stiffness of the cable drive in the hip joint is already reflected to the joint side. There is a secondary reduction stage for the hip sagittal joint (implemented by cable and pulley system) and the ankle sagittal joint (using a lever system). Both have a reduction factor of 1.5. Furthermore, the passive springs are mounted at the joints for the sagittal knee and ankle as shown in Fig. 8.77.
Table C.1: Mechanical parameters of CoMan.

<table>
<thead>
<tr>
<th>Body</th>
<th>Mass (kg)</th>
<th>CoM (m)</th>
<th>Length (m)</th>
<th>Inertia (kg.m²)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(B1 &amp; B7¹)</td>
<td>1.5721</td>
<td>(−6.3401e−4)</td>
<td>0</td>
<td>6.383e−3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(−7.05e−3)</td>
<td>0</td>
<td>5.8657e−3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(1.15618e−1)</td>
<td>0.2013</td>
<td>4.5611e−4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(−1.9936e−3)</td>
<td>0</td>
<td>2.8048e−6</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(1.084e−3)</td>
<td>0</td>
<td>2.0037e−2</td>
</tr>
<tr>
<td>(B2 &amp; B6¹)</td>
<td>1.8696</td>
<td>(−9.048e−2)</td>
<td>0.22663</td>
<td>3.9096e−4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(1.6274e−3)</td>
<td>0</td>
<td>2.0658e−4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(7.022e−3)</td>
<td>0.038</td>
<td>4.9053e−6</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(8.4371e−5)</td>
<td>0</td>
<td>8.713e−8</td>
</tr>
<tr>
<td>(B3 &amp; B5¹)</td>
<td>0.90235</td>
<td>(−1.3444e−2)</td>
<td>0</td>
<td>3.4351e−2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(4.4035e−2)</td>
<td>0.0882</td>
<td>1.8625e−2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(1.0791e−1)</td>
<td>0</td>
<td>1.3519e−6</td>
</tr>
<tr>
<td>B4</td>
<td>5.5444</td>
<td>(9.8282e−3)</td>
<td>0</td>
<td>6.613e−4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(1.0230251e−2)</td>
<td>0</td>
<td>5.9292e−4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(2.1197e−4)</td>
<td>0</td>
<td>1.0634e−5</td>
</tr>
<tr>
<td>B8</td>
<td>0.898</td>
<td>(2.4261541e−2)</td>
<td>0</td>
<td>5.0798e−4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(−1.7244e−4)</td>
<td>0</td>
<td>1.2057e−2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(−3.2841e−2)</td>
<td>0.0603</td>
<td>3.2984e−6</td>
</tr>
</tbody>
</table>

[1]: The CoM vector for these bodies are derived by subtracting the CoM of the mirror bodies from their length. i.e. $c_5 = d_3 - c_3$, $c_6 = d_2 - c_2$, $c_7 = d_1 - c_1$

Moreover, the linearized mass-inertia (M) and gravity matrices (G) are given in Equations (C.1) and (C.2), respectively.
C.2 Actuator Parameters

There are two types of motors used in the joints. The two ankle lateral joints use Kollmorgen RBE0713 and the rest of the joints that are modelled in this paper use Kollmorgen RBE1211. All motors are configured with A-windings. The parameters of these motors are given in table C.2.

All the joints that are modelled in this paper use harmonic drive gearbox CSD17-2A with reduction ratio of 100 : 1. The gearbox inertia is $5.4e-6 \text{ (kg.m}^2\text{)}$ which is added to the motor inertia and then reflected to the joint (multiplied by the reduction ratio squared). The gearbox minimum stiffness is $0.84e4 \text{ (N.m/rad)}$ that is added in series to the passive compliant value of the joint.
Table C.2: CoMan’s actuator parameters.

<table>
<thead>
<tr>
<th>Joints</th>
<th>$q_1$</th>
<th>$q_2$</th>
<th>$q_3$</th>
<th>$q_4$</th>
<th>$q_5$</th>
<th>$q_6$</th>
<th>$q_7$</th>
<th>$q_8$</th>
<th>$q_9$</th>
<th>$q_{10}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$J (kg.m^2)$</td>
<td>0.0787</td>
<td>0.3121</td>
<td>0.1387</td>
<td>0.1387</td>
<td>0.3121</td>
<td>0.1387</td>
<td>0.1387</td>
<td>0.3121</td>
<td>0.1387</td>
<td>0.0787</td>
</tr>
<tr>
<td>$K_s (Nm/rad)$</td>
<td>8400</td>
<td>188.11</td>
<td>185.8</td>
<td>8400</td>
<td>981.74</td>
<td>981.74</td>
<td>8400</td>
<td>185.8</td>
<td>188.11</td>
<td>8400</td>
</tr>
<tr>
<td>$B_m (Nmsec/rad)$</td>
<td>22.247</td>
<td>57.411</td>
<td>25.516</td>
<td>25.516</td>
<td>57.411</td>
<td>25.516</td>
<td>25.516</td>
<td>57.411</td>
<td>22.247</td>
<td></td>
</tr>
</tbody>
</table>
Appendix D

LMI-LQR Simulation for the Compass Gait

In this appendix, numerical results of the LMI-LQR formulation is shown on a two degree of freedom rigid and compliant compass gait model to illustrate the results on a low dimensional (2 DoF) model.

D.1 The Rigid Compass Gait

The rigid compass gait model of walking is used as a toy example with 2 DoF to illustrate the numerical results of Algorithm (4.20). A picture of the compass gait model is shown in Fig. D.1. All masses are considered to be point masses, where each leg has a lumped mass $m$ that is located in the middle of each leg and the hip has mass $m_H$. The input ankle/hip torques are assumed to be provided via rigid actuators (as ideal torque sources). The system is discretized with 1 ms and the augmented

![Figure D.1: Planar compass gait with hip and ankle actuation.](image-url)
The system is formulated as in (4.3). The mechanical parameters of the compass gait model is provided in Table D.1. The linearized mass-inertia and gravity matrices are

\[
M = \begin{bmatrix}
2.5 & -0.25 \\
-0.25 & 0.25
\end{bmatrix}, \quad G = \begin{bmatrix}
-29.43 & 4.905 \\
4.905 & 4.905
\end{bmatrix}.
\] (D.1)

The joint ordering in this formulation should be changed to group the state space variables of each joint together via a permutation matrix. This matrix groups the joint variables into blocks that consist of \([q_i, \dot{q}_i, z_i]^T\) that are discrete time joint position, joint velocity and integrator of \(i^{th}\) joint. The LQR state and input penalties in the original coordinates are \(Q = \text{diag}\{100 \ I_n, \ 100 \ I_n, \ 20 \ I_n\}\) and \(R = 0.1 \ I_n\) and the input noise variance is \(\beta = 0.01\). \(I_n\) is an identity matrix of size \(n = 2\). The discrete time LQR gains are

\[
\begin{bmatrix}
K_a \\
K_h
\end{bmatrix} = \begin{bmatrix}
1667 & 96.012 & -13.861 & -78.112 & -6.6395 & 0.090536 \\
-79.222 & -6.6819 & 0.10229 & 971.1 & 36.302 & -12.99
\end{bmatrix}
\]

The decentralized gains are

\[
\begin{bmatrix}
K_a \\
K_h
\end{bmatrix} = \begin{bmatrix}
1787.8 & 109.12 & -13.861 & 0 & 0 & 0 \\
0 & 0 & 0 & 1085.5 & 44.309 & -13.119
\end{bmatrix}
\]

The decentralized closed loop step response and the corresponding torques are shown in Figs. D.2 and D.3. This algorithm can be used to derive the decentralized gains for
higher bandwidths by reducing the penalty $R$ on the actuators. In the next section, Algorithm (4.20) is applied to a rigid model of CoMan with 10 DoF. In section D.2, the compliant compass gait model with actuator dynamics is used to derive a PD gain on the motor states and a PID on the link. The ankle and hip are assumed to be actuated. The compass gait model in [162] is modified to include compliance and drive dynamics to demonstrate the use of this method.

## D.2 The Compliant Compass Gait

In this section, compliance and actuator dynamics are integrated to the dynamic model (3.1) and the decentralized feedback gains are computed. Then the symmetry constraint is imposed and the decentralized-symmetric feedback gains are derived. The compliance and drive system for both ankle and hip are assumed to be the same. The motor inertia matrix $J = \text{diag}\{J_m, J_m\}$, the stiffness matrix $K_s = \text{diag}\{k_s, k_s\}$, the motor damping matrix $D_m = \text{diag}\{d_m, d_m\}$, the voltage to torque gain matrix $V_m = \text{diag}\{vt, vt\}$. In practice, the structural damping matrix $D_s = \text{diag}\{d_s, d_s\}$ is very small and it can be approximated by zero. The parameters of these matrices are given in Table D.1. The system is discretized with 1 ms and the augmented system is formulated as in (3.11). The linearized mass-inertia and gravity matrices are given.
in (D.1).

The joint ordering in this formulation should be changed to group the state space variables of each joint together via a permutation matrix. This matrix groups the joint variables into blocks that consist of \([q_i, \dot{q}_i, q_{mi}, \dot{q}_{mi}, z_i]^T\) that are discrete time joint position, joint velocity, motor position, motor velocity and integrator of the \(i\)th joint. The LQR state and input penalties in the original coordinates are \(Q = \text{diag}\{20 \ I_n, \ 0, \ 5 \ I_n\}\) and \(R = I_n\) and the input noise variance is \(\beta = 0.01\). \(I_n\) is an identity matrix of size \(n\) and \(0\) is a zero vector of appropriate dimension. The penalties have to be transformed under the permutation matrix. The LMI formulation has 43 decision variables and after 72 iterations the decentralized LQR-LMI gains are computed as

\[
\begin{bmatrix}
K_a \\
K_h
\end{bmatrix} =
\begin{bmatrix}
342.51, \ 45.49, \ 161.56, \ 0.80, \ -2.20 \\
-53.72, \ 11.94, \ 323.96, \ 1.49, \ -2.16
\end{bmatrix}
\]

where, \(K_a, K_h\) are the PD-PID gains for the ankle and hip respectively. The zero gain blocks that appeared in (4.10) are omitted for convenience. The gains are ordered in a similar order as the permuted state space vector that is a PD on joint and a PD on motor while the last gain is the link integral gain. Furthermore, using the same LQR penalties, a symmetry constraint is imposed on ankle and hip gains that reduces the number of decision variables to 23 and after 83 iterations the decentralized-symmetric gains are

\[
K_a = K_h = [-82.48, \ 125.99, \ 956.33, \ 3.59, \ -2.06]
\]

Comparing the closed loop eigenvalues for the decentralized and decentralized-symmetric feedback design shows that the later is slower, i.e. the largest eigenvalue for the decentralized gain is 0.9939 (time constant of 163 ms) and the largest eigenvalue for the decentralized-symmetric design is 0.9976 (time constant of 416 ms). Since the step response is dominated by the slowest time constant, the latter design is approximately 253 ms slower than the former design. However, this is not a limitation and a faster response can be computed for the decentralized-symmetric feedback just by reducing the input penalty \(R\). Fig. D.4 depicts the step response of the decentralized closed loop systems. Fig. D.5 shows the corresponding control signals as motor voltages.
APPENDIX D. LMI-LQR SIMULATION FOR THE COMPASS GAIT

Figure D.4: Closed-loop step response of actuated compass gait.

Figure D.5: Control inputs of compass gait closed-loop step response.

Table D.1: Compass gait parameters

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
<th>Value</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>$m_H$</td>
<td>Hip mass</td>
<td>2</td>
<td>kg</td>
</tr>
<tr>
<td>$m$</td>
<td>Leg mass</td>
<td>1</td>
<td>kg</td>
</tr>
<tr>
<td>$l$</td>
<td>Leg length</td>
<td>1</td>
<td>m</td>
</tr>
<tr>
<td>$g$</td>
<td>Gravity constant</td>
<td>9.81</td>
<td>$kg.m^2/s^2$</td>
</tr>
<tr>
<td>$J_m$</td>
<td>Motor inertia</td>
<td>0.1387</td>
<td>$kg.m^2$</td>
</tr>
<tr>
<td>$d_m$</td>
<td>Motor damping</td>
<td>25.5158</td>
<td>$Nmsec/rad$</td>
</tr>
<tr>
<td>$k_s$</td>
<td>Joint stiffness</td>
<td>185.7974</td>
<td>$Nm/rad$</td>
</tr>
<tr>
<td>$vt$</td>
<td>Voltage to torque gain</td>
<td>6.1747</td>
<td>$N.m/Volt$</td>
</tr>
</tbody>
</table>
Appendix E

Calibration and Polarity Data

The first step before each experiment was to make sure that the sensor readings are accurate and also the input output polarity of the feedback loops are identified correctly. These information are provided in this appendix.

Table E.1: CoMan’s calibration data (Angles in degree and link encoder positions are in counts).

<table>
<thead>
<tr>
<th>Left leg’s joints</th>
<th>Hip (S)</th>
<th>Hip (L)</th>
<th>Knee</th>
<th>Ankle (S)</th>
<th>Ankle (L)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Min angle</td>
<td>-25</td>
<td>-45</td>
<td>-92</td>
<td>-18</td>
<td>-13</td>
</tr>
<tr>
<td>Min link encoder</td>
<td>2590</td>
<td>2558</td>
<td>5093</td>
<td>2818</td>
<td>4091</td>
</tr>
<tr>
<td>Min motor encoder</td>
<td>9617</td>
<td>8717</td>
<td>16006</td>
<td>13168</td>
<td>9718</td>
</tr>
<tr>
<td>Max angle</td>
<td>90</td>
<td>22</td>
<td>0</td>
<td>20</td>
<td>26</td>
</tr>
<tr>
<td>Max link encoder</td>
<td>3872</td>
<td>1804</td>
<td>4045</td>
<td>3259</td>
<td>3687</td>
</tr>
<tr>
<td>Max motor encoder</td>
<td>200041</td>
<td>84212</td>
<td>119789</td>
<td>99407</td>
<td>50868</td>
</tr>
</tbody>
</table>
Table E.2: CoMan’s calibration data (Angles in degree and link encoder positions are in counts).

<table>
<thead>
<tr>
<th>Right leg’s joints</th>
<th>Hip (S)</th>
<th>Hip (L)</th>
<th>Knee</th>
<th>Ankle (S)</th>
<th>Ankle (L)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Min angle</td>
<td>-25</td>
<td>-45</td>
<td>-92</td>
<td>-18</td>
<td>-13</td>
</tr>
<tr>
<td>Min link encoder</td>
<td>1901</td>
<td>3330</td>
<td>2732</td>
<td>3390</td>
<td>3628</td>
</tr>
<tr>
<td>Min motor encoder</td>
<td>203484</td>
<td>86445</td>
<td>115058</td>
<td>95061</td>
<td>50991</td>
</tr>
<tr>
<td>Max angle</td>
<td>90</td>
<td>22</td>
<td>4</td>
<td>20</td>
<td>26</td>
</tr>
<tr>
<td>Max link encoder</td>
<td>585</td>
<td>4084</td>
<td>3806</td>
<td>2951</td>
<td>4036</td>
</tr>
<tr>
<td>Max motor encoder</td>
<td>11668</td>
<td>10408</td>
<td>11099</td>
<td>4637</td>
<td>10273</td>
</tr>
</tbody>
</table>

Table E.3: CoMan’s input (voltage)-output (motor/joint position) polarity convention.

<table>
<thead>
<tr>
<th>Joints</th>
<th>(q_1)</th>
<th>(q_2)</th>
<th>(q_3)</th>
<th>(q_4)</th>
<th>(q_5)</th>
<th>(q_6)</th>
<th>(q_7)</th>
<th>(q_8)</th>
<th>(q_9)</th>
<th>(q_{10})</th>
</tr>
</thead>
<tbody>
<tr>
<td>Voltage-Motor position</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>Voltage-Joint position</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
</tbody>
</table>
Appendix F

Video Material

Video material plays an essential role in experiments in robotics. Therefore, this thesis is accompanied by a collection of videos to demonstrate the experimental results explained in chapter 8. This appendix provides a quick introduction to the video clips which are uploaded to the following web-site:


The first video is related to the experiment on a single compliant joint which was described in section 8.1. The title of the video is “Sine wave tracking using a compliant joint” and direct link to the video is

http://www.youtube.com/watch?v=X7RniBY1ke0.

The next set of videos are related to the initial experiments which were carried out on iCub over a period of one year to study the feasibility of walking. These videos are:

1. Initially the knee joint was controlled via observers and LQR in a stable position, where a square wave with the period of 4 sec is used in this experiment, (link: http://www.youtube.com/watch?v=zi_6Yj1Yp0Q).

2. A similar experiment is repeated with a sine wave, (link: http://www.youtube.com/watch?v=yWRkCGRGyGA).

3. In the next step, both hip and knee were controlled by observer based LQR (link: http://www.youtube.com/watch?v=ny9XDkWhZII).
4. Subsequently, all the joints of the right leg were controlled (link: front view http://www.youtube.com/watch?v=ijqXUrZxQL4, and the same test with side view http://www.youtube.com/watch?v=CZq04AmTImw).

5. In the next step, the observer based LQR was applied to the robot’s unstable upper body with two lateral and sagittal joints. The difference between this experiment and the previous experiments is the fact that the upper body is similar to an unstable inverted pendulum, (link: http://www.youtube.com/watch?v=_BiOy_r2KMM).

6. Furthermore, the control method was extended to apply to both legs and the torso at the same time in order to prepare the experiment for the final goal of applying the LQR stabilization on the ground, (link: http://www.youtube.com/watch?v=aatD2iRcBB0).

Once the upgraded robot, CoMan was developed, a series of experiments were conducted, including regulation on the ground, sway and squat motions. The links to these videos are:

1. Applying the centralized observer based LQR to CoMan, with external perturbation (link: http://www.youtube.com/watch?v=Jckq6NoIdqc),

2. a sway test on CoMan, (link: http://www.youtube.com/watch?v=HJ7fxwm-r0o),

3. a slow squat test (link: http://www.youtube.com/watch?v=0XRqtqFQjII),
   and a faster squat test (link: http://www.youtube.com/watch?v=AUHioOs-s04).

Finally, the video of the limit cycle described in the technical challenges section in chapter 8 can be viewed at http://www.youtube.com/watch?v=XYQKQB2Uak4.
Bibliography


[40] (2012), Online photo, website: http://www.jst.go.jp/pr/info/info461/zui_e.html.


[64] S. Blank, T. Wahl, T. Luksch and K. Berns, “Biologically Inspired Compliant Control of a Monopod Designed for Highly Dynamic Applications”, *in


Modular Bio-Inspired Architecture for Movement Generation for the Infant-Like Robot iCub”, in Proc. IEEE RAS and EMBS Int. Conf. on Biomedical Robotics and Biomechatronics (BioRob), 2008, pp. 795-800.


[152] (2012), iCub Simulator Website, [online], Available: http://eris.liralab.it/wiki/ODE.


