

**A DECENTRALIZED PATTERN GENERATOR FOR THE
MOTION GENERATION OF AN UNDER-ACTUATED
HUMANOID ROBOT NUSTBOT-3**



Author

ZAID AHSAN SHAH

Registration Number

NUST201362042MSMME62113F

Supervisor

DR. YASAR AYAZ

Co-supervisor

DR. FAHAD MUMTAZ MALIK

DEPARTMENT OF ROBOTICS & ARTIFICIAL INTELLIGENCE
SCHOOL OF MECHANICAL & MANUFACTURING ENGINEERING
NATIONAL UNIVERSITY OF SCIENCES & TECHNOLOGY
ISLAMABAD, PAKISTAN

AUGUST, 2017

**A Decentralized Pattern Generator for the Motion Generation of
an Under-actuated Humanoid Robot NUSTBOT-3**

Author

ZAID AHSAN SHAH

Registration Number

NUST201362042MSMME62113F

A thesis submitted in partial fulfilment of the requirements of the
degree of MS Robotics & Intelligent Machines Engineering

Thesis Supervisor

Dr. Yasar Ayaz

Thesis Co-Supervisor

Dr. Fahad Mumtaz Malik

Thesis Acceptance Certificate

It is certified that the final copy of MS Thesis written by Zaid Ahsan Shah (Registration No. NUST201362042MSMME62113F), of SMME (School of Mechanical & Manufacturing Engineering) has been vetted by undersigned, found complete in all respects as per NUST statutes / regulations, is free of plagiarism, errors and mistakes and is accepted as partial fulfilment for award of MS/MPhil Degree. It is further certified that necessary amendments as pointed out by GEC members of the scholar have also been incorporated in this dissertation.

Signature: _____

Name of Supervisor: Dr. Yasar Ayaz

Date: _____

Signature (HOD): _____

Date: _____

Signature (Principal): _____

Date: _____

Declaration

I, Zaid Ahsan Shah, certify that this research work titled “A Decentralized Pattern Generator for the Motion Generation of an Under-Actuated Humanoid Robot NUSTBOT-3” is my own work. The work has not been presented elsewhere for assessment. The material that has been used from other sources has been properly acknowledged / referred.

Signature of Student

ZAID AHSAN SHAH

NUST201362042MSMME62113F

Plagiarism Certificate

(Turnitin Report)

This thesis has been check for Plagiarism. Turnitin report endorsed by the Supervisor is attached at the end of this report.

Signature of Student

ZAID AHSAN SHAH

NUST201362042MSMME62113F

Signature of Supervisor

Copyright Statement

- Copyright in text of this thesis rests with the student author. Copies (by any process) either in full, or of extracts, may be made only in accordance with instruction given by the author and lodged in the Library of NUST School of Mechanical & Manufacturing Engineering (SMME). Details may be obtained by the Librarian.
- This page must form part of any such copies made. Further copies (by any process) may not be made without the permission (in writing) of the author.
- The ownership of any intellectual property rights which may be described in this thesis is vested in NUST School of Mechanical & Manufacturing Engineering, subject to any prior agreement to the contrary, and may not be made available for use by third parties without the written permission of the SMME, which will prescribe the terms and conditions of any such agreement.
- Further information on the conditions under which disclosures and exploitation may take place is available from the Library of NUST School of Mechanical & Manufacturing Engineering, Islamabad.

Acknowledgements

In all honesty to state that this work, the formulation of a new approach for making robots walk and the development of the associated hardware of the robot NUSTBOT-3 and the research work completed and published under this study, is a one man show, is a gross understatement. It would be an impossible feat, had it not been for a plethora of encouraging people around me who kept motivating and supporting me throughout this work.

I am forever indebted to my parents, who have given me more than I can possibly even imagine. In all the challenging and difficult stages I faced during this work they were my compass and always directed me towards the true north. I would like to thank my brother and sister who, in their own ways, have served as my shrinks to help me balance everything in critical times.

I have the utmost gratitude for my supervisors Dr. Yasar Ayaz and Dr. Fahad Mumtaz Malik, who helped me build my work from a mere concept to the grandiose form it has today. I have questioned them and asked for their assistance at the most bizarre of times, midnight or public holidays, in the toughest and most challenging matters and their motivation and willingness to guide me, in every way, on every day, has been of paramount significance to help me through all of the highs and lows of my thesis.

I thank my GEC members for their valuable suggestions and practical guidelines to improve my work and extend it further for future students and researchers to build upon.

For the due credits I find it necessary to mention all the technical staff at the Department of Manufacturing & Resource Center (DMRC), Mr. Faraz, Mr. Shouqat, Mr. Nisar and Mr. Wajid under the supervision of Col. (Retd.) Naveed. I would also like to offer my greatest gratitude to my machinist Mr. Ayub Sheikh (Bhola) and both his sons for their tireless efforts to assist me in the mechanical fabrication of the robot NUSTBOT-3.

My sincerest regards are due towards the current and past student members and the lab technicians of RISE Labs, CVML & Advanced Control Systems Lab, HSL, CAD/CAM Labs, Prosthetics and Bio-Chemistry Labs for their contributions and assistance towards my work.

*Dedicated to the lost ones who are finding the way
and to the ones who found it and lost it anyway.*

Abstract

The Central Pattern Generator for Humanoid Robots based on Preview Control has certain limitations at which the system becomes unstable or unsolvable. Two of these corner cases are lack of actuation (reduction in the number of actuated degrees of freedom) and Zero-Moment Point moving outside the ground contact polygon. A Decentralized Pattern Generator is presented in this study to overcome these limitations and the suggested approach is developed as a multi-rigid-body dynamics solution which is a step towards finding a necessary condition for humanoid robot stability instead of a sufficient condition, as suggested by the humanoid robotics research community.

The formulation of the pattern generation problem as a multi-body dynamics problem is a key part in deriving the DPG for under-actuated humanoid robots. A number of robot models with fully-actuated and under-actuated configuration are analyzed and motion generation is performed on 5 different cases in this study. Rigid-body dynamics and multi-body dynamics are used to incorporate the robot hardware along with the choice of joint actuators in the motion generation process for the robot. An optimal control system topology for the DPG is also presented in this study.

The DPG uses an iterative approach for solving the equations of motion and joint forces/torques under the position and velocity constraints imposed on the robot hardware. An under-actuated humanoid robot NUSTBOT-3 is designed and developed with un-actuated ankles and a ROS based solution incorporating the decentralized pattern generator is implemented to overcome the limitations of the central pattern generator. The multi-body dynamics simulation and analyses are performed to verify the solutions generated by the decentralized pattern generator.

Table of Contents

| | |
|--|-----------|
| Abstract..... | x |
| List of Figures | xiv |
| List of Abbreviations | xvi |
| 1 Introduction | 1 |
| 1.1 Bio-Inspiration vs Bio-Mimicry..... | 1 |
| 1.2 Human Centered Design | 2 |
| 1.3 Under-Actuated Robotics..... | 3 |
| 1.4 Existing Solutions for Legged Locomotion | 4 |
| 1.5 Challenges of Legged Locomotion | 4 |
| 1.6 Challenges of Under-Actuated Robotics | 5 |
| 1.7 Under-Actuated Legged Robotics | 6 |
| 1.8 Thesis Overview | 8 |
| 2 Literature Review | 10 |
| 2.1 Walking Mechanisms | 10 |
| 2.1.1 Passive Dynamic Walkers..... | 10 |
| 2.1.2 Foot Placement Based Robots | 11 |
| 2.1.3 Hybrid Zero Dynamics Based Systems | 14 |
| 2.1.4 Central Pattern Generators Based Robots..... | 15 |
| 2.2 Dynamics of Legged Locomotion | 16 |
| 2.3 Balance and Gait Stability | 17 |
| 2.4 Dynamical Control of Motion..... | 19 |
| 2.5 Rigid Body Dynamics | 20 |
| 2.6 Multi-Body Dynamics..... | 21 |
| 2.7 Under-Actuated Robotics..... | 21 |
| 2.8 Pattern Generator as a Multi-Body Dynamics Problem | 22 |
| 3 Decentralized Pattern Generation | 23 |
| 3.1 Multi-body Description of Robot Systems | 23 |
| 3.1.1 Assumptions..... | 23 |
| 3.1.2 Motion Primitives and Motion Constraints | 23 |
| 3.1.3 Contact Model..... | 27 |
| 3.1.4 Dynamics of a Rigid Body | 27 |
| 3.1.5 Linear & Angular Momenta of a Rigid Body..... | 28 |
| 3.1.6 Stable Pose..... | 29 |
| 3.1.7 Pose Recoverability Margin | 29 |

| | | |
|-------|---|----|
| 3.1.8 | Action Subspaces | 33 |
| 3.2 | Motion Generation Process of a DPG | 34 |
| 3.2.1 | Input Via-Poses | 34 |
| 3.2.2 | Position Inverse Kinematics Check..... | 34 |
| 3.2.3 | Spline Generator | 34 |
| 3.2.4 | Pose Recoverability Margin | 35 |
| 3.2.5 | Velocity Inverse Kinematics Check..... | 35 |
| 3.2.6 | Equation of Motion Solver | 35 |
| 3.2.7 | Force Controller & Actuator Drive | 35 |
| 3.2.8 | Flowchart | 37 |
| 3.3 | Mathematical Derivation of Decentralized Pattern Generators | 38 |
| 3.4 | CASE I: Planar Single Inverted Pendulum..... | 40 |
| 3.4.1 | Mechanical Design | 40 |
| 3.4.2 | Stable Pose | 41 |
| 3.4.3 | Target Trajectory..... | 41 |
| 3.4.4 | DPG Solution | 41 |
| 3.5 | CASE II: Fully-actuated Double Pendulum | 42 |
| 3.5.1 | Mechanical Design | 42 |
| 3.5.2 | Stable Pose..... | 42 |
| 3.5.3 | Target Trajectory..... | 43 |
| 3.5.4 | DPG Solution | 43 |
| 3.6 | CASE III: Under-actuated Double Pendulum (Acrobot) | 43 |
| 3.6.1 | Mechanical Design | 43 |
| 3.6.2 | Stable Pose | 43 |
| 3.6.3 | Target Trajectory..... | 44 |
| 3.6.4 | DPG Solution | 44 |
| 3.7 | CASE IV: Fully-actuated 12 DOF Humanoid Robot - NUSTBOT-3 | 44 |
| 3.7.1 | Mechanical Design | 45 |
| 3.7.2 | Stable Pose | 46 |
| 3.7.3 | Target Trajectory..... | 48 |
| 3.7.4 | DPG Solution | 48 |
| 3.8 | CASE V: Under-actuated 12 DOF Humanoid Robot - NUSTBOT-3 | 48 |
| 3.8.1 | Mechanical Design | 48 |
| 3.8.2 | Stable Pose..... | 50 |
| 3.8.3 | Target Trajectory..... | 50 |

| | | |
|-------|---|----|
| 3.8.4 | DPG Solution | 50 |
| 3.9 | Final Note on Pattern Generators..... | 50 |
| 4 | Analysis & Results | 51 |
| 4.1 | Multi-body Simulation | 51 |
| 4.2 | DPG Motion Generation | 52 |
| 4.3 | CASE I: Single Inverted Pendulum..... | 52 |
| 4.4 | CASE II: Fully-actuated Double Pendulum | 54 |
| 4.5 | CASE III: Under-actuated Double Pendulum..... | 57 |
| 4.6 | CASE IV: Fully-actuated 12 DOF Humanoid Robot - NUSTBOT-3 | 58 |
| 4.7 | CASE V: Under-actuated 12 DOF Humanoid Robot - NUSTBOT-3 | 61 |
| 4.8 | Comparison DPG & CPG..... | 63 |
| 4.9 | Stability vs Recoverability | 63 |
| 5 | Conclusions | 64 |
| 5.1 | Decentralized Pattern Generators..... | 64 |
| 5.2 | Robot Design Consideration | 65 |
| 5.3 | Target Trajectory & Motion Execution | 65 |
| 5.4 | Stability Criteria..... | 66 |
| 6 | Future Work | 67 |
| 6.1 | Pattern Generation for Online Walking..... | 67 |
| 6.2 | Generalized Pattern Generation..... | 67 |
| 6.3 | Stability for DPG..... | 68 |
| 6.4 | Hardware Experimentation..... | 68 |
| | References | 69 |

List of Figures

| | |
|--|----|
| Figure 1.1 HONDA ASIMO kicking a football towards a human [98]..... | 2 |
| Figure 1.2 Competitors for DAPRA Robotics Challenge 2015 [99]..... | 3 |
| Figure 1.3 Cubli - An under-actuated system by ETH Zurich [100]..... | 3 |
| Figure 1.4 Cubli: lifting up and balancing action [100]..... | 6 |
| Figure 1.5 NUSTBOT-3 Side View..... | 8 |
| Figure 2.1 Passive Dynamic Walker developed by Collins et.al. [101] | 10 |
| Figure 2.2 Passive Dynamic Walker Sano Laboratory, Nagoya Institute of Technology | 11 |
| Figure 2.3 Poincaré map based single-legged hopper by Leg Lab, MIT [102]..... | 12 |
| Figure 2.4 PETMAN developed by Boston Dynamics [103] | 13 |
| Figure 2.5 Boston Dynamics BigDog walking on natural uneven terrains [104]..... | 13 |
| Figure 2.6 Hybrid Zero Dynamics based walking robots [105], [106], [107] | 14 |
| Figure 2.7 Humanoid Robot Projects (HRP) [108] | 15 |
| Figure 2.8 Phases of Biped Walking..... | 16 |
| Figure 2.9 MBD based Spatial COM Trajectories used by DPG | 17 |
| Figure 2.10 Spatial Vectors for a rigid body in 3D space..... | 20 |
| Figure 3.1 Force Transmission through the joint i between adjacent links $i - 1$ and i | 24 |
| Figure 3.2 A Simple Inverted Pendulum | 30 |
| Figure 3.3 Phase Diagram (Vector Field) of a Simple Inverted Pendulum | 31 |
| Figure 3.4 Pose Recoverability vs ZMP Stability..... | 32 |
| Figure 3.5 DPG as a Control System Architecture | 36 |
| Figure 3.6 Decentralized Pattern Generation Flowchart..... | 37 |
| Figure 3.7 Cylindrical Single Pendulum..... | 41 |
| Figure 3.8 The Double Inverted Pendulum Model. | 42 |
| Figure 3.9 Double Pendula; (a) Fully-Actuated (b) Under-Actuated | 44 |
| Figure 3.10 NUSTBOT 3 Simulation Model..... | 45 |
| Figure 3.11 NUSTBOT-3 Revolute Joint Axes in right leg. | 46 |
| Figure 3.12 NUSTBOT-3 in the Stable Pose..... | 47 |
| Figure 3.13 The Humanoid Robot NUSTBOT-3 Hardware Model with un-torqued ankles .. | 49 |
| Figure 4.1 Angle of the single inverted pendulum with only torque constraints..... | 52 |
| Figure 4.2 Angle trajectory comparison DPG vs PI Controller..... | 53 |
| Figure 4.3 Sustained Oscillations of the Single Inverted Pendulum | 54 |

| | |
|--|----|
| Figure 4.4 Fully Actuated Double Pendulum Angles under External Disturbance..... | 55 |
| Figure 4.5 Fully Actuated Double Pendulum - End Effector Oscillation with Stiff Base..... | 55 |
| Figure 4.6 Fully Actuated Double Pendulum - Oscillation with Stiff Elbow..... | 56 |
| Figure 4.7 Fully Actuated Double Pendulum - Oscillations with proportional actuation | 56 |
| Figure 4.8 Swing-up trajectory angle profiles of the Under-Actuated Double Pendulum | 57 |
| Figure 4.9 NUSTBOT-3 Walking Via Poses..... | 58 |
| Figure 4.10 ZMP trajectory during step for fully-actuated NUSTBOT-3 by CPG | 59 |
| Figure 4.11 ZMP trajectory during step for fully-actuated NUSTBOT-3 by DPG..... | 59 |
| Figure 4.12 Right ankle pitch torque, generated by CPG, during support phase | 60 |
| Figure 4.13 Right ankle pitch torque, generated by DPG, during support phase | 60 |
| Figure 4.14 Via-Poses for the NUSTBOT-3 walking trajectory for the DPG..... | 61 |
| Figure 4.15 COM Profile for NUSTBOT-3 walking motion generated by a DPG | 62 |
| Figure 4.16 ZMP Profile for NUSTBOT-3 walking motion generated by a DPG | 62 |

List of Abbreviations

| | |
|--------|---|
| DPG | Decentralized Pattern Generator |
| CPG | Central Pattern Generator |
| DOF | Degrees of Freedom |
| COM | Center of Mass |
| ZMP | Zero-Moment Point |
| RBD | Rigid-Body Dynamics |
| MBD | Multi-Body Dynamics |
| MABEL | Michigan Anthropomorphic Biped with Electronic Legs |
| NASA | National Aeronautics and Space Administration |
| DARPA | Defense Advanced Research Project Agency |
| HRP | Humanoid Robotics Project |
| ASIMO | Advanced Step in Innovative Mobility |
| PETMAN | Protection Ensemble Test Mannequin |
| PI | Proportional-Integral Controller |

1 Introduction

One of the most popular frontiers of cutting-edge science, engineering and technology is nature-inspired or bio-inspired robotics, and among those legged and humanoid robots are arguably at the forefront [1-5]. A number of interesting system designs have been developed recently which capture the human-like architecture for walking robots brilliantly [6-8]. The rapid research and development of a number of legged robots, besides humanoid robots, of varying shapes and sizes have also undergone considerable development in the recent years [9-12]. The increased interest in legged robotics is primarily on accounts of “natural” characteristics, such as stepping motion and humanlike or animal-like maneuverability.

A number of challenges are associated with the advanced legged robots and the current techniques for solving these challenges are ingenious in their own sense[13-16][17] but they have their defined shortcomings. In this study some of the limitations of the existing techniques for making legged robots move are addressed with primary focus towards humanoid robots.

1.1 Bio-Inspiration vs Bio-Mimicry

Bio-mimicry is the domain of science which deals with adopting natural solutions to the existing problems. Bio-inspiration is a slightly different field in which insights are drawn as how the said problems are addressed in natural systems and generating solutions which perform in a similar manner. Generally natural systems have impressive designs and optimized performance on accounts of billions of years of repeated evolution. This has resulted in extraordinary machines which are not only efficient in the way they perform but also elegant in the manner in which they operate. However copying the technique in which natural machines work is less fruitful since these systems are varied in performance and diverse in applications.

An albatross can fly across oceans without the slightest change in its metabolism or the night owl can hunt its prey with as little sound as that of rustling tree leaves. A flying robot can have similar shape to a bird [18], [19] or it can have a completely unnatural mechanical design [20], but both can draw insights from natural systems to enhance their flight. A cheetah or an ostrich can accelerate from standing to bounding gaits within a couple of steps. Their bodies are designed to perform in a diverse set of environments and in different states of action. Robots which draw inspiration from these animals can have similar attributes [9], [21]. These systems or animals are fascinating in their own regards, however when it comes to utility, the solutions which are to be employed are application oriented and hence a better performance is desired in a certain domain instead of an overall average performance.

Bio-inspired robotics [22-25] is similar to, but not the same as bio-mimetic robotics, in the sense that it is focused around techniques which are found in nature, instead of the physical structure. In the field of applied legged robotics bio-inspiration is far more valuable than bio-mimicry solely for the reason that it can lead to intelligent machines, performing closely linked but different tasks instead of an automated animal which does everything.

1.2 Human Centered Design

Manmade environment, machinery and equipment is designed with the intent to be used by humans, everything from stairs to cellphones are structured so as to best suit for human utility. Naturally it follows that if robots are to be introduced in human lives, they should not only interact with the man-made environment and equipment but that too in a manner similar to humans. It is far more convenient and feasible to devise a robot system which fits in to the human lifestyle than to reconfigure the entire surrounding to suit the robotic system.



Figure 1.1 HONDA ASIMO kicking a football towards a human [98]

A number of events which promote the human-centered design have been underway in recent years which have encouraged the development of human-like machinery or human-like behavior. Two of the most prominent events in recent years are the DARPA Robotics Challenge [26], [27] and the International RoboCup [28] which challenge researchers to develop systems which are human-like and perform in human-like environments for tasks such as rescue and reconnaissance and multi-agent coordination.

The element of human-centered design is not only limited to robotics but this is also a key factor which influences the impact any technology will have on the human-life in general. Thus it is not surprising that newer robotic applications are emerging in the fields such as human-robot interaction, pioneered by humanoid robots.



Figure 1.2 Competitors for DAPRA Robotics Challenge 2015 [99]
SCHAFT by Google (left), Valkyrie by NASA (center), PETMAN by Boston Dynamics (right)

1.3 Under-Actuated Robotics

Under-Actuated Robotics [29] is the branch of robotics which deals with reduced actuation. The term under-actuation implies the robot's DOF are not fully actuated. Formally a robot is said to be under-actuated if in one or more states, it cannot be accelerated along any of its unconstrained degrees of freedom.

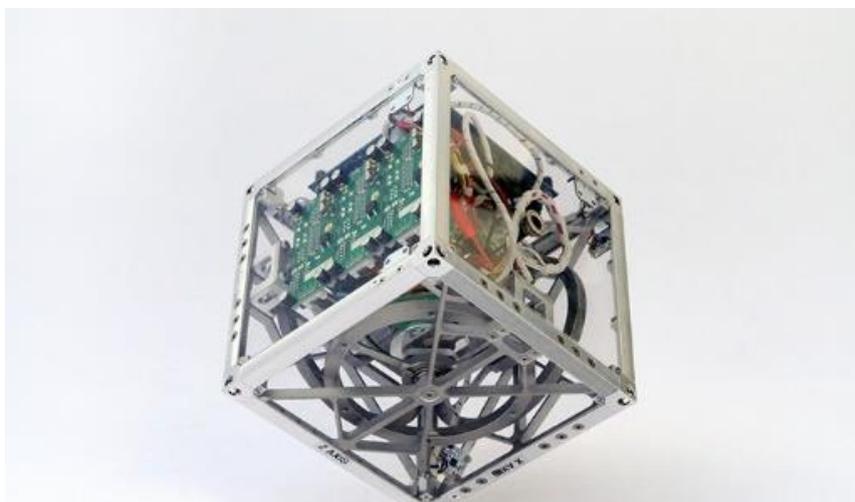


Figure 1.3 Cubli - An under-actuated system by ETH Zurich [100]

The conventional approach for using high gain feedback control system for overriding the dynamical interplay of the mechanisms cannot be applied to under-actuated systems. In case of under-actuation the robot control is accomplished through very interesting techniques which employ the dynamics of the different robot segments. Two of these techniques namely multi-body dynamics and rigid-body dynamics are employed in this study to generate solutions for a few robots.

By definition a humanoid robot, which can walk in the 3D space, is an under-actuated system, since at every given moment it cannot be given a linear or angular acceleration along all three axes for each of its segments. In this study, however, to avoid conflict the reduced actuation definition is followed.

1.4 Existing Solutions for Legged Locomotion

Legged locomotion [12], [30-33] is a broad research field which deals with a lot of robots such as bipeds, quadrupeds and hexapods and their locomotion is of particular interest in the research communities. There have been a number of solutions which solve the walking problem for legged robots under certain assumptions.

One of the most important techniques for generating walking motion in legged robots is central pattern generation [1], [14], [34-36], which generates an actuation sequence for all the joints of a robot in order to make it walk all the while adhering to a stability criteria set forth by geometric descriptions of a mechanical entity such as the COM or the ZMP. The central pattern generators (CPGs) have the requirement that the robot under consideration must have full actuation, i.e. all its joints must be torqued.

One of the drawbacks of central pattern generators is a typically very stiff ankle joint (very high torque requirement) and a loss of solution in case of reduced actuation (or under-actuation). A preview control based application technique which estimates the stability of the motion prior to execution has been proposed with great success [37]. However a preview control based central pattern generation solution for walking robots essentially checks for a sufficient condition for robot stability and not a necessary condition [38].

1.5 Challenges of Legged Locomotion

A motion generation technique which can allow the robot to perform multiple tasks, with the single ‘necessary’ stability condition has not been proposed so far. One of the major

challenges for legged locomotion is to formulate a technique through which a complete solution can be given for any desired motion of the robot, not just walking.

Practically this means that a single pattern generator should be able to solve the actuation sequence of a robot for a diverse range of motions, from walking to running to climbing stairs and so on. So far the approach has been primarily to generate a pattern for different stages of the entire task and execute them individually one after another. A technique which can give a universal approach for solving all motion generation problems is far more valuable in terms of versatility and that technique will incorporate a necessary condition for robot stability instead of a sufficient condition.

This problem is addressed in this study by reexamining the notion of stability for a robot and by introducing a new criteria of robot recoverability, instead of robot stability.

1.6 Challenges of Under-Actuated Robotics

Under-actuation in robot systems is a problem which lies in the domain of non-linear systems [39-43]. The control systems used for non-linear systems are inherently non-linear hence this is a very wide field in itself. Our concern for under-actuated robotics is for overcoming the non-linear constraints such as saturation, introduced in the robot architecture by either the mechanical design, application under consideration or the choice of actuators used in the robot.

A good example case is the robot Cubli [44-46], developed by ETH Zurich, which uses 3 motors to move a cube shaped robot by lifting it up and balancing it on an edge and then further balancing it on a corner. The robot uses non-linear controllers to control the change in angular momentum of 3 reaction wheels through brushless DC motors, which causes the entire robot to move in 3D space. Naturally a lot of feedback sensors are used for this application as opposed to a simple speed control problem of a DC motor. The following figures 1.4 (a) and 1.4 (b) shows the lifting motion of the Cubli in two steps.

First the robot jumps up from the lying down position to edge-balance, then it jumps further to balance on a corner. All of this motion is performed by 3 reaction wheels inside the cube, and no external forces act on it. This design shows the complexity of under-actuated systems which poses itself as a challenging modeling and control problem.

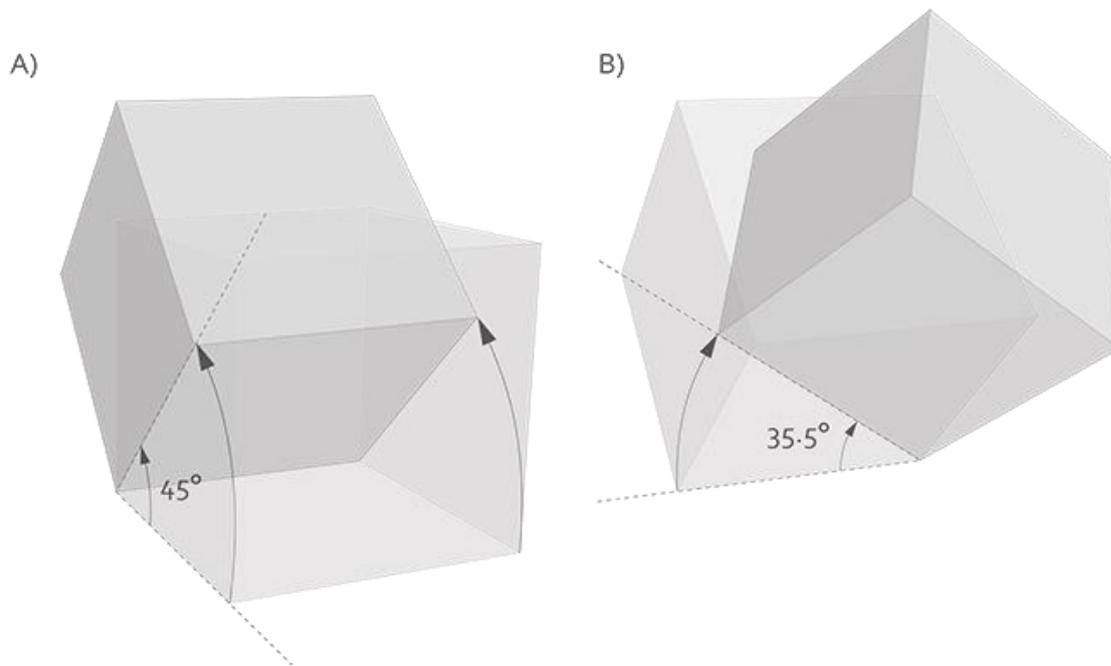


Figure 1.4 Cubli: lifting up and balancing action [100]

Under-Actuated robotics has a lot of challenges, major of which are dynamical modeling and non-linear control. A good technique to generate a reliable motion from an under-actuated system is the use of state flow vectors, which show the change in state of the robot, where all states are mapped onto a n-Dimensional hyperplane, the only problem is the complexity of the solution. Thus in this study a numerical and iterative solution is proposed for solving the motion generation problem for under-actuated robots for 2 pendula and a 12 DOF biped humanoid.

1.7 Under-Actuated Legged Robotics

As mentioned previously, legged robots which can move in 3D space are by definition under-actuated since there is no (actuated) joint which connects the world or ground plane to the robot feet. The way in which natural systems overcome this limitation is by neural network based learning of both modeling and control of the machine. This is also evident from the fact that a child learns to sit first, then roll, crawl, stand and walk in stages during infancy. Even in case of animals the newborn takes time to ‘learn’ its model and control.

The ZMP or COM based CPG description of the robot as a point in space [16], [47-49] reduces the capabilities of the robot by a great amount. A proposed complete solution would incorporate the relative motion of the robot segments as well as the entire robot. Hence the solution proposed in this study essentially divides the robot into its constituent parts which are jointly solved and the motion sequences are executed for each joint. This distribution of the

robot into multiple segments and evaluation of the motion of each of these segments is called multi-body dynamics. The multi-body dynamics approach enables the entire robot to execute a variety of motion sequences which are not possible given a single point description of the entire robot system.

The expansion of the robot from a single point in 3D space to a set of interacting bodies in 6D space of Plücker's vectors resolves the referencing problem as well. All the segments once described in a fixed universal frame outside the robot can make the transformations straightforward. Although this technique is fairly simple to employ, it is computationally expensive and may consume a lot of resources without optimization.

In particular the robot of interest in this study is the robot NUSTBOT-3 [50] under-development at the RISE Laboratories and Research Center, SMME, NUST, Pakistan. The robot has 2 simulation models for full actuation and reduced or under-actuation. Figure 1.5 below shows the side view of the robot hardware currently under development.

The reduced-actuation or under-actuation is from the fact that the robot ankles are un-torqued, both along the pitch and roll axes. This is because one limitation of central pattern generators is the high ankle torques (high stiffness) and another limitation is the loss of solution in case of loss of actuation for a degree of freedom. Both of these limitations are addressed in this study through a decentralized pattern generator which expands the problem of stability of a single representative point in 3D space to a relative and universal recoverability of all the segments of the robot in 6D space.

These 6 dimensions are 3 linear displacements and 3 angular displacements in the universal frame of reference. From rigid-body dynamics [51-54] the equation of motion for a single body can be derived which is expanded over all the segments of the robot through multi-body dynamics. Iteratively solving these set of equations a motion generation sequence can be solved for which gives the actuation sequences for only the actuated joints. This approach is essentially different from preview control based CPG which generates a single solution for the representative point (COM or ZMP) of the entire robot and the inverse kinematics (or inverse dynamics) is used to solve for the joint trajectories.



Figure 1.5 NUSTBOT-3 Side View

1.8 Thesis Overview

This dissertation is divided into 6 chapters, to enhance the readability of the entire document. The distribution is made on the basis of relevance to the discussion and auxiliary information which is not a part of this study is not addressed in much detail, whereas the main focus in each chapter has been given to the subject matter at hand. Following overview details the different parts of this document;

- Chapter 1 is introduction, which lays the foundations of this study and generates motivation for the examined problems. A number of robots, under-actuated systems and applications are discussed and only the key aspects of the robots relevant to this study are highlighted. The problem statements are referred in this dissertation and a few chosen solutions are also discussed. The need for a decentralized pattern generator is informally emphasized based on various settings and applications.
- Chapter 2 builds with the formal background of the study and in-depth review is performed on the chosen titles of literature. The references used are kept concise to facilitate the reading and to associate with consecutive stages of the development. The equations used as starting points for this study are kept minimal to avoid redundancy.
- Chapter 3 carries out the mathematical and dynamical computation of the decentralized pattern generation process. The various components of a DPG are gradually introduced and the mathematical descriptions are written in the iterative numerical format to allow computer generated solutions for all the steps. The test cases of DPG, 5 different dynamical robot systems, are also introduced with mechanical design and target trajectories.
- Chapter 4 deals with the analysis and simulation results for the DPG. Popular techniques of computational multi-body dynamics, rigid-body dynamics and contact and impact dynamics are incorporated. The simulation results, for all 5 cases described in chapter 3, generated in the Simulink SimMechanics environment are compiled. For each of these cases the target trajectories are analyzed by the DPG for robot recoverability and the viable trajectory is executed by the robot. Salient insights are drawn from the simulation results and briefly added prior to collection in the penultimate chapter.
- Chapter 5 compiles the conclusions drawn from the analyses and describes the key advantages and drawbacks of the DPG in comparison to CPG. The topics under consideration are given final words in regards to their utility and applications.
- Chapter 6 records the future work and further possibilities of improving decentralized pattern generation. Among other criterion, the stability criteria and recoverability criteria are detailed and a direction towards a universal criteria for robot motion generation is proposed which incorporates the stability in sense of contact polygon and recoverability in sense of multi-body dynamics.
- Following chapter 6 sources for references are cited which can be used by the reader for further understanding of the concepts collected in this dissertation.

2 Literature Review

This chapter will formally introduce a few of the fundamental concepts of different aspects of legged robotics, under-actuated robotics and multi-body and rigid-body dynamics. The key concepts and their correlation with simpler systems will also be highlighted and gradually an approach for solving the distributed inverse dynamics for multi-body systems is outlined which is effectively the decentralized pattern generation discussed in this dissertation.

2.1 Walking Mechanisms

The development of walking machines particularly biped or humanoid machines has been very active since at least the past 4 decades. A number of complex robots, dynamical systems and their applications have seen the light of day in the recent years [6], [7], [14], [48], [55-59]. Legged robotics is in itself a very diverse field and can be divided into following four major domains based on the type of actuation strategies they incorporate;

2.1.1 Passive Dynamic Walkers

Passive Dynamic walkers as the name implies, use passive higher order dynamics of mechanical systems which make the robot walk on simpler terrains. One particular robot which is shown in the figure 2.1 below is the passive dynamic walker [60] developed through research headed by Steve Collins of Carnegie Mellon University and Andy Ruina at of Cornell University. This robot has extremely high energy efficiency and can walk by utilizing the gravitational potential energy stored by the mechanism itself.

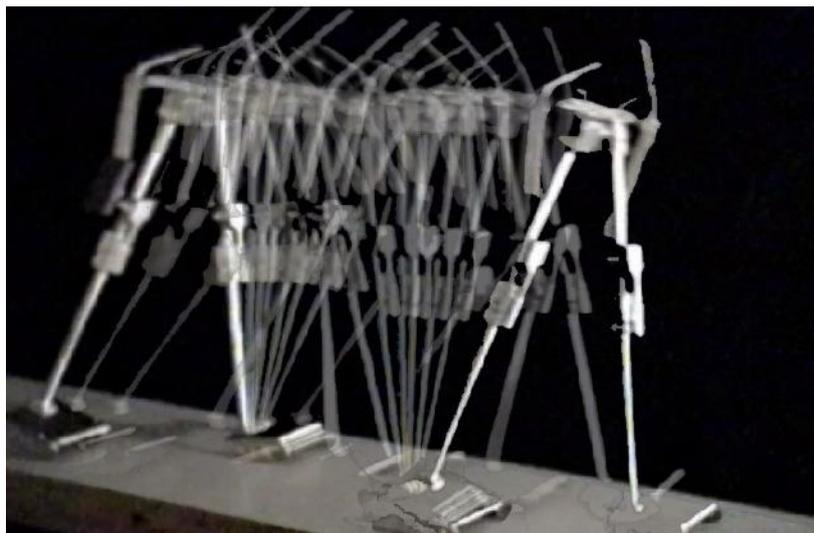


Figure 2.1 Passive Dynamic Walker developed by Collins et.al. [101]

Recent developments in passive dynamic walking for maximizing energy efficiency of walking robots has resulted in fascinating robot systems one such case is the passive biped walker developed at Seno Laboratories at Nagoya Institute of Technology is shown in figure 2.2 below.



Figure 2.2 Passive Dynamic Walker Seno Laboratory, Nagoya Institute of Technology [61]

These passive dynamic walkers are efficient in regards of energy consumption, for instance the passive walker developed by Nagoya Institute of Technology can walk up to 2 hours and 10 minutes on a single push [61]. However despite their high energy efficiency these robots can only perform a single motion or a few closely related motions and hence these are either very special in applications or subjects of advanced research with very little to no applications in human life.

2.1.2 Foot Placement Based Robots

The foot placement based robot typically utilize the dynamical hopping of a single leg or foot in which the end effector trajectory is executed by the robot. Fundamental contributions in the field of hopping machines were done by Leg Lab at MIT by Raibert et.al. [62], [63].

The 3D single legged hopper developed by the Leg Lab at MIT is shown below in figure 2.3 below. The robot used Poincaré mapping [64] to generate motion in 3D space as an open-loop system.

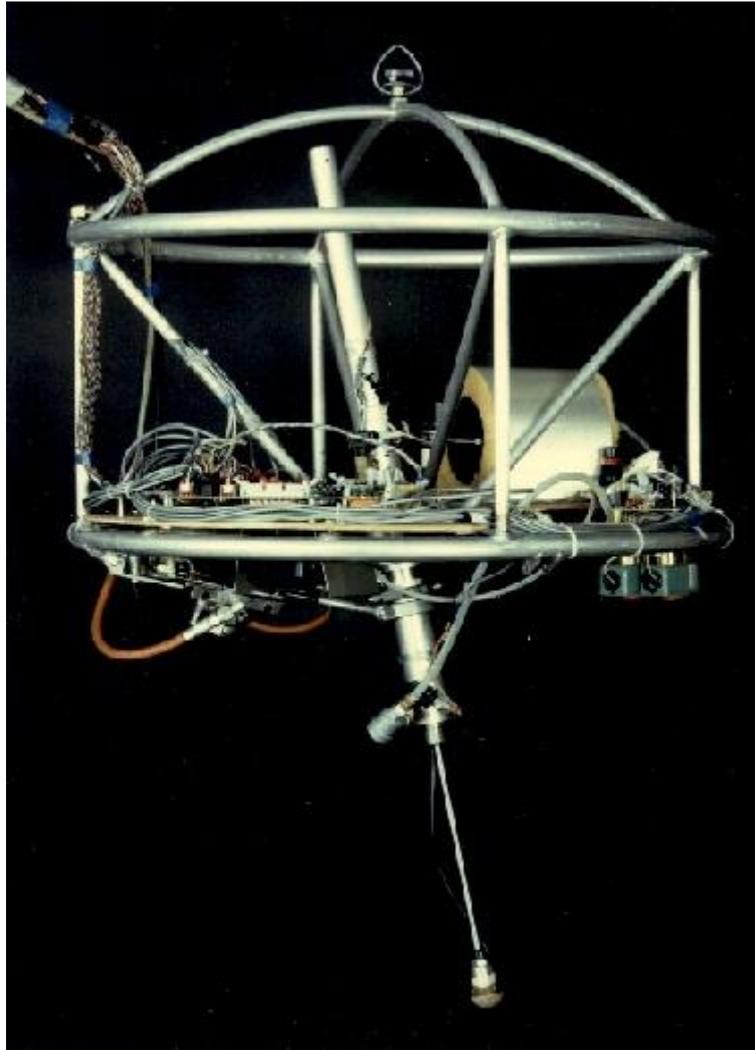


Figure 2.3 Poincaré map based single-legged hopper by Leg Lab, MIT [102]

The Leg Lab at MIT gave rise to Boston Dynamics which has developed a number of robots in recent years which use the foot-placement or end-effector trajectory mapping technique for motion generation. Big Dog, PETMAN and Atlas are three of the defining robots under research at Boston Dynamics for walking over rough terrain and generating walking patterns [65], [66].

The robot Atlas, shown in figure 1.2, was provided by Boston Dynamics to the DARPA Robotics Challenge as the standard robot for the contestants who developed only the algorithms for the specified tasks in the competition and required a hardware to experimentally test the performance of their algorithms. Boston Dynamics has provided a plethora of robots and dynamical systems to DAPRA in recent years, which vary from uneven and slippery terrain traversing robots to exoskeletons for human operators which reduce the mechanical effort. Honda has also developed a solution for the aged individuals to assist in their mobility [67].

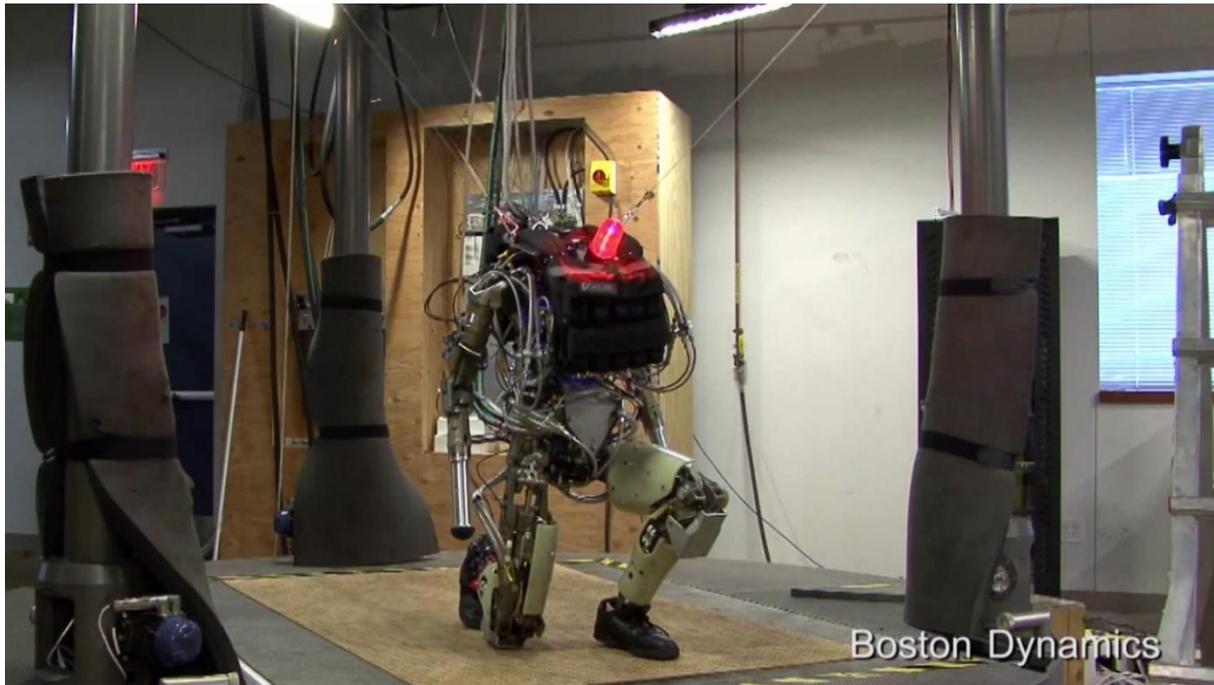


Figure 2.4 PETMAN developed by Boston Dynamics [103]

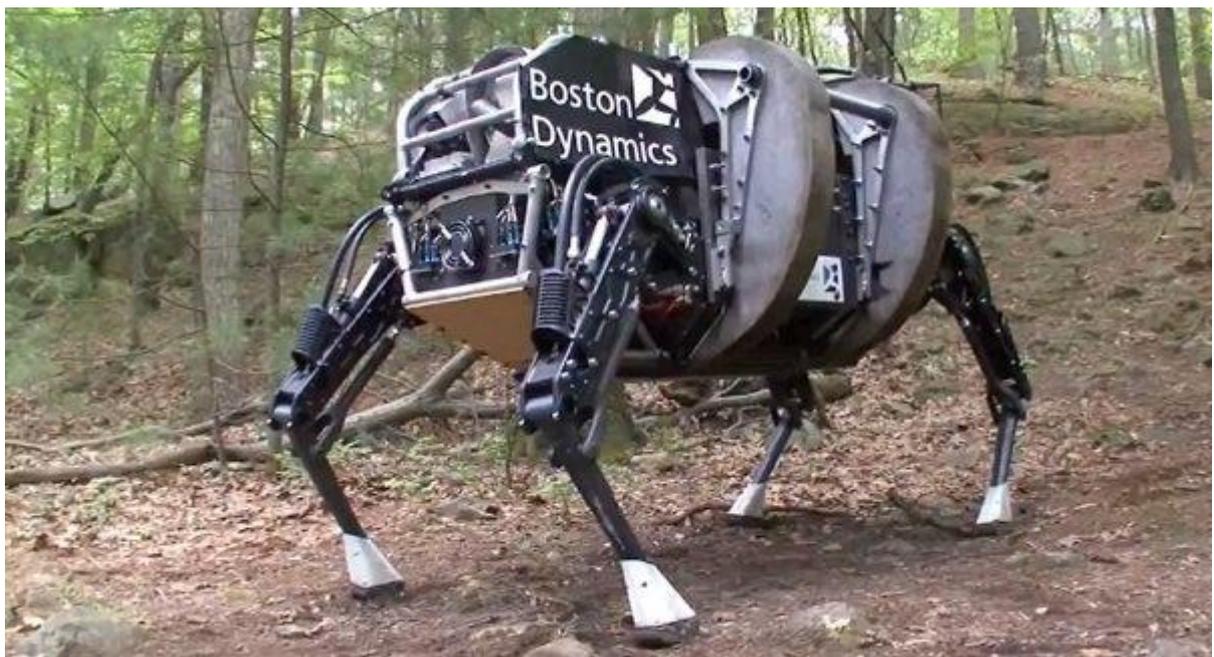


Figure 2.5 Boston Dynamics BigDog walking on natural uneven terrains [104]

Fundamentally the single legged hopping technique is employed by these robots to perform in phase or shifted phase motion which generates the overall motion of the robot. These systems have foot-placement techniques which solve the terrain traversing problem but the complex motion of the robot are not treated individually. These solutions are extended from the single legged hopper and use similar foot-placement algorithms for legged locomotion.

2.1.3 Hybrid Zero Dynamics Based Systems

The Hybrid Zero Dynamics Systems are at a cross over between passive dynamic walkers and dynamical control. The robots which use Hybrid Zero Dynamics essentially replicate the natural motion (varying stiffness) for legged machines by applying controller which change the overall dynamics of the system to match a simplified mass spring or rotational inertial system.

The robots MABEL [68] and Atrias [69] utilize the feedback controllers to map the dynamics of the robot onto a phase diagram which is similar to that of passive dynamic systems. This ensures that the robots get a trade-off between the two approaches, high efficiency from passive dynamic walkers and relatively higher range of mobility as in the case of foot placement robots.

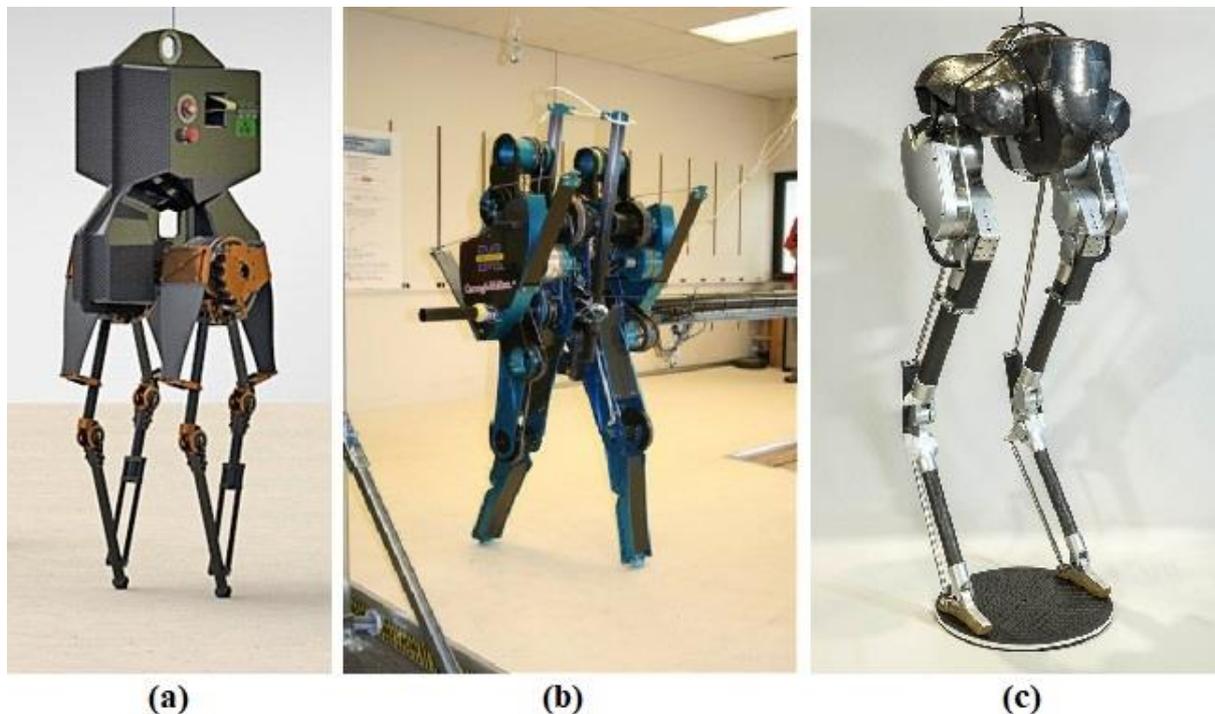


Figure 2.6 Hybrid Zero Dynamics based walking robots [105], [106], [107]
(a) Atrias by OSU, (b) MABEL by University of Michigan, (c) Cassie by University of Michigan and Agility Robotics

The hybrid zero dynamics approach is also used by the robot CASSIE, developed by Michigan University, which is a very agile, high speed biped robot. The dynamical feedback control used by the robot to generate a dynamic gait similar to an ostrich replicates its dynamics in the overall dynamics of the robot.

2.1.4 Central Pattern Generators Based Robots

The robots which utilize central pattern generators for their motion generation have a different approach for solving the walking problem for legged robots. These robots employ high gain feedback control (high stiffness control) for actuators which effectively overrides the system dynamics to introduce stiffness.

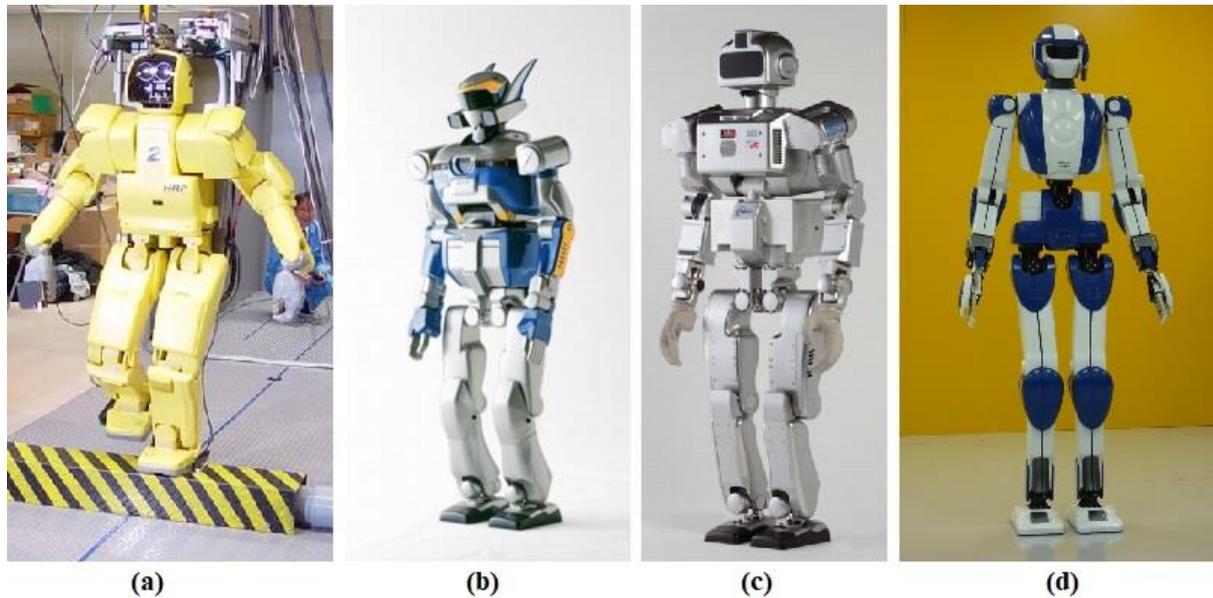


Figure 2.7 Humanoid Robot Projects (HRP) [108]
(a) HRP-1S, (b) HRP-2, (c) HRP-3P, (d) HRP-4

Honda's ASIMO and the Humanoid Research Project (HRP) [70-72] are two of the most popular robots which use CPG for their motion generation. As opposed to dynamical systems or foot trajectory generation systems these robots can perform very diverse range of motions with the cost of high energy. Particularly of importance is the high torque required at the ankle of the robot to balance the entire robot around it. Needless to say in the human body during human gait the ankles play a very important role of force / torque transmission, however they are relatively unpowered as compared to the powerful joints of hips and knees. In humans the major forces generated during most of the motions concerning legs are generated by powerful muscles surrounding the hip and knee joints. The ankles in human beings are used for a smaller angle of attack which enhances force transmission from the ground impact to the limbs [73].

In all of these approaches the techniques employed resolve the actuation sequence problem into either a dynamical response of the entire system, the trajectory mapping of the end-effector (foot) of the robot or through high joint stiffness control which overrides the

dynamics of the system. The solution approach proposed in this study comprises of a solution which resolves the motion generation problem into a multi-body and rigid-body dynamics problem, by dividing first the robot into its constituent parts (segments or links) and further expanding those segments into spatial mass distributions with linear and angular displacements along 6 axis. This approach is inherently different from the previously mentioned topologies because it treats legged robots as a system of multi-body dynamics instead of a combined entity with singular point of control.

2.2 Dynamics of Legged Locomotion

Legged locomotion has been described in great detail for multiple robot systems with insights drawn from the fields of biomechanics and applied robotics [73-75]. The dynamics of legged locomotion is generally described as interplay of different legs in different states. For Biped locomotion the stance and swing phases of the legs are generally defined as shown in figure 2.8 below;

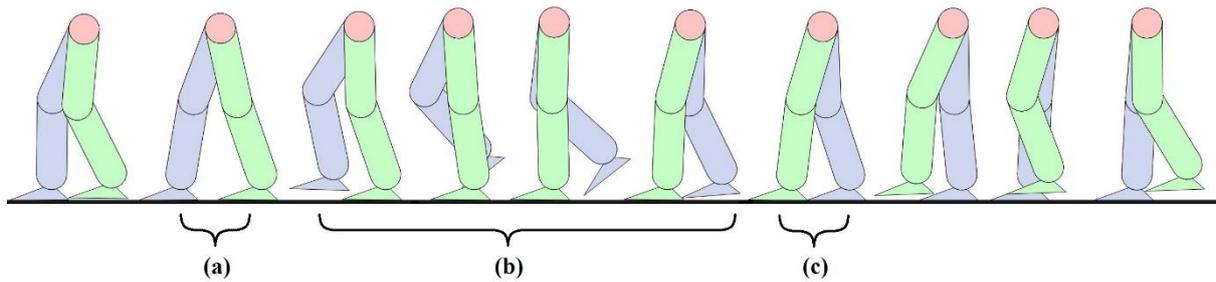


Figure 2.8 Phases of Biped Walking

- (a) Double Support Phase, (b) Stance Phase left leg (Green), Swing Phase right leg (Blue)
(c) Double Support Phase

During the gait cycle, for a biped, both legs cycle between the swing and stance phases with double support phases interspersed between the transitions of the legs. During stance phase the friction between the ground and the supporting foot ensures the robot (or human) has no slippage. During stance phase the supporting leg bears the dynamic weight of the entire structure, and the reason why central pattern generators have high torque requirement for the ankle (pitch) motors is that the entire weight of the robot is counterbalanced about the ankle pitch axis.

There is another way to visualize the same gait cycle through multi-body dynamics in which the robot segments are independently assigned their motion primitives and instead of considering the robot as a manipulator it is defined as a set of bodies with force/torque interactions at the joints. The spatial trajectories for the left thigh and right shin, used by a DPG

is shown for 3 via poses in the figure 2.9. In order to avoid complications spatial trajectories for only 2 segments are shown here, whereas the DPG solves for the spatial trajectories of all the segments of the robot simultaneously.

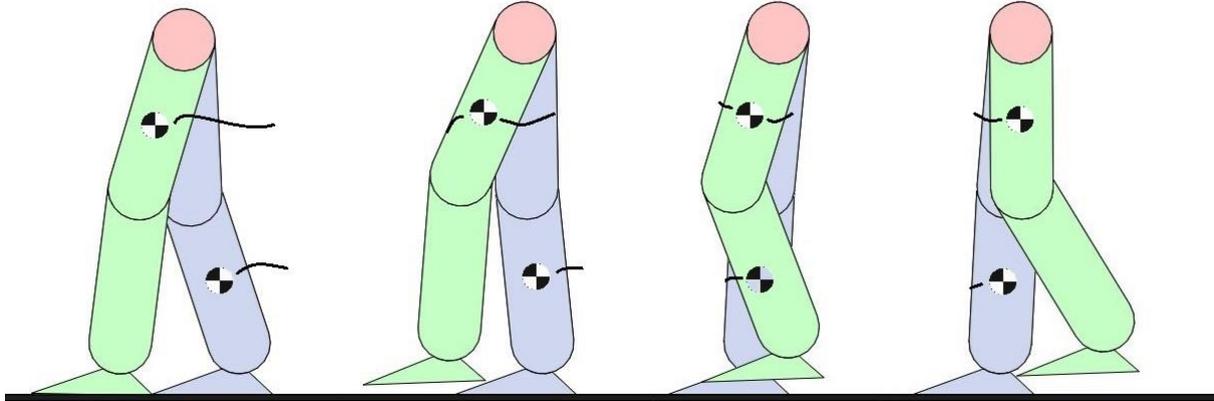


Figure 2.9 MBD based Spatial COM Trajectories used by DPG

The multi-body dynamics of the walking cycle utilized by DPG is not based on the support / stance / swing phases, instead it is simply based on the interactions between multiple links (or segments or rigid bodies) in space. The notion of balance and gait stability in this application take on different interpretations, similar to those used by flying robots, instead of legged walkers or manipulators.

2.3 Balance and Gait Stability

The balance of biped robots [1] is defined as the condition for keeping the robot upright, whereas the stability for central pattern generators [16], [76], [77] is defined as the inclusion of the ZMP within the contact polygon, on the ground plane as a sufficient stability criteria [38] and not a necessary one. The ZMP [15], [48] is defined as;

$$\hat{p}_{ZMP} = \left[\begin{array}{c} \frac{\iint_{x,y=-\infty}^{x,y=+\infty} \hat{p}_{x,y} \vec{f}_{norm}(x,y) dx dy}{\iint_{x,y=-\infty}^{x,y=+\infty} \vec{f}_{norm}(x,y) dx dy} \\ 0 \end{array} \right]^T \quad (2.1)$$

The first two terms, x and y coordinates, of the ZMP position vector in the above equation are the calculated by normalizing the position vector with force applied over the entire surface of the contact foot. The set of all points within the contact polygon \bar{P}_G on the ground plane is defined for the supporting foot, and the normal force (ground reaction force) at the coordinates (x, y) is represented by $\vec{f}_{norm}(x, y)$.

The robot is stable, when the sufficient condition is satisfied;

$$\hat{p}_{ZMP} \in \bar{P}_G$$

For computational purposes the discrete version of the ZMP equation is defined as;

$$\hat{p}_{ZMP} = \left[\begin{array}{c} \frac{\sum_{\forall x} \sum_{\forall y} \hat{p}_{x,y} \vec{f}_{norm}(x,y)}{\sum_{\forall x} \sum_{\forall y} \vec{f}_{norm}(x,y)} \\ 0 \end{array} \right]^T \quad (2.2)$$

The reason only the normal force or z component of the force vector is treated in this equation is that effectively at the Zero-Moment-Point, all the planar torques are canceled out by the robot. This means at the zero moment point;

$$\tau_x = \tau_y = 0$$

Hence the robot cannot fall over while executing the motion.

In most of the applications the ground reaction force is not directly deducible prior to motion execution, hence in order to check the robot for stability preemptively the following equation can be used to compute the ZMP;

$$\hat{p}_{ZMP} = \left[\begin{array}{c} p_{c,x} - \frac{p_{c,z}}{g} \ddot{p}_{c,x} \\ p_{c,y} - \frac{p_{c,z}}{g} \ddot{p}_{c,y} \\ 0 \end{array} \right]^T \quad (2.3)$$

The position vector \hat{p}_c represents the position of the center of mass from the frame of reference and g is the gravitational acceleration;

$$\hat{p}_c = [p_{c,x} \quad p_{c,y} \quad p_{c,z}]^T$$

The reduced equation of the ZMP position vector describes the ZMP with as a function of the position and accelerations of the COM of the robot only. Computationally this is very straightforward as the ZMP computed from this equation can be solved by using the forward kinematics of the entire robot.

For a decentralized pattern generator the robot stability cannot be described, however the robot recoverability is valuable in the sense that it is a qualitative measure and a Boolean check of the capability of the robot to restore its position. This means the DPG checks not the stability of the entire robot, instead it computes the ability of the robot to restore a defined configuration, and if that configuration is stable, then the robot can restore itself to that stable configuration. The notion of robot recoverability will be expanded on further in the consequent chapters.

2.4 Dynamical Control of Motion

Control Systems for legged robots understandably employ some of the most ingenious solutions for dynamical systems. For legged robots in general and biped robots in particular the preview control [8], [37], [78-80] technique in order to control the motion extrapolates the ZMP trajectory prior to motion execution and curve fits it in order to maintain the robot stability. For discrete systems the preview control can be understood from the following nonlinear equations [41], [43], [81];

$$X[k + 1] = A(X[k], U[k]) \quad (2.4)$$

$$\hat{p}_{zmp} = C(X[k]) \quad (2.5)$$

Here; $X[k + 1]$ is the updated state, $X[k]$ is the current state, $U[k]$ is the current input, and $A()$ is the state update function. The nonlinear transformations are not homogeneous as in the case of linear systems. The state update can drive the ZMP to new positions based on the input function, thereby modifying the trajectory prior to execution.

For multi-body systems, such as the description used by DPG the robot segments can be controlled dynamically instead of plug-in type controllers, and hence the system equation changes the form to a rigid body equation of motion [29], [82], [83];

$$H(X[k])X[k + 2] + C(X[k], X[k + 1])X[k + 1] + G(X[k]) = B(X[k])U[k] \quad (2.6)$$

Here the functions $H(X[k])$, $C(X[k], X[k + 1])$, $G(X[k])$ and $B(X[k])$ represent the inertial, Coriolis, gravitational and state control functions. For an invertible system the optimal control function is;

$$U[k] = f_c(H(X[k]), C(X[k], X[k + 1]), G(X[k])) \quad (2.7)$$

As the equation implies it is both inconvenient and infeasible to derive the analytical $U[k]$ for systems with more than two body segments, thus a numerical form which is iterative can be used from the nonlinear equation of to generate the optimal control law which satisfies the constraints as well as the dynamics of the system.

This study does not include the optimal control system derivation, it is assumed for the simulations that an optimal control system executes the target trajectories, an online system must include an online control law, along with an online pattern generator. The computation for the control law can be reduced by unfolding onto a parallel processing hardware, instead of an on-board microprocessor, however that is beyond the scope of this study.

2.5 Rigid Body Dynamics

A rigid body is an incompressible body i.e. the mass distribution of a rigid body is constant however it does not necessarily have a uniform density. For a rigid body in space, the 6D equation of motion [51], [84] is given in the form of Plücker's vectors as;

$$\hat{f} = \frac{d}{dt}(\bar{I}\hat{v}) = \bar{I}\hat{a} + v \times^* \bar{I}\hat{v} \quad (2.8)$$

The force vector f is the combined force/torque acting on the body, with mass distribution and the parallel axis theorem transformation represented by \bar{I} . The motion vectors \hat{v} and \hat{a} represent the velocity and acceleration of the rigid body in 6D space. Square matrix form of the cross-product is represented by the skew symmetric matrix \tilde{v} as following;

$$\tilde{v} = \hat{v} \times^*$$

The figure 2.10 below shows the 3D spatial velocity vectors v and ω , the 3D spatial force vector f and the 3D spatial torque vector τ ;

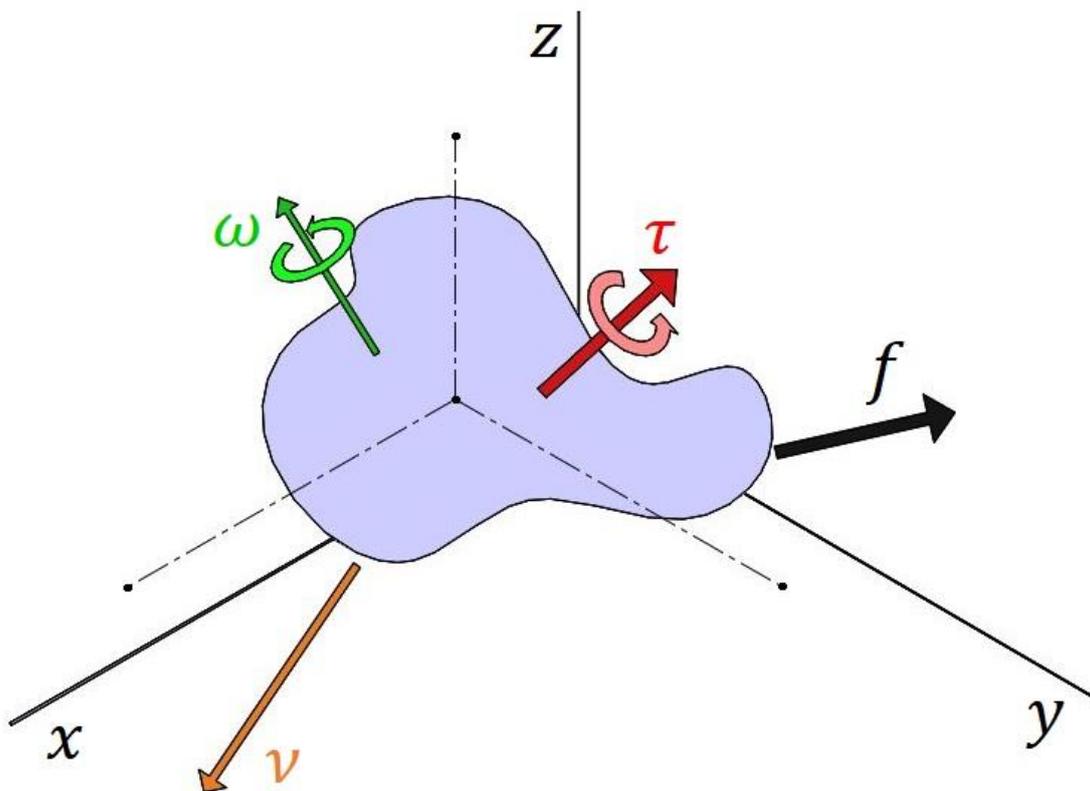


Figure 2.10 Spatial Vectors for a rigid body in 3D space

The 3D spatial vectors are related to the 6D vectors as following;

$$\hat{v} = \begin{bmatrix} v \\ \omega \end{bmatrix} \quad (2.9)$$

$$\hat{f} = \begin{bmatrix} f \\ \tau \end{bmatrix} \quad (2.10)$$

The Plücker's 6D spatial vectors are compounded vectors of the linear and angular motion and force vectors in 3D space.

2.6 Multi-Body Dynamics

In considerations to multi-body dynamics, the interactions of multiple bodies are represented by [54], [85] the interplay for force vectors for all the segments of a robot. The Newton-Euler Equations [86] or the Lagrange's Equations [87] give the following form of the generalized multi-body dynamics equation in the 6D Plücker's vectors notation as;

$$M(\hat{p})\hat{a} + C(\hat{p}, \hat{v})\hat{v} + G(\hat{p}) = \sum \hat{f} \quad (2.11)$$

Here $M(\hat{p})$ is the inertial term, $C(\hat{p}, \hat{v})$ is the Coriolis term and $G(\hat{p})$ is the gravitational term, and the net force acting on the body is described as;

$$\sum \hat{f} = \hat{f}_c + \hat{f}_{ext}$$

The actuators apply external forces \hat{f}_{ext} and the constraint forces can be given by the Lagrange's equations as;

$$\hat{f}_c = J(\hat{p})\lambda$$

In the above equation J is the Jacobian transformation and λ is the Lagrange multiplier. These are continuous time equations for a system of multiple bodies, for practical applications the computation methodologies are used in robotic systems [88].

2.7 Under-Actuated Robotics

An under-actuated system is one in which instantaneous acceleration cannot be generated along all the axis representing degrees of freedom [29]. For an under-actuated multi-body system of rigid bodies the forces of interaction between each body are used as in equations of motion, these forces comprise of contact forces, constraint forces, gravitational forces and the actuator forces.

The inverse dynamics of under-actuated multi-body systems [89-91] give the combined equations of actuator forces acting on all the rigid segments in the following form;

$$\begin{aligned}
f_A^1 &= T_1^U(f_1^1) + T_2^U(f_2^1) + \dots + T_n^U(f_n^1) \\
f_A^2 &= T_1^U(f_1^2) + T_2^U(f_2^2) + \dots + T_n^U(f_n^2) \\
&\vdots \\
f_A^n &= T_1^U(f_1^n) + T_2^U(f_2^n) + \dots + T_n^U(f_n^n)
\end{aligned} \tag{2.12}$$

Here the left hand side of each equation expresses the force acting on the link n due to all the actuators. Given the values of gravitational, contact [87], [92-94] and Coriolis components, numerically solving these equations can give the convergent values of the actuator forces f_j^i , force on link i due to actuator j . The transformation functions T_i^U from link i to the universal frame of reference are non-linear and have a defined format in the 6D Plücker's notation.

After the numerical convergence of all the actuator forces the final force by each actuator acting on the complete rigid-multi-body;

$$f_{Act-n} = T_U^n(\bar{P}, \sum_{\forall i} f_n^i)$$

This inverse transformation from the universal frame of reference U to the joint frame n is a function of all the individual joint force contributions to all rigid-body segments and the position matrix (state matrix) of the entire robot \bar{P} .

2.8 Pattern Generator as a Multi-Body Dynamics Problem

Finally we from the above topological derivations we can formulate the pattern generation (or more generally the motion generation) problem as a multi-body inverse dynamics problem and for our case of rigid body assumption for all segments of the robot, the pattern generation is carried out by the decentralized pattern generation. Furthermore the in case of reduced actuation (under-actuation) the CPG cannot generate solutions, hence the expanded multi-rigid-body system with under-actuation is solved by the DPG.

The complete pattern generation (motion generation) process for the multi-rigid-body system with under-actuation is explained in detail in the following chapter, which constructs the mathematical and sequential components of the DPG with example cases for different types of robots with different topologies.

3 Decentralized Pattern Generation

This chapter discusses a multi-body mechanics based description of manipulators and the ensuing mathematical derivation of a Decentralized Pattern Generator will be carried out. Five different cases for the motion generation of different robot systems will be discussed, starting from the inverted pendulum, the double pendulum and finally a 12 DOF under-actuated humanoid robot.

3.1 Multi-body Description of Robot Systems

A legged robot can be understood as a mobile parallel manipulator, for a biped robot the number of serial manipulator chain in parallel is two and each chain has a foot as the end-effector. Each parallel chain consists of rigid body segments which move relative to each other to generate a motion pattern. The relative motion of each segment is generated in a manner to ensure the stability of the robot.

A decentralized pattern generator divides the pattern generation of a complete robot into the dynamic motion generation of multiple bodies in 3D space. The different components of a decentralized pattern generator are briefly introduced in the following sub-sections and formally derived in section 3.2.

3.1.1 Assumptions

The following assumptions are taken for the decentralized pattern generation solutions;

- 1- High ground friction and no slippage between the feet and the ground floor.
- 2- No backlash in the actuators or transmission mechanisms.
- 3- Lossless rigid body mechanics applicable for all robot segments (links).
- 4- The ground surface is smooth / even.
- 5- The contacts between bodies are elastic (conservation of momentum).

3.1.2 Motion Primitives and Motion Constraints

The robot motion generation cases discussed in this dissertation consist of revolute joints only. The relative motion for different robot segments (links) is constrained by the mechanical joint limits for angular displacements θ . The DPG design caters for the actuators of the robot as well hence the joint angular velocity ω and angular acceleration α is constrained by the actuator of choice.

The relative motion of bodies at joints is also constrained by the number of degrees of freedom of the joint. A revolute joint with 1 DOF allows only the relative rotational motion of the adjacent robot segments about a single axis i.e. the joint axis. The remaining 5 DOF of the joint are constrained and no relative linear or angular motion can occur along or about these remaining axes. The effect of these constraints is shown in the figure below, the only difference between the forces acting on adjacent links (segments) is the force/torque introduced by the joint between the two links and the respective force/torque introduced by the gravity, the remaining forces/torques (constrained Degrees of Freedom) are transmitted through the joint.

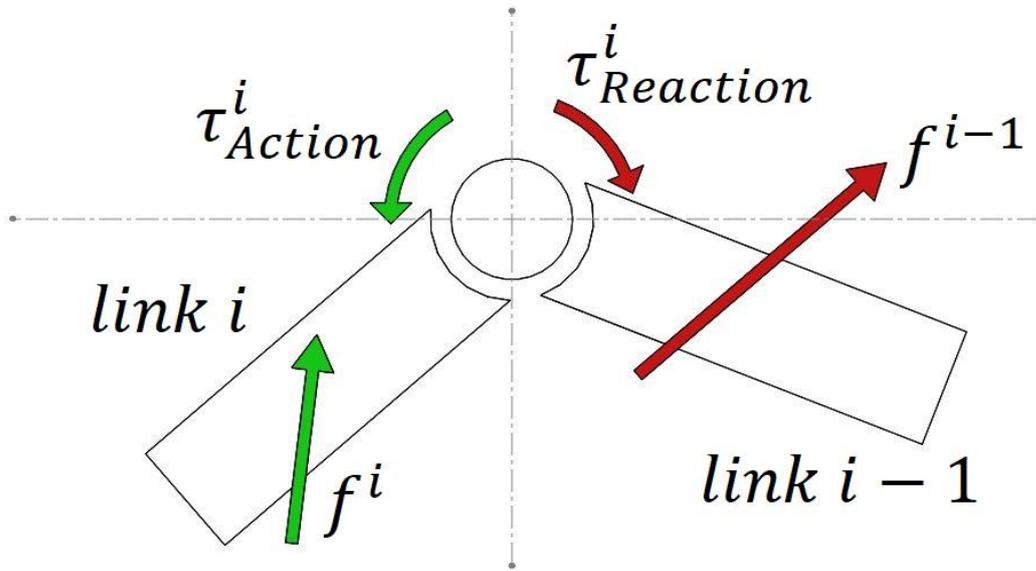


Figure 3.1 Force Transmission through the joint i between adjacent links $i - 1$ and i .

The net 6D force acting on a link is described as the sum of actuator force, force due to gravity and the force propagation due to constraints. The force propagation through the revolute joints between the robot links as described below;

The net force acting on link i is denoted by f^i ;

$$f^i = f_G^i + f_A^i + f_C^i \quad (3.1)$$

Here, f_G^i is the 6D Plücker's force vector due to gravity, f_A^i is the 6D Plücker's actuator force vector and f_C^i is the 6D Plücker's force vector due to constraints. The joint constraints contribute towards the total constraints of the robot, and the force propagation due to link to link transmission is part of the constraint force acting on the link.

For the case shown in figure 3.1 we can write the constraints forces acting on the link i as;

$$f_C^i = P_{C,i-1}^i f^{i-1} + P_{C,i+1}^i f^{i+1} \quad (3.2)$$

Here $P_{C,i-1}^i$ is a row of the position- constraint matrix P_C of link i , this row corresponds to the force applied from link $i - 1$. Given the pose (configuration) matrix of the robot \bar{P} . The complete pose collisions and constraint matrix P_C^i has the following structure;

$$P_C^i(\bar{P}) = \begin{bmatrix} c_{p1,1}^i(\bar{P}) & \dots & c_{p1,6}^i(\bar{P}) \\ \vdots & \ddots & \vdots \\ c_{pn,1}^i(\bar{P}) & \dots & c_{pn,6}^i(\bar{P}) \end{bmatrix} \quad (3.3)$$

Here $c_{pa,b}^i$ is the positional constraint function of the link i with respect to link a for the motion along b axis. Here the link is represented by a , where

$$a \in \{1,2, \dots, n\}$$

Here n is the total number of links of the robot, and b is the axis along or about which the constraint function is being evaluated.

$$b \in \{1,2, \dots, 6\}$$

These are the 6 members of the 6D Plücker's position vector representing the linear displacement from origin along the x, y and z axes, and angular displacement (orientation) about the x, y and z axes, respectively.

$$\bar{p} = [p_x \quad p_y \quad p_z \quad \theta_x \quad \theta_y \quad \theta_z]^T$$

The matrix \bar{P} is the nx6 position or configuration matrix information of all the links of the complete robot, and the corresponding velocity matrix \bar{V} captures the time derivative of \bar{P} .

$$\bar{P} = \begin{bmatrix} p_x^1 & p_y^1 & p_z^1 & \theta_x^1 & \theta_y^1 & \theta_z^1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ p_x^n & p_y^n & p_z^n & \theta_x^n & \theta_y^n & \theta_z^n \end{bmatrix} \quad (3.4)$$

The corresponding velocity Matrix for the robot is;

$$\bar{V} = \frac{\partial}{\partial t} \bar{P}$$

The constraint matrix solves for the motion of the robot segments in space too, and the environment contacts, which are not due to collision between robot parts, are represented in the ith row for the ith link.

This means for a freely floating robot consisting of only one link, the position/configuration constraint matrix is a row matrix with 6 values which describe the environmental collisions of the robot as constraints.

For a flying robot such as a helicopter or a quadcopter the equation for \bar{P} is a simple 3D space collision detection matrix of the form;

$$\bar{P}_{C\text{Quad}} = [c_x \quad c_y \quad c_z \quad c_\alpha \quad c_\beta \quad c_\gamma]$$

And the angles α, β and γ represent the pitch, roll and yaw angles of the flying robot.

The velocity constraints of the robot are dependent upon the choice of actuators, primarily the limiting velocity of the actuators.

$$V_C^i(\bar{P}, \bar{V}) = \begin{bmatrix} c_{v1,1}^i(\bar{P}, \bar{V}) & \dots & c_{v1,6}^i(\bar{P}, \bar{V}) \\ \vdots & \ddots & \vdots \\ c_{vn,1}^i(\bar{P}, \bar{V}) & \dots & c_{vn,6}^i(\bar{P}, \bar{V}) \end{bmatrix} \quad (3.5)$$

Unlike the position constraint of the robot, the velocity constraint of the entire robot is a function of both pose/configuration matrix and the velocity matrix. The function $c_{va,b}^i$ solves the velocity constraint of the robot for the i th link. The velocity constraint of the links are dependent upon the relative position as well as the relative motion of the links. An interesting point to note in this case is the instantaneous velocity, along with the direction of motion with the pose matrix constrains the actual acceleration of the robot as well. This need not be solved for, but the idea is same, the acceleration of robot links is limited by the constraints on velocity.

It is also worth noting here that although the position / configuration constraints and the velocity constraints are intuitive they are not related through integration or differentiation. These constraints are non-holonomic. The fundamental reason being that the constraints defined by the position / configuration constraint matrix arise from the physical construction of the robot, whereas the velocity constraints are caused by the choice of actuators as well as the configuration of the robot. The constraint value of 0 denotes the lack of mobility in both cases and a constraint value of 1 signifies the allowed mobility of the robot segment (link) in the particular axis.

If a robot contains prismatic joints instead of revolute joints, the angular constraints are replaced by linear constraints depending upon the mechanism of the prismatic joint and the linear actuator of choice. The dimensions of the constraint matrices remain the same in either case.

3.1.3 Contact Model

The ground surface is assumed to be planar hence the simplest multibody contact model [87], [92], [94], [95] is used for the simulations and can be approximated by a flat plane with $z = 0$ for all x and y locations;

$$p_{c,z}^i = \begin{cases} 0 & z \leq 0 \\ 1 & \text{otherwise} \end{cases}$$

The contact model is assumed to be elastic, hence the linear and angular momenta before and after the collisions are conserved.

The robot segments (links) cannot penetrate the neighboring segments or the ground plane and are assumed to be rigid hence the Newtonian principle of action and reaction is maintained during collisions. The actuators are also assumed to be in elastic contact with the bodies. In other words the actuator force and torque applied on one segment are equal in magnitude and opposite in direction to the reaction force and torque experienced on the adjacent robot segment as highlighted in figure 3.1.

3.1.4 Dynamics of a Rigid Body

The equation of motion of a single rigid body in 3D space is represented using the 6D Plücker's vectors for position, velocity and acceleration as following;

$$f = \frac{d}{dt}(\bar{I}\hat{v}) = \bar{I}\hat{a} + v \times^* \bar{I}\hat{v}$$

In this equation of motion, f is the 6D Plücker's force/torque vector applied to the body, \bar{I} is the 6x6 inertia tensor which incorporates the mass, center of mass and mass distribution of the entire body. The vector \hat{v} is the 6D Plücker's velocity vector and \hat{a} is the 6D Plücker's acceleration vector of the body.

The effect of mass distribution and the Coriolis effect are all captured in the second term on the right hand side of this equation, where $v \times^*$ is the conversion of velocity vector \hat{v} to a 6x6 skew symmetric matrix for cross product multiplication, this can be rewritten as \tilde{v} .

The notation \hat{p} is a vector with respect to the universal frame of reference and p is a column matrix consisting of the values of the vector. The notation with capital letters \bar{P} is used for matrices with more than one rows, which are composed of transposed column matrices.

3.1.5 Linear & Angular Momenta of a Rigid Body

The combined mass of a rigid body m , center of mass p_c and mass distribution of a rigid body in 3D space I is combined into a 6x6 mass-inertia tensor \bar{I} , such that;

$$\bar{I} = \begin{bmatrix} I - m(p_c \times^*)(p_c \times^*) & mp_c \times^* \\ m(p_c \times^*)^T & ml_3 \end{bmatrix} = \begin{bmatrix} I - m\tilde{p}_c\tilde{p}_c & m\tilde{p}_c \\ m\tilde{p}_c^T & ml_3 \end{bmatrix} \quad (3.6)$$

In order to avoid confusion between the two, the 3D inertia tensor is expressed as I , whereas the 3x3 identity matrix is written as I_3 . Notice the parallel axis theorem is incorporated in the first 3x3 terms and hence the center of mass p_c is the 3x1 position matrix of the center of mass and I is the 3x3 inertia matrix at the center of mass.

The 6D force/torque vector applied on a rigid body is equal to the time derivative of the 6D momenta vector. For the 6D Plücker's vector of a rigid body the following generalized equation can be used as an equation of motion;

$$f = \frac{d}{dt}(\bar{I}\hat{v})$$

The inverse of this equation also holds true and can be used to solve for the combined momentum of the rigid body;

$$H(\hat{p}, \hat{v}) = \int_0^t f(\tau)d\tau = \bar{I}\hat{v} + H(\hat{p}_0, \hat{v}_0) \quad (3.7)$$

The combined momentum of the rigid body during spatial motion is a function of mass distribution, current instantaneous velocity, initial position and initial velocity.

We rewrite the equation for forces which is the input for the dynamics equation of motion;

$$\Sigma f^i = f^i = f_A^i + f_G^i + f_C^i \quad (3.8)$$

For a humanoid robot with multiple links (segments) the presence of these forces takes different meanings / interpretations. The elements on the right side of this equation are respectively sum of actuator forces, gravitational force and contact or constraint force. The force due to gravity is at all times vertically downward and the contact or constraint force occurs between different segments or between the feet of the robot and the ground plane. The contact force is universal for all multibody systems and gives rise to constraining forces whereas the actuator force causes the relative motion of the robot segments.

The actuator force in equation 3.8 is the force generated by the force/torque controller. A hybrid position and force controller, dependent on the pose and velocity matrices of the robot can be developed to ensure the trajectory of the link, but that is a topic left for a future study.

3.1.6 Stable Pose

The stable pose of a robot is the pose at which the robot can be, theoretically, powered off and it will maintain that pose provided no external forces or torques act on it. Practically however it is the dynamic equilibrium pose of the entire robot. For a fully-actuated robot this pose is also described as the static stability pose. For an under-actuated robot this pose is the dynamic balance pose, also referred to as the unstable fixed point, but for the sake of differentiation the word pose is used instead of point.

The stable pose is of practical utility as well. This is the pose at which the robot requires considerably low energy to maintain its pose. The energy required to perform any work from the pose or around the stable pose is also significantly lower. For instance the gait pattern of an under-actuated humanoid robot designed around the dynamic balance pose (stable pose) is an efficient gait pattern. This is also the reason why for a human the stable pose is the standing pose which requires least energy to maintain while walking.

A robot system can have multiple stable poses and the choice is dependent upon the application. For a pendulum the stable pose can be defined as vertically downward or vertically upward, depending upon the type of action desired from the robot.

3.1.7 Pose Recoverability Margin

A DPG relies on the pose recoverability of the entire robot, which is a measure of the extent from which a robot can recover itself to a stable pose. This is a function of the current position and velocity of all the robot segments and depends upon the position constraints and velocity constraints of the robot.

The pose recoverability for a single link is given by;

$$r_p^i = f_{i,\bar{P},\bar{V}}(\bar{P}, \bar{V}) \quad (3.9)$$

To estimate the pose recoverability of the robot, the computational non-linear dynamics model is used. Although an analytical solution can be generated to solve for the pose recoverability in some cases, it is far more practical, feasible and convenient to generate an iterative pose recoverability margin for robot systems.

In order to demonstrate the pose recoverability of a generic system (under the action of a linear or non-linear controller) the vector field (phase diagram) of the dynamical system is generated. A constrained vector field (phase diagram) which flows away from the stable pose signifies a loss of pose recoverability.

Pose recoverability is iteratively solved and a dynamic margin is established which ensures the robot does not fall over during its motion. A distinction which must be made here is that the pose recoverability margin is not the same as the stability margin of the support polygon in central pattern generators. Whereas the CPG contact polygon is a point based stability metric for the entire robot, the pose recoverability margin is essentially a dynamic measure of the robot's ability to restore its posture from or after the current motion is executed.

As an example case the pose recoverability of a simple pendulum is easily described as the ability of the pendulum to recover to the stable position once it moves out of that position after the execution of the desired motion. In the figure below θ_d is the desired angle of the pendulum (vertically upward) and θ is the current angle, whereas τ_A is the actuator torque applied at the base.

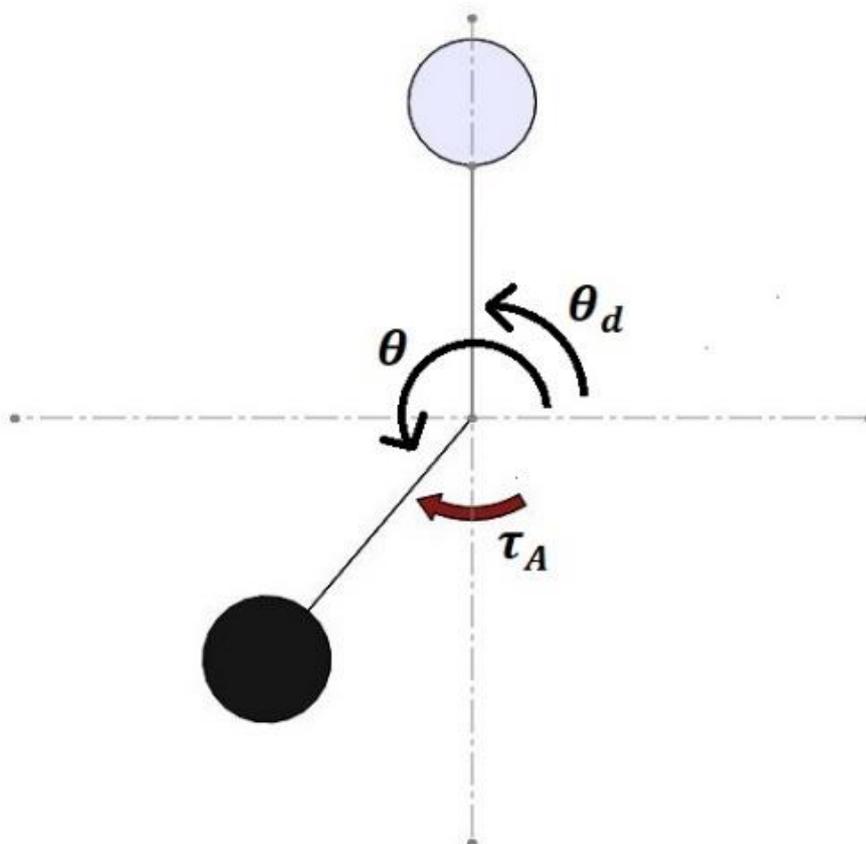


Figure 3.2 A Simple Inverted Pendulum

By applying estimated torque (constrained by the choice of actuator) a motion sequence can be generated which balances the pendulum vertically upward. However if the actuator torque required to bring the inverted pendulum towards the desired position is outside the limits (actuator constraints) then the vector field (phase diagram) flows away from the stable pose and the pose recoverability check fails. The phase diagram of the simple inverted pendulum is shown below in figure 3.3.

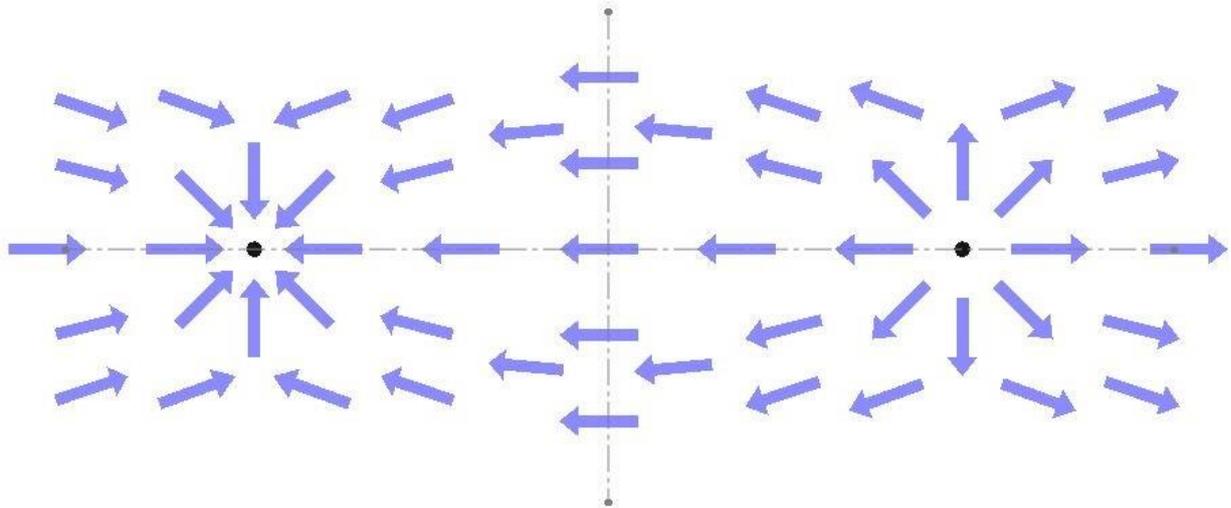


Figure 3.3 Phase Diagram (Vector Field) of a Simple Inverted Pendulum

The pose recoverability of a humanoid robot cannot be described for 6D spatial vectors in a vector field diagram (phase diagram), instead it is numerically computed by solving for the desired trajectory of the complete system.

The pose recoverability is a threshold function which is computed for all the robot segments (links) poses in 6D space. If any of the values corresponding to the pose recoverability returns 0 then executing the trajectory will make the robot fall over with certainty.

In case of the humanoid robot NUSTBOT-3, with un-actuated feet the differentiation between pose recoverability and the ZMP stability is described in the following figure. The green marker represents the position of the Center of Mass and the Red Marker shows the position of the ZMP in the ground plane in all 3 of the cases.

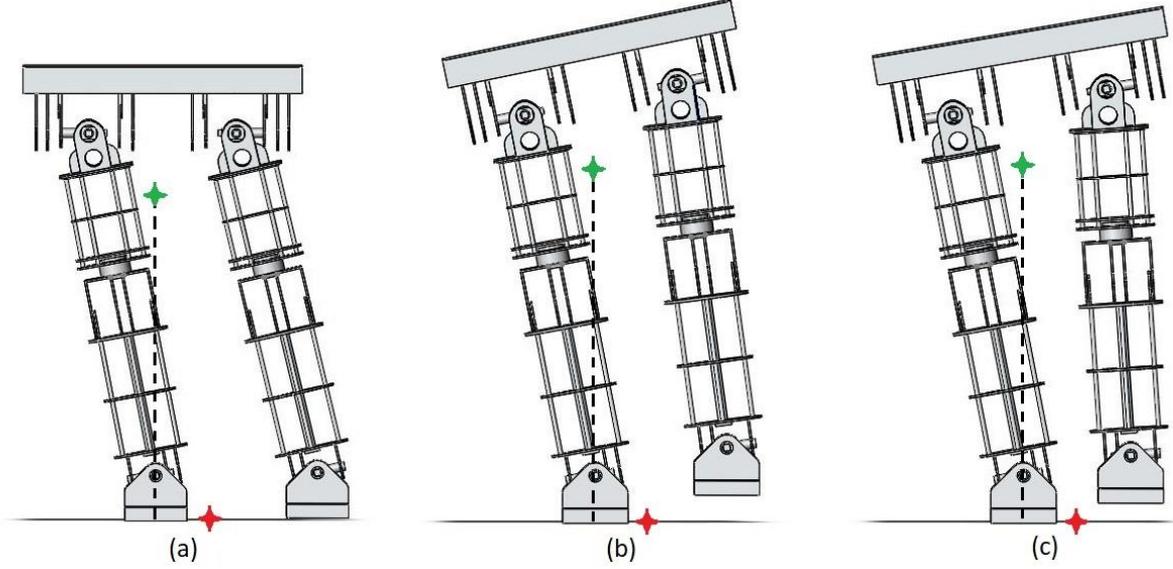


Figure 3.4 Pose Recoverability vs ZMP Stability.

In Figure 3.4 (a) the robot is in the double support phase and hence the ZMP lies within the contact polygon and the robot has fulfilled the ZMP stability check. In Figure 3.4 (b) the robot is in swing phase and the ZMP lies outside the supporting foot, the robot should fall over given the stability criteria is failed, and the observed motion is the same. In Figure 3.4 (c) the robot is again in swing phase and the ZMP lies outside the supporting foot, however in this case a reversed angular momentum applied on the left leg of the robot (the leg in air shown on the right in the figure), generates a dynamic trajectory which does not make the robot fall over.

In this case the ZMP stability criteria as described in [2] ensures the robot stability as long as the ZMP lies within the contact polygon, however the stability criteria cannot give a definitive answer once the ZMP lies outside the support polygon. In the case of Figure 3.4 (b) the ZMP stability criteria is not satisfied, and the pose recoverability check fails for the all the segments of the robot. The pose recoverability check in Figure 3.4 (c) is satisfied for all the links of the robot and hence it continues on the generated trajectory.

In case of Figure 3.4 (c) the pose recoverability is (at the instant shown in the figure);

$$p_r^i = 1 \mid \forall i \in \{1,2, \dots, 13\}$$

In case of Figure 3.4 (b) the pose recoverability is 0 for some of the robot segments.

As seen in the figures in some cases the support polygon stability criteria is not satisfied yet the robot does not fall over, instead it continues with its trajectory generated by the DPG. This loss of contact polygon stability to ensure the greater pose recoverability can be used to

generate a class of motions which cannot be generated by the central pattern generators. Furthermore the pose recoverability also allows the motion generation of under-actuated legged robots, which has been a challenge for conventional CPGs.

3.1.8 Action Subspaces

The action subspace of an actuated joint is the region of the \mathbb{R}^6 space within which the actuator forces can be applied to the robot segments without compromising the pose recoverability of the robot. These 6D limits of the robot joint forces and torques are an inverse solution of the pose recoverability. The following corollaries hold in all cases;

Corollary 1:

The joint force vectors which fall within the action subspace satisfy the check for pose recoverability of the robot.

Corollary 2:

The action subspace is a set of all Plücker's force vectors for a joint which satisfy the pose recoverability.

For a dynamic trajectory generated by the DPG;

$$f_A^i \in S_{Action}^i$$

The action subspace S_{Action}^i is the subset of 6D space real which denotes the actuator forces/torque for the i th joint.

$$S_{Action}^i \subset \mathbb{R}^6$$

The action subspaces of all the joints of the robot can be calculated numerically by using the pose recoverability functions and verifying it for all the joints. Essentially action subspaces are the region in which the solution for the decentralized pattern generator exist.

If no action subspace is found valid i.e. all force vectors fail the pose recoverability check, then the robot cannot generate any motion after which it can recover to a stable pose. The action subspace for a decentralized pattern generator can give significant insight during the robot design, particularly for the choice of actuators.

3.2 Motion Generation Process of a DPG

The DPG generates motion patterns by iteratively solving mathematical functions with defined purposes. Following steps are followed sequentially in the complete process;

3.2.1 Input Via-Poses

The Via Poses are given as the input to the DPG, these are the robot postures through which the robot moves. The Via-Poses are tested for stability and the recoverability margin is calculated in order to generate a stable motion. Intermittent motion can be generated by the DPG however for the complete motion the starting and ending poses should be Stable Poses.

The foot-step planner of a CPG can be replaced by a pose-planner for a DPG, this will generate the via-poses which make the robot segments move through their respective trajectories in order to accomplish the generated motion. A feedback based DPG pose-planner solves for the robot poses which can be used to interpolate between the starting and ending poses.

It must also be noted here that the via-poses given to the walking robot must be consistent i.e. the transition between one via-pose and the next via-pose should not take more than one step of the robot. If the via-poses are far apart then the via-poses are modified by the DPG to generate more intermediate poses but all of those will be the stable poses and that might increase the performance time and energy consumption of the robot.

3.2.2 Position Inverse Kinematics Check

The Position Inverse Kinematics Check is used to ensure that the via-poses do not break the robot structure i.e. the robot joint angles are within the limits and all the constraints are adhered to. Position Inverse Kinematics Check is performed for all poses and the poses which fail the check are modified iteratively until all the via-poses fixed.

3.2.3 Spline Generator

A 6D spline generator interpolates motion functions for the Plücker's vectors for all segments (links) of the robot. The spline generator interpolates between the via-poses and a velocity profile is generated for each segment of the generated spline. The velocity profile is required along with the position profile to generate the pose-recoverability margins.

The spline generator used in this study is a b-spline generator for the robot segments (links). It computes the position, velocity and acceleration profiles for all the segments in the

6D space using iterative solvers. The 3 6D vectors for position, velocity and acceleration are Plücker's vectors representing the linear and angular components for each segment of the robot.

3.2.4 Pose Recoverability Margin

The pose recoverability margin is solved using the position profile and velocity profile of all the segments (links) of the robot. The pose recoverability margin is generated by the following equation. This pose recoverability margin is used in the velocity inverse kinematics test to verify if the robot can recover to a stable pose.

3.2.5 Velocity Inverse Kinematics Check

The velocity inverse kinematics check ensures that the generated velocity profile b-splines adhere to the pose recoverability margins. This means that for all the generated velocity and position profiles the robot can recover to a stable pose after the generated motion is executed.

If the velocity inverse kinematics check fails for the generated pose recoverability margins a modified set of splines is generated and a newer set of pose-recoverability margins is generated. A unique pose-recoverability margin exists for each set of generated splines and hence a corresponding velocity inverse kinematics test is unique for all splines.

3.2.6 Equation of Motion Solver

The 6D equation of motion is solved for the 6D Plücker's force vector which generates the force vector for all the segments (links) of the robot. These generated 6D force vectors applied on the links are then solved by the distributed inverse dynamics to solve for the actuator forces applied to the joints of the robot.

The equation of motion used to solve for the motion of each segment is the rigid body equation of motion. It must be clarified here that the generated force vector is a 6D force applied on the links of the robot. An inverse dynamics solver is used to compute the actuator torques for each actuated joint of the robot.

3.2.7 Force Controller & Actuator Drive

A force/torque controller is used to apply the corresponding force profile to each joint. The force profile for each actuated joint is that generated by the DPG. The force profile is solved only for the actuated joints whereas the unactuated joints are not solved by the inverse dynamics solver.

The reason why unactuated joint forces/torques are not solved is because the force profile for each segment (link) of the robot already incorporates the motion of unactuated joints. The actuated joint forces are transmitted through the unactuated joints and the dynamic solution is the same. The mechanics of the robot system can be utilized to develop a control system [96].

For each link of the robot described by the equation of motion 3.2, a distributed control architecture of the form shown in figure 3.5 below can be developed which incorporates the DPG within the control loop, where each element of the link is moved by the combined force/torque vectors generated by all the actuated joints.

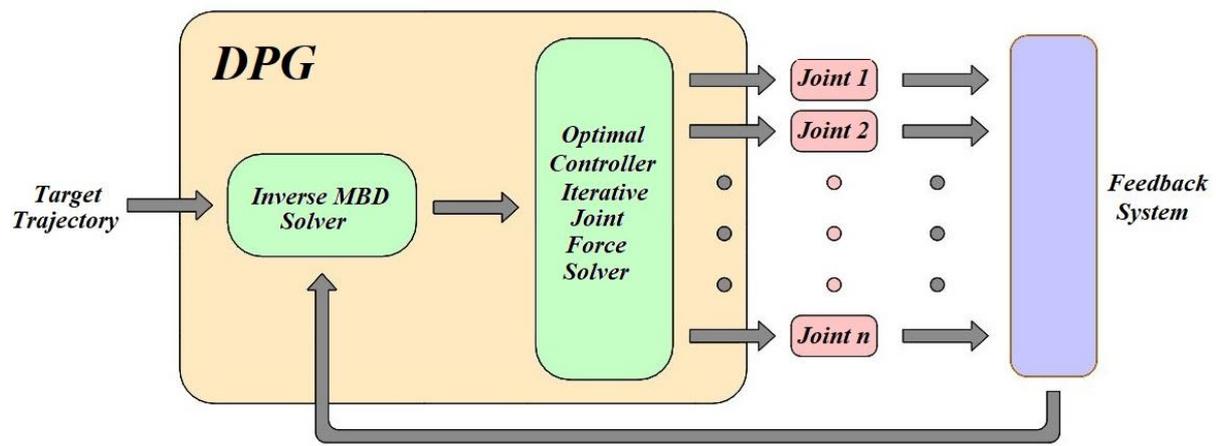


Figure 3.5 DPG as a Control System Architecture

The iterative force/torque solver takes the individual link forces, from the multi-body inverse dynamics solver, as input and computes the joint actuation force/torques iteratively. An optimal controller for solving the joint actuations can combine the equations 2.13 and 2.14 while maintaining the pose-recoverability margin of all the segments of the robot. This is a nonlinear optimal control which combines the thresholding pose recoverability function and the multi-body contact and constraint forces into a single block.

The control system architecture of the DPG is beyond the scope of this study and will not be treated here, the DPG in the form of a control system can be used along with a foot-step or a trajectory planner and a feedback network to allow online pattern generation, however this dissertation deals with offline pattern generation only.

If the actuated DOF of the robot change i.e. joint actuation is changed then the generated motion also changes because essentially a directly actuated joint is different from a reactively actuated joint. Thus a complete planning based algorithm must incorporate the DPG as well as the feedback network of the entire system in a closed loop topology. This will be evident in the

subsequent sections of the study where the motion generation is performed for both the fully actuated and the under-actuated configuration of the same robots.

3.2.8 Flowchart

The following flowchart describes the different steps followed by a DPG to generate the motion pattern of a walking robot.

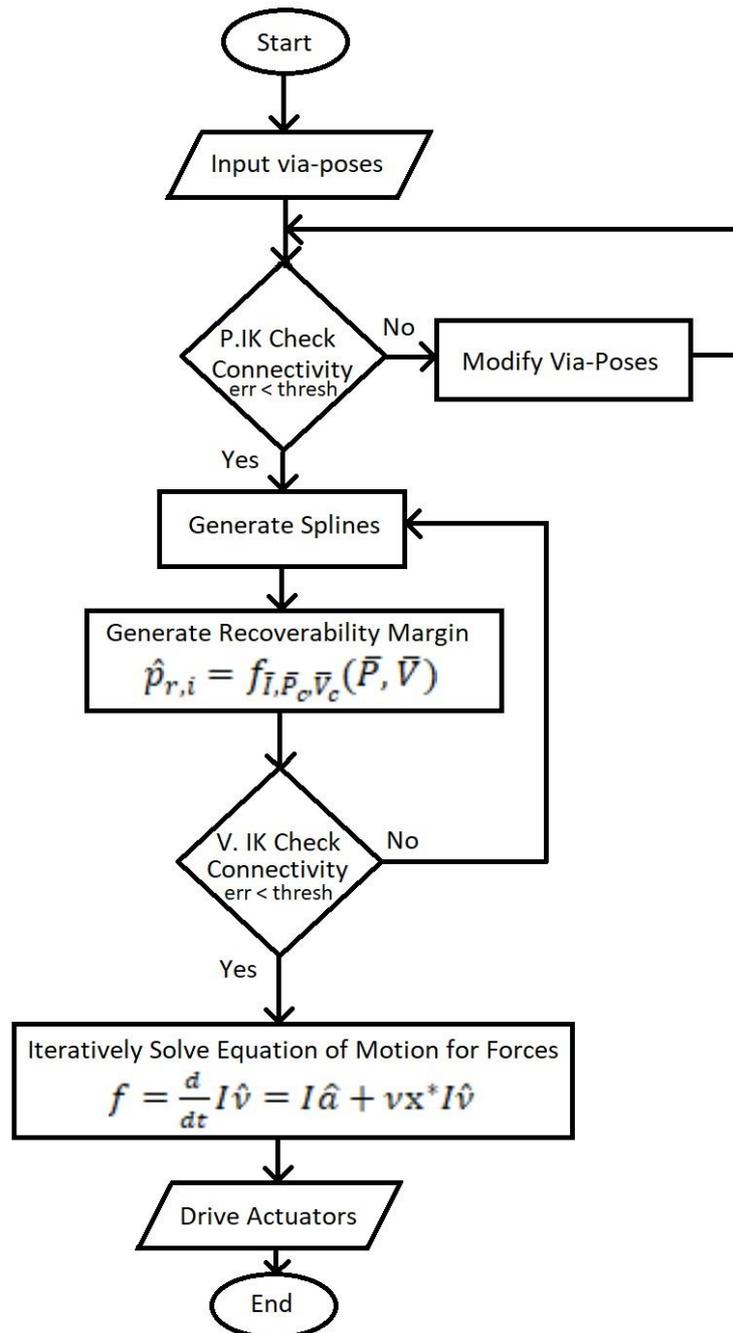


Figure 3.6 Decentralized Pattern Generation Flowchart

3.3 Mathematical Derivation of Decentralized Pattern Generators

A DPG utilizes multibody dynamics and rigid body dynamics to generate the motion of all the individual segments (links) of a robot instead of the motion of an estimated center of the robot like ZMP or COM. In order to solve for actuator forces following mathematical formulae are of importance;

The via-poses given by the trajectory planner or by the user, are tested for position inverse kinematics check while satisfying the positional constraints and used to by the spline generator (interpolator) to generate the trajectory profile of each rigid segment (link) of the robot.

$$\forall i; \hat{p}_i = f(\bar{P}_c, \{C_{via}\}) \quad (3.10)$$

Here C_{via} is the set of all via poses given as input to the DPG, also known as capture points or knots (in terms of spline generation).

Acceleration and velocity in 6D spatial coordinates is given by the double and single time derivatives of the desired trajectory respectively.

$$\hat{a}_i = \frac{d}{dt} \hat{v}_i \quad ; \quad \hat{v}_i = \frac{d}{dt} \hat{p}_i$$

And the 6D inertia tensor (in universal frame of reference) is given by;

$$\bar{I}_i = \begin{bmatrix} I - m\tilde{p}_c\tilde{p}_c & m\tilde{p}_c \\ m\tilde{p}_c^T & m1_3 \end{bmatrix} \quad (3.11)$$

The pose recoverability margins are tested for the velocity profile for each link and then tested for pose recoverability margin satisfaction, if the generated spline fails the pose-recoverability test, the generated splines for position, velocity and acceleration are updated until the pose recoverability margin is satisfied.

$$r_p^i = f_{\bar{I}, \bar{P}, \bar{V}}(\bar{P}, \bar{V}) \quad (3.12)$$

The 6D force vector acting on link i , as expressed in equation 2.8 is used in the following equation of motion of every link (rigid body) in the robot and the desired force is computed iteratively;

$$f^i = \frac{d}{dt} (\bar{I}_i \hat{v}_i) = \bar{I}_i \hat{a}_i + v_i \times^* \bar{I}_i \hat{v}_i$$

The net force acting on link i is given by equation 3.8 as following;

$$f^i = f_A^i + f_G^i + f_C^i$$

The force acting on link i due to gravity is the i th row of the generalized gravitational force/torque matrix \overline{F}_G . The generalized gravitation force matrix is a $n \times 6$ matrix;

$$\overline{F}_G = f(\overline{P}, \overline{I}_1, \overline{I}_2, \dots, \overline{I}_n) \quad (3.13)$$

Where the matrices $\overline{I}_1, \overline{I}_2, \dots, \overline{I}_n$ are the 6×6 inertia matrices for all the links, expressed in the universal frame of reference.

The constraint force acting on link i are the forces which are transmitted between the mutually constrained axes of translation and/or rotation at every joint, and the forces due to collision with the ground plane are computed by the contact force solver [3]. The force propagated through a rigid body adjacent to the link i be it in the form of collisions or joined segments of the robot must be transformed into the universal frame of reference.

$$f_C^i = f_C^i(\overline{P}, \overline{V})$$

The actuator force is combined force/torque applied by all the actuated joints to the link i , as in the case of constraint force estimation, the forces acting on link i must be transformed into the universal frame for the consistency of the solution;

$$f_A^i = \sum_{n=1}^N T_n^U(f_n^i)$$

$$n \in \{1, 2, \dots, N\}$$

In this equation f_n^i is the force acting on link i due to the actuator n , transformed into the universal frame by the function $T_n^U()$.

The penultimate step is to solve the actuator forces for each actuator in the actuator frame, this is done by iteratively solving the following system of non-linear transcendental equations for the actuated joint force/torque vectors for each link f_n^i .

The set of equations 2.12 is used to iteratively solve the joint force/torque vectors. In this description the 6D Plücker's force/torque vector f_n^i is the force due to the actuator n on link i .

$$\begin{aligned}
f_A^1 &= T_1^U(f_1^1) + T_2^U(f_2^1) + \dots + T_n^U(f_n^1) \\
f_A^2 &= T_1^U(f_1^2) + T_2^U(f_2^2) + \dots + T_n^U(f_n^2) \\
&\vdots \\
f_A^n &= T_1^U(f_1^n) + T_2^U(f_2^n) + \dots + T_n^U(f_n^n)
\end{aligned} \tag{3.14}$$

The values for all the 6D force/torque vectors are signed quantities. Lastly in order to determine the force/torque to be generated by the actuator (or the force/torque control system applied on to the actuator) is given by;

$$f_{Act-n} = T_U^n \left(\sum_{\forall i} f_n^i \right)$$

It must be noted that since the system is non-linear and transcendental, it is however invertible, but the transformation function from the universal frame of reference to the actuator frame T_U^n , is not constant and changes according to the change in position and orientation of the actuated joint under consideration, the parentheses have been dropped for the sake of convenience in notation, however the transformation functions are dependent upon the robot position matrix.

$$f_{Act-n} = T_U^n(\bar{P}, \sum_{\forall i} f_n^i) \tag{3.15}$$

These formulae are used iteratively by the DPG in the following sections for pattern generation of the specified motion of the robot under consideration. Using equations 3.10 through 3.15 numerically to compute the actuator forces the decentralized pattern generator generates the actuation sequences prior to execution by the robot.

3.4 CASE I: Planar Single Inverted Pendulum

The first design that will be considered is a simple 1 DOF inverted pendulum. The system has a fixed pivot in the world frame an actuator is connected to the pivot and the output shaft of the revolute joint rotates the terminal segment (link) of the pendulum.

3.4.1 Mechanical Design

The mechanical design of a planar single inverted pendulum is very straightforward, it consists of a rigid rod connected to the shaft of a motor at the pivot. The following figures show the simulation model and the equivalent multi-body diagram of the planar single inverted pendulum.

3.4.2 Stable Pose

The stable state of the single inverted pendulum is the state at which the angle of the inverted pendulum is 90° with respect to the ground i.e. the pendulum is balanced vertically. The grey cylinder in the figure below describes the stable pose of the inverted pendulum, while it is referenced from the positive y axis.

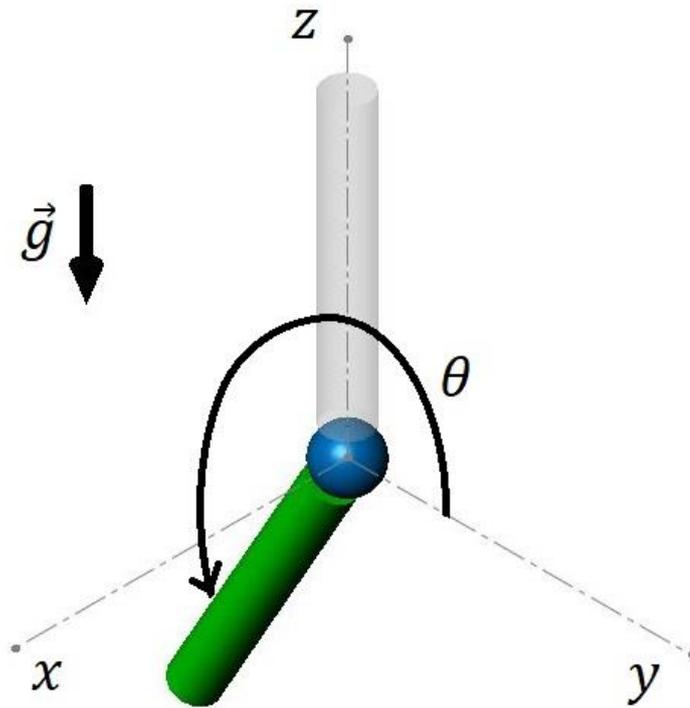


Figure 3.7 Cylindrical Single Pendulum

3.4.3 Target Trajectory

Two desired trajectories are to be generated by the DPG for the single inverted pendulum;

- 1- Stable pose maintaining trajectory, given finite external disturbances.
- 2- Sustained oscillations of the end effector around the stable pose with a desired amplitude.

3.4.4 DPG Solution

The target trajectories are the position profiles and their time derivatives give the velocity profiles. The mass and mass distribution of the rotating link are represented by m inertia matrix I . Both of these are used to solve for the pose recoverability margin. The actuator torque generated as a result by the DPG is given as the input to the motor torque controller.

3.5 CASE II: Fully-actuated Double Pendulum

The double pendulum is a 2-R robot i.e. it consists of two revolute joints connected to a base frame. The fully-actuated double pendulum consists of two actuated revolute joints. The 2 DOF design has 2 joints whose axes are parallel.

3.5.1 Mechanical Design

The fully-actuated double pendulum consists of two rods connected to two actuated revolute joints in a single series link. The mechanical design is shown in the following figure;

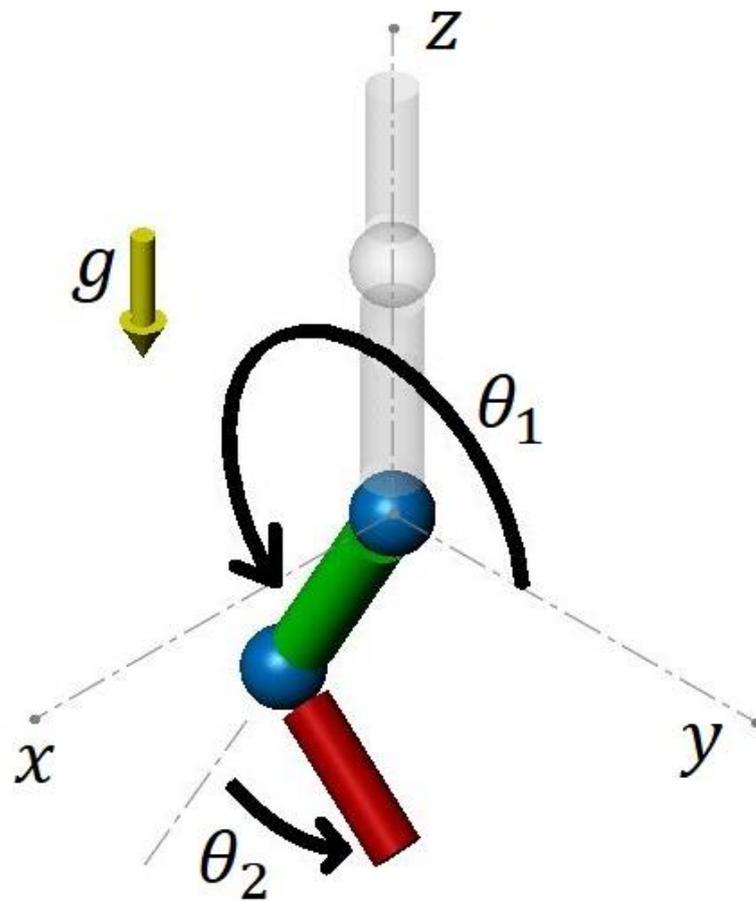


Figure 3.8 The Double Inverted Pendulum Model.

3.5.2 Stable Pose

The stable pose of the fully actuated double pendulum is standing vertically upwards i.e. $\theta_1 = 90^\circ$ and $\theta_2 = 0^\circ$. Both the links are balanced vertically in the stable pose, this is the unstable equilibrium pose (unstable fixed point) from which the average energy required to change into any other target pose is minimum.

3.5.3 Target Trajectory

The following two target trajectories, similar to the single inverted pendulum, will be generated by the DPG for the fully actuated double pendulum;

- 1- Maintenance of the stable pose, under finite external disturbances.
- 2- Sustained oscillations of the end effector around the stable pose, with-in a defined amplitude.

3.5.4 DPG Solution

The target trajectories are used as the position profile and the time derivative of the target position profile gives the velocity profile.

3.6 CASE III: Under-actuated Double Pendulum (Acrobot)

The modified version of the fully actuated double pendulum in which θ_1 is unactuated will be studied in this case. The spatial distribution is the same in both cases, however the actuation arrangement is slightly different, the figure 3.9 shows both the cases side by side. The under-actuated version of the double pendulum is an interesting control problem and has been studied under the title Acrobot.

3.6.1 Mechanical Design

This is similar in design to the fully-actuated double pendulum with the difference being that the first joint is a free rotating joint and only the second revolute joint is actuated. This design shows how a DPG can generate the force / torque actuation pattern for the under-actuated system.

3.6.2 Stable Pose

The stable pose for the under-actuated double pendulum is the same as the fully-actuated double pendulum case i.e. both rods vertically upward. The unactuated joint angle θ_0 is 90° and the actuated joint is kept at 0° .

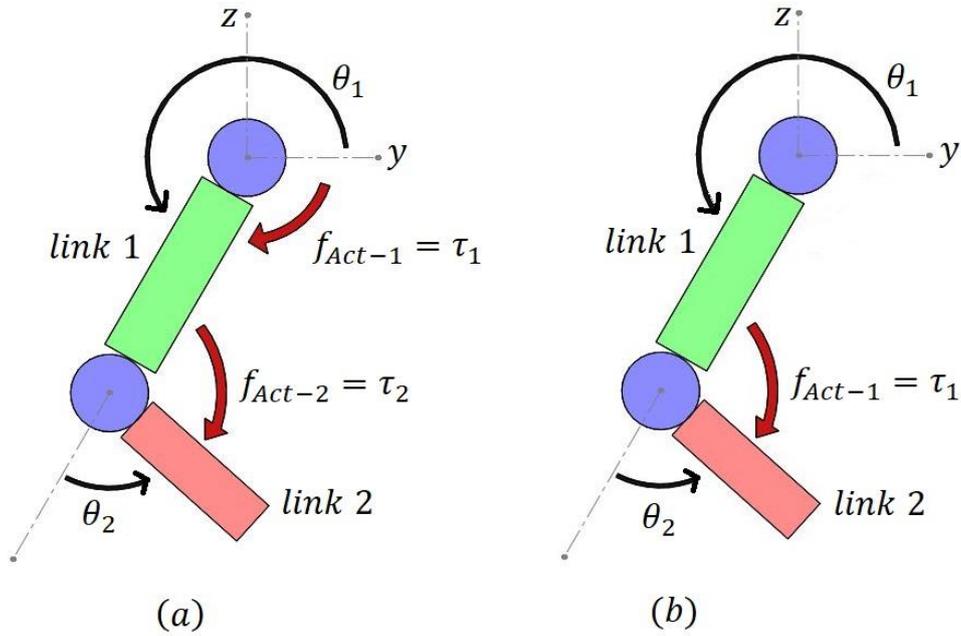


Figure 3.9 Double Pendula; (a) Fully-Actuated (b) Under-Actuated

3.6.3 Target Trajectory

The target trajectory of interest in case of the under-actuated double pendulum is that of swinging up from the vertically downward position to the stable pose (vertically upward). Only the terminal joint is provided actuation and the corresponding dynamical response of the base joint is studied as the robot brings itself to the vertical balance pose.

3.6.4 DPG Solution

The DPG generates the force profiles for the actuated joint for the three trajectories and the force/torque controller executes the trajectories for the under-actuated double pendulum.

3.7 CASE IV: Fully-actuated 12 DOF Humanoid Robot - NUSTBOT-3

A fully actuated 12 DOF humanoid robot model is used as a test bench for the pattern generation problem for a DPG. The software model of NUSTBOT-3 is shown below in the figure 3.10. It contains 3 DOF hip joints, 1 DOF knee joints and 2 DOF ankle joints. The robot NUSTBOT-3 is under development at the Robotics & Intelligent Systems Engineering Laboratories & Research Center at SMME, NUST.

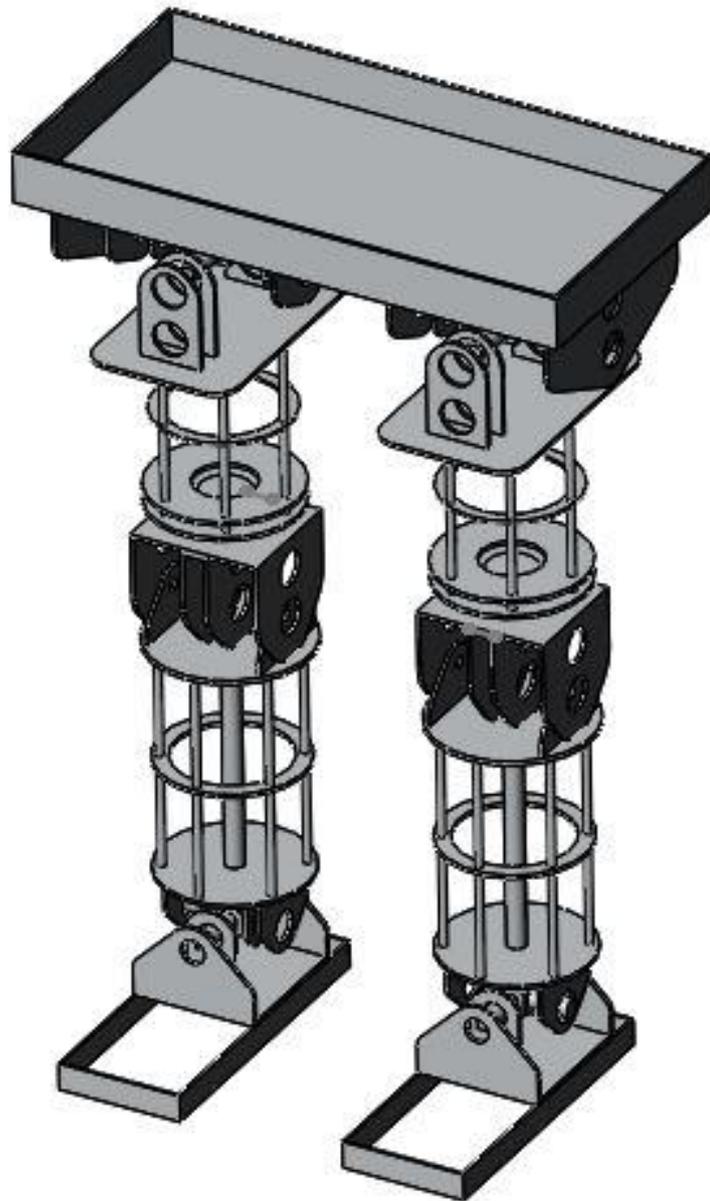


Figure 3.10 NUSTBOT 3 Simulation Model.

3.7.1 Mechanical Design

The hip pitch and roll joints are connected to a cross shaft inside the hip joint. The hip yaw joint is inside the thigh of the robot which rotates the knee. The knee joint is connected between the knee and the shin of the robot. The ankle joints are connected to the foot through a cross shaft inside the ankle of the robot. The robot consists of the 9 major body parts which are connected together through 6 or 8 joints depending upon the reference topology used. All 12 DOF are actuated through motors with torque controllers. The revolute joints of the right leg are shown in the figure 3.11. The (hip) yaw axis is highlighted in purple, the pitch axes are highlighted in blue and the roll axes are highlighted in red.

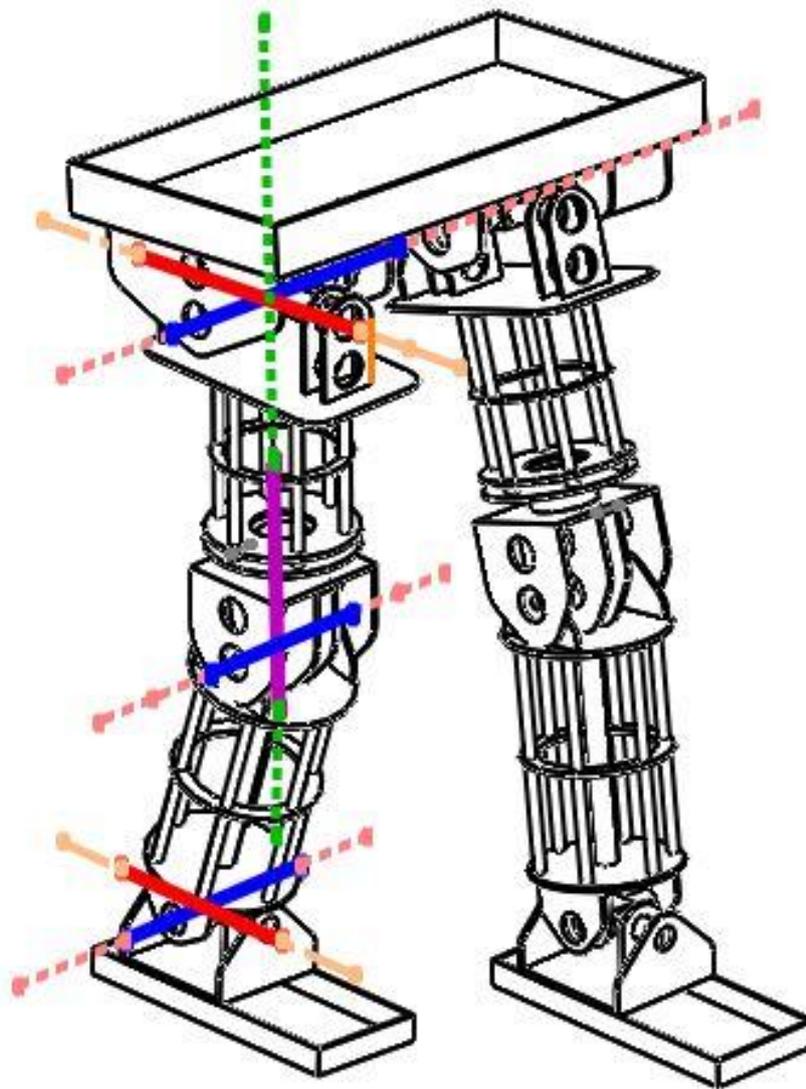


Figure 3.11 NUSTBOT-3 Revolute Joint Axes in right leg.

Although the actual hardware robot NUSTBOT-3 has 13 links (rigid segments), however there are 2 reduced simulation models as well which consist of 9 and 7 links. For our simulations we will use the 9 link models for both the fully-actuated and the under-actuated versions.

3.7.2 Stable Pose

The stable pose for the fully-actuated version of NUSTBOT-3 is defined as the robot standing upright with both feet in contact with the ground. This simulation model uses the area under the foot as the contact area with the ground plane. The joint configuration of the robot in stable pose is shown in the table 3.1 below along with the figure 3.12 showing NUSTBOT-3 in its stable pose.

| Axis - Right Leg | Angle (Degrees) | Axis - Left Leg | Angle (Degrees) |
|------------------|-------------------------------|-----------------|-------------------------------|
| Hip Pitch | $\theta'_{Rhp} = 289.8^\circ$ | Hip Pitch | $\theta'_{Lhp} = 70.2^\circ$ |
| Hip Roll | $\theta'_{Rhr} = 90^\circ$ | Hip Roll | $\theta'_{Lhr} = 90^\circ$ |
| Hip Yaw | $\theta'_{Rhy} = 0^\circ$ | Hip Yaw | $\theta'_{Lhy} = 0^\circ$ |
| Knee Pitch | $\theta'_{Rkp} = 21.6^\circ$ | Knee Pitch | $\theta'_{Lkp} = 338.4^\circ$ |
| Ankle Pitch | $\theta'_{Rap} = 99.7^\circ$ | Ankle Pitch | $\theta'_{Lap} = 279.7^\circ$ |
| Ankle Roll | $\theta'_{Rar} = 0^\circ$ | Ankle Roll | $\theta'_{Lar} = 0^\circ$ |

Table 3.1 NUSTBOT-3 Stable Pose Joint Configurations

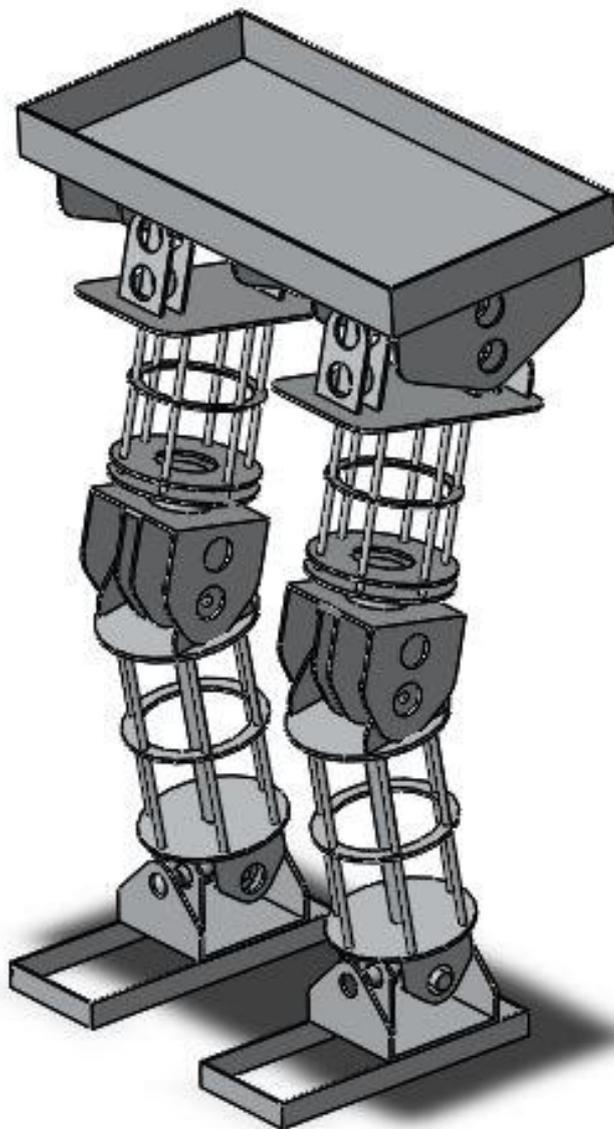


Figure 3.12 NUSTBOT-3 in the Stable Pose

3.7.3 Target Trajectory

To generate the motion of the fully actuated NUSTBOT-3 with the DPG we will use the walking motion of the robot as a target trajectory. For the sake of comparison, the fully actuated NUSTBOT-3 model will also be simulated for straight walk with a CPG using the ZMP stability criteria.

3.7.4 DPG Solution

The force profiles are generated for all 12 joints of the fully actuated version of NUSTBOT-3, by using the straight walking motion as the target trajectory. The same trajectory is used for both the central and decentralized pattern generators. An interesting observation that will be shown in the next chapter is the change in torque profile for the ankle motors with the choice of pattern generator chosen.

3.8 CASE V: Under-actuated 12 DOF Humanoid Robot - NUSTBOT-3

The final case under this study is the under-actuated version of the 12 DOF humanoid robot NUSTBOT-3. The design is similar to the previous case but the ankles of the robot are unactuated. The feet of the robot can rotate freely about the ankle axes.

3.8.1 Mechanical Design

The under-actuated 12 DOF humanoid robot NUSTBOT-3 contains 8 joints out of which only 6 are actuated. The actuated joints are 2 DOFs as hip pitch and roll, 1 DOF hip yaw and 1 DOF knee pitch. The 2 DOF ankle roll and pitch joints are unactuated. This model corresponds the closest to the actual hardware of the robot shown in figure 3.13 below. The ankles of the robot are un-torqued but damped with springs along both the pitch and roll axes of both ankles. The hip and knee joints are fully-actuated whereas the ankles are un-actuated.

A differentiation to be made here is that although, virtually, every humanoid robot is by definition an under-actuated system. In the course of this study to distinguish between the two cases of the 12 DOF humanoid robot NUSTBOT-3, with 8 actuated joint axes and 12 actuated joint axes, the fully-actuated model refers to the NUSTBOT-3 model with 12 actuators for 12 DOF, whereas the under-actuated model refers to the NUSTBOT-3 model with reduced actuation, 8 motors for a total of 12 DOF.



Figure 3.13 The Humanoid Robot NUSTBOT-3 Hardware Model with un-torqued ankles

3.8.2 Stable Pose

The stable pose of the under-actuated version of NUSTBOT-3 is chosen to be the same as that for the fully-actuated version of the simulation model, the robot joint configuration is described in the table 3.1 and the figure 3.11 shows the same.

3.8.3 Target Trajectory

The target motion trajectory for the under-actuated version of NUSTBOT-3 is the walking gait starting from the upright stable pose and ending in the stable pose.

3.8.4 DPG Solution

The actuator force/torque for the 8 actuated joints is solved using the DPG approach. The comparison between decentralized pattern generator and a central pattern generator based on either COM or ZMP cannot be made because central pattern generators do not solve the reduced actuation problem for legged robots.

3.9 Final Note on Pattern Generators

The central pattern generators incorporate very high gain feedback approach (high stiffness control) which overrides the system dynamics and hence a complete actuation is necessary for that to work.

The decentralized pattern generator treats the robot motion generation problem as a multi-body dynamics problem instead of a fully actuated, high gain (high stiffness) control problem. This expands not only the type of systems the solution or the technique can be applied to but also the types of solutions which can be generated.

Whereas the central pattern generator approach is a very quick fix for most of the walking problems of full actuation legged robots, it doesn't provide much flexibility when it comes to dynamical motions, such as jumping robots, robot maneuverability with spatial contacts instead of planar contacts and the mechanical or dynamical fault tolerance in applied robotic systems. Some solutions have been proposed to such problems [16], [96], [97] but a general solution cannot be acquired without considering the complete dynamics of the system.

The decentralized pattern generator expands the robot from a single central description (ZMP or COM) to an interplay of multi-body dynamics and then expands further into rigid body dynamics for all the segments of the robot.

4 Analysis & Results

Computational Multibody Dynamics is a very broad field of study in and of itself. Dynamical systems modeling, setup and analysis incorporate a very wide range of subjects related to robotics. In this chapter we will start by describing the simulation and analysis test-bench used to analyze the systems under this study.

The software simulations of the multi-body dynamics solutions are performed in the SimMechanics module of Simulink MATLAB. The results are verified through numerical solvers for multi-body dynamics in the MATLAB environment. The set of equations used are explained in sections 3.1, 3.2 and 3.3 above. The important observations are shared and significant insights are drawn from the analyses performed.

4.1 Multi-body Simulation

The simulation of multi-body dynamical systems, particularly those involving spatial contacts, is a computationally expensive process. The best approach to streamline the process is to make computational dynamics iterative, the solvers used for numerical solutions for our case are based on Newton's method, which has fast convergence. Also using intelligent guesses for initial conditions (primarily drawn from the Denavit-Hartenberg parameters or the spatial construction of the robot) can reduce the computation time significantly.

The rigid body mechanics employed in this study involves the homogenous 6D Plücker's spatial vectors, hence these can be quite cumbersome to follow through, in our case defining matrices which aren't directly used in any equation but which have a 1-1 correspondence between rows of the matrix and links of the robot reduced the complexity of the solutions significantly.

SimMechanics is the mechanical simulation module of the Simulink environment of MATLAB, with very powerful solvers which can effortlessly generate numerical solutions for very complex problems such as spatial collisions, multi-body dynamics, rigid body dynamics and control system optimization. The systems under consideration were simulated, solved and tested within the Simulink SimMechanics environment. A few of the plots exported from the MATLAB workspace are shared in this following sections.

4.2 DPG Motion Generation

As stated earlier the decentralized pattern generator solves the multi-body dynamics and rigid body dynamics of all the links in a robot and iteratively solves the joints force/torques functions required by the actuators to generate the solutions.

Most, if not all, the equations of almost all the systems discussed under this study are non-linear and hence the generated force/torque time functions are numerically computed. As mentioned previously the analytical solution can be computed in some of the cases, but generally the iterative numerical solution is far more feasible when it comes to application.

4.3 CASE I: Single Inverted Pendulum

For the single inverted pendulum when actuator constraints are applied to the system, the DPG generates a solution based on the dynamics of the pendulum. The two trajectories for which motion sequences were generated are swinging up to the vertical position and sustained oscillations about the stable pose.

For the first trajectory (swing up and balance) the DPG solution is the back and forth swinging trajectory which brings the robot to the vertical position.

The following figure shows the plot for angle θ measured from the positive y axis counter-clockwise for the first 10 seconds, under constrained joint torque.

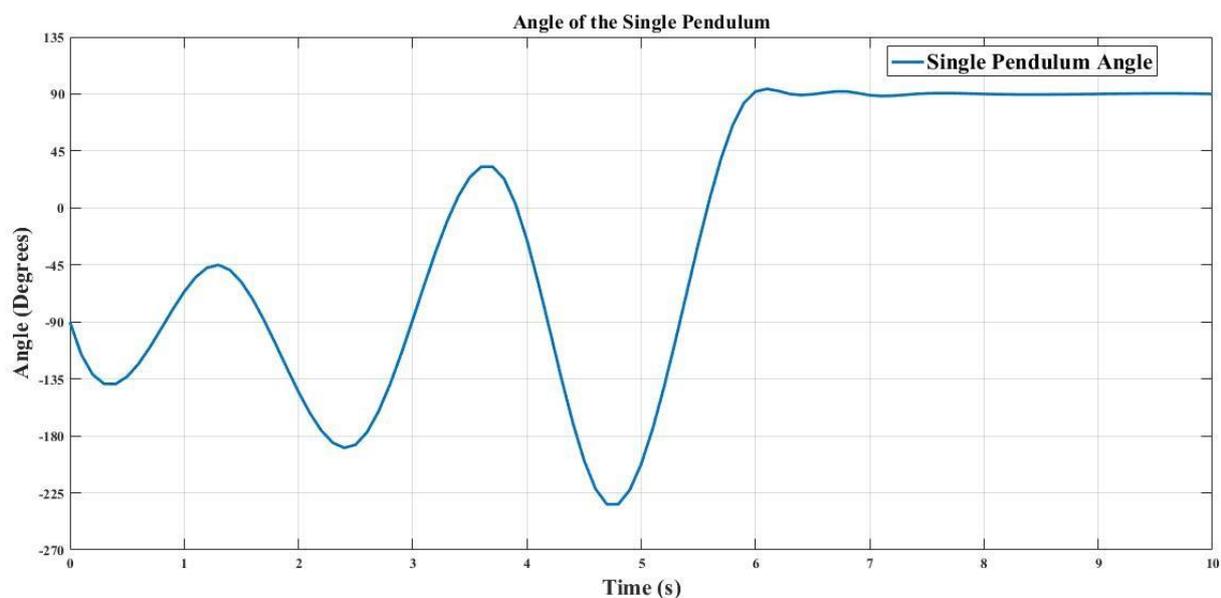


Figure 4.1 Angle of the single inverted pendulum with only torque constraints

The pendulum oscillates back and forth until it reaches the top and balances at the vertical stable pose, notice how this is the case for an unconstrained joint angle θ , measured from the positive y axis in the counter clockwise direction. If the case is of a constrained joint angle instead of a constrained joint actuator, the DPG generates a solution similar to the one generated by a high gain feedback (high stiffness) control system.

The following figure shows the solution generated by a high gain feedback (stiffness) PI controller tuned manually for the same vertical balancing problem and the gradually moving trajectory generated by the DPG.

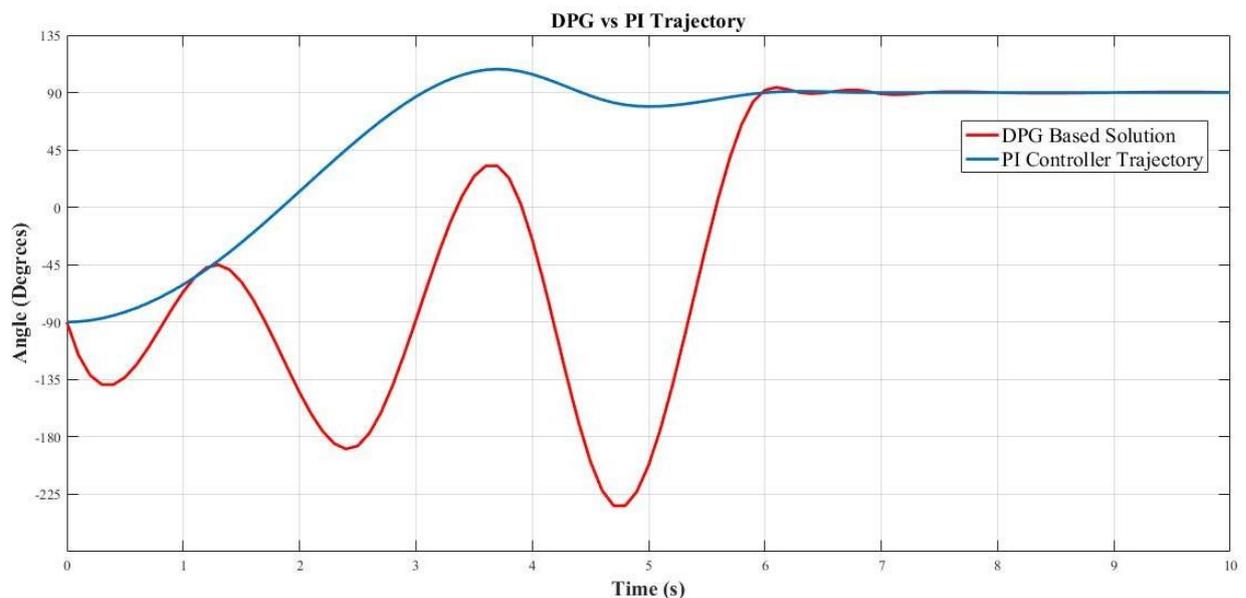


Figure 4.2 Angle trajectory comparison DPG vs PI Controller

The trajectory generated by the high gain feedback PI controller generates a solution without any constraints, whereas the DPG solution is constrained only by the actuator torque. It is worth mentioning here that the DPG solution cannot be generated if there is an angle constraint as well as the torque constraint, simply put there are no trajectories which can bring the pendulum to a vertical standing position if both the joint and the joint torques are constrained within absolute limits.

In order to avoid confusion about the angle wrap around ($+180^\circ = -180^\circ$) the plots in figure 4.1 and 4.2 are not absolute limited around $\theta = \pm 180^\circ$, instead the relative angular positions are plotted.

For the second target trajectory (moving about the stable pose with sustained oscillations) we get the trajectory plot shown in the figure 4.3. The trajectory is bounded in a $\pm 20^\circ$ region

about the vertical position, demonstrated by the dashed red lines in the plot. In this case the DPG generates a solution which is both angle and torque constrained, because the constraints do not force the robot (single inverted pendulum) to violate the pose-recoverability margin.

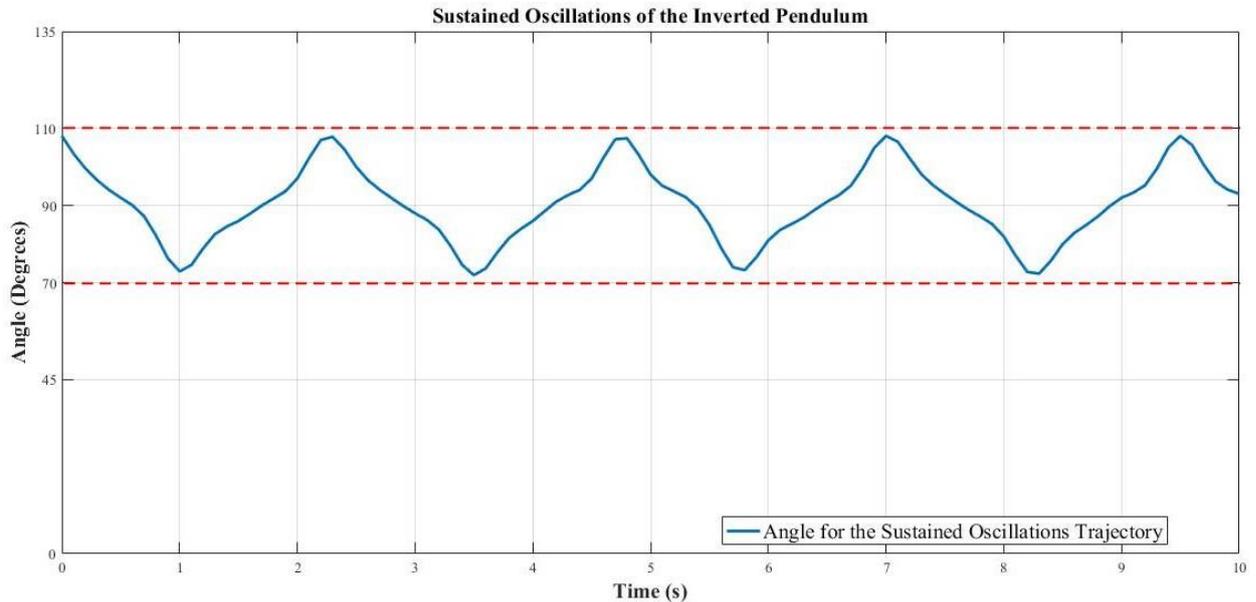


Figure 4.3 Sustained Oscillations of the Single Inverted Pendulum

From both of these trajectories we can draw a very important insight about the pose recoverability margin that, the pose recoverability margin can also vary depending upon the type and values of the robot position and velocity constraints as well as the actuator torque constraints (contributing towards the action space of the robot).

4.4 CASE II: Fully-actuated Double Pendulum

The first target trajectory for the fully-actuated double pendulum is executed to maintain the stable pose (vertically upward) while rejecting external disturbances. The actuators and robot segments are not constrained for this test case, in order to analyze for external disturbances of varying magnitudes.

The following figure 4.4 shows the trajectory of the angles of the fully-actuated double pendulum starting from the stable pose under the effect of external disturbances as generated by the DPG. The double pendulum tries to reject the external disturbances by actuating both joints. The disturbance torque acts on the base of the pendulum (joint 1) but the effects are also observed on the terminal link due to the dynamics of the double pendulum. The external disturbance torque acts differently on both links.

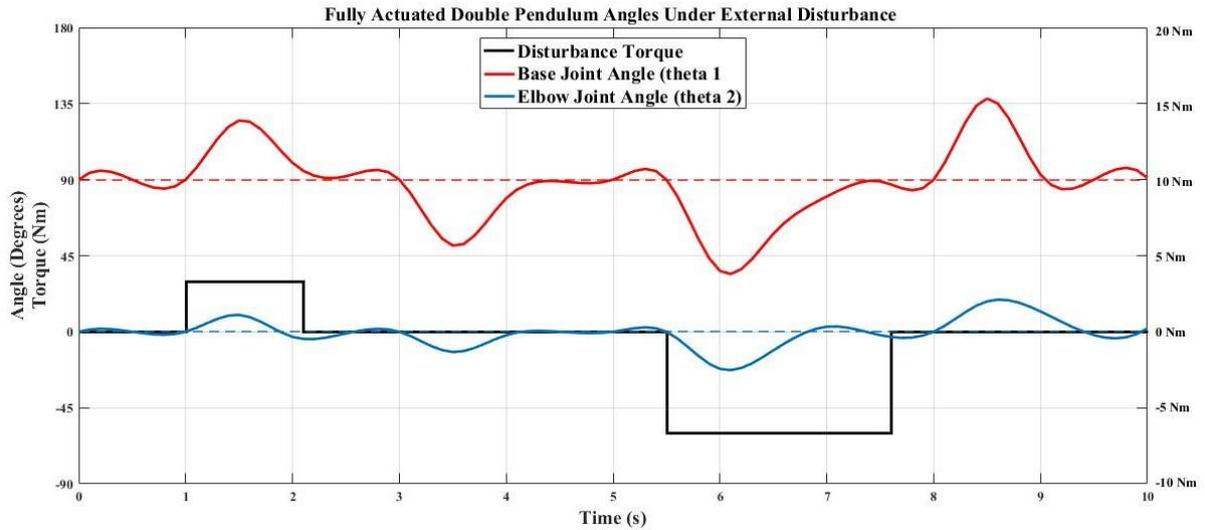


Figure 4.4 Fully Actuated Double Pendulum Angles under External Disturbance

The second case of the fully actuated double pendulum (i.e. sustained oscillations of the end effector) can have multiple solutions, the DPG allows a number of trajectories for the end-effector at a given angle. A few of the solutions executed are shown below in the consequent figures 4.5, 4.6 and 4.7.

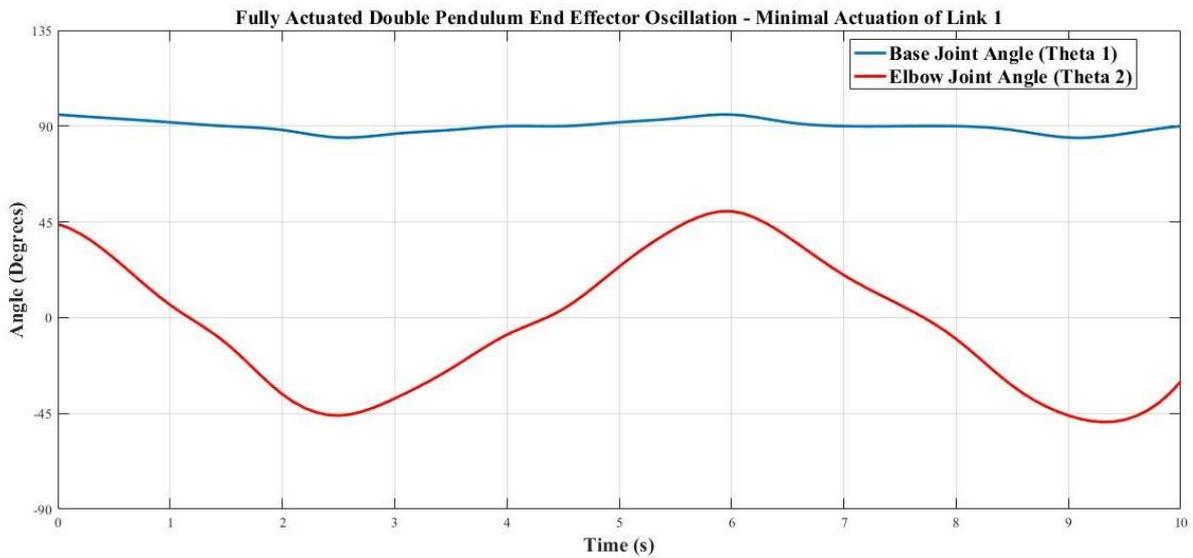


Figure 4.5 Fully Actuated Double Pendulum - End Effector Oscillation with Stiff Base

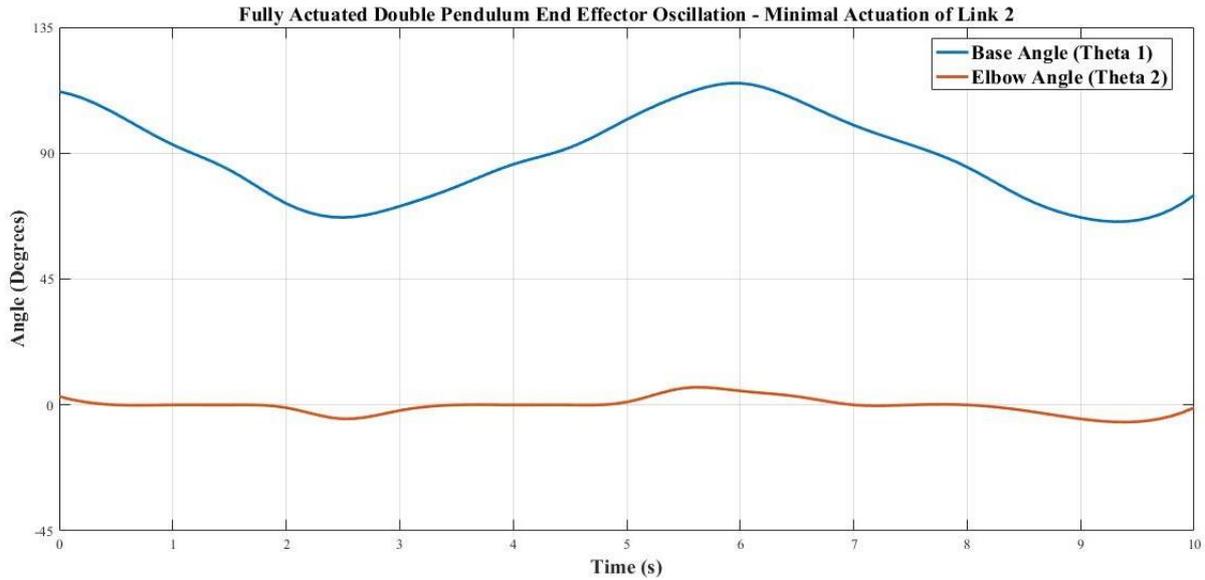


Figure 4.6 Fully Actuated Double Pendulum - Oscillation with Stiff Elbow

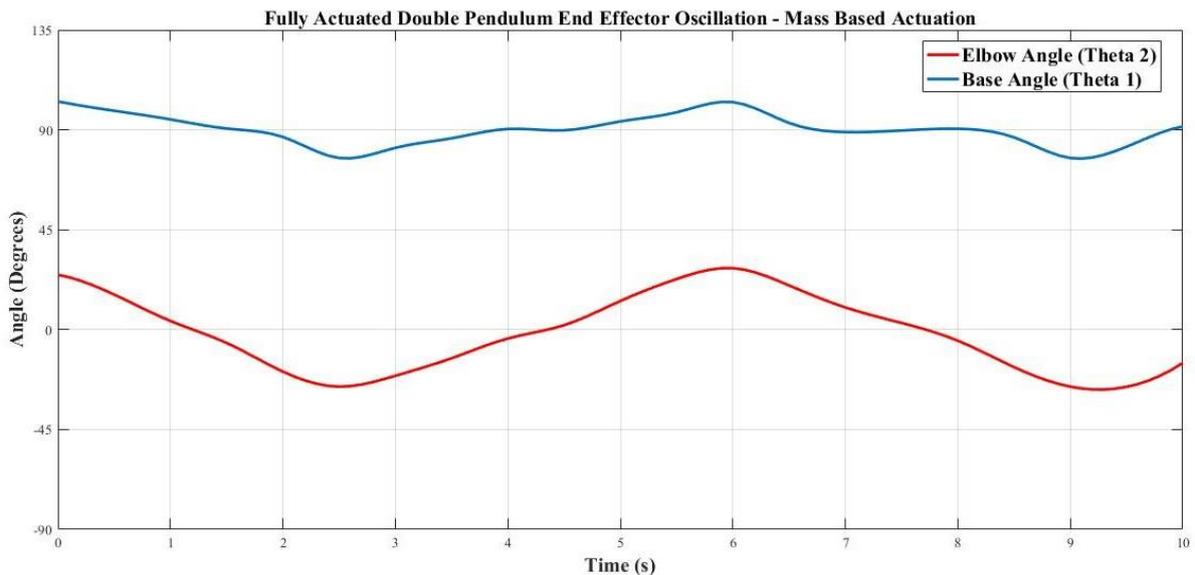


Figure 4.7 Fully Actuated Double Pendulum - Oscillations with proportional actuation

The primary difference between these three solutions is that in case of figure 4.5 the base link is move minimally and maximal actuation is performed on the terminal link. In case of figure 4.6 the base link is moved maximally and the terminal link is moved minimally. In the third case shown in figure 4.7 both the joints are actuated in proportion to the masses of both links. In all of these cases the constraints are set only on the angle of the displacement vector of the end effector from the positive y axis. If additional constraints are added, the number of solutions decreases significantly (action sub-space converges).

4.5 CASE III: Under-actuated Double Pendulum

The target trajectory for the under-actuated double pendulum is the case in which the robot starts hanging down and gradually swings itself up to the vertical stable pose. The joint torque of the actuated joint (terminal joint) is constrained and the joint angles of both joints are unconstrained.

The angular trajectories for both (actuated and unactuated) joints for the under-actuated double pendulum to start from the vertically downward position and arrive at the vertically upward position are shown in the figure 4.8 below.

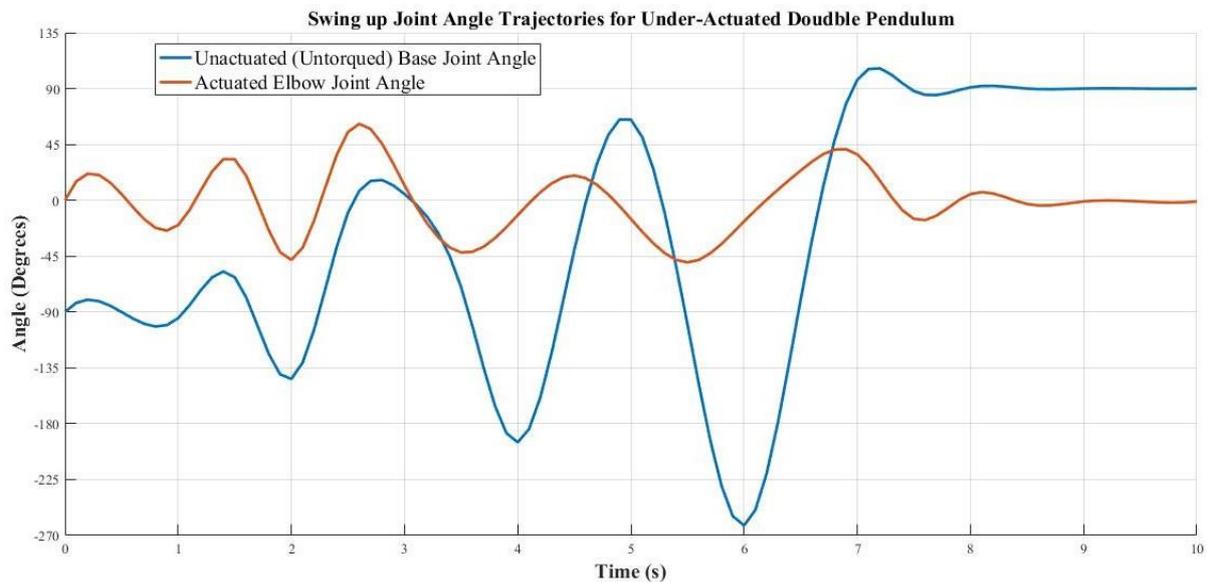


Figure 4.8 Swing-up trajectory angle profiles of the Under-Actuated Double Pendulum

This plot shows an important observation, the terminal link is always ahead of phase as compared to the link connected to the un-actuated revolute joint at the base of the pendulum. This correlates to the idea that the pendulum oscillates back and forth with increasing amplitudes until it reaches the vertically upward angle of 90° . As in the previous angle plots in order to cater for the wraparound of the joint angles ($+180^\circ = -180^\circ$) the joint angles are plotted for the relative positions, with respect to the starting angles.

An interesting observation here is that the elbow joint has a significantly smaller range of operation as compared to the base joint (un-actuated joint). This is because since the applied torque of the actuated joint is constrained even if the angle is not constrained it cannot exceed a bound beyond which no further torque can be applied by the actuator.

4.6 CASE IV: Fully-actuated 12 DOF Humanoid Robot - NUSTBOT-3

The fully-actuated version of NUSTBOT-3 is described in the previous chapter and the target trajectory for the fully-actuated NUSTBOT-3 is walk on an even surface. Two different solutions were generated for the fully-actuated humanoid robot, one by using a ZMP based central pattern generator and the second by using a DPG.

In order to highlight the differences between the DPG and CPG motions a single step out of the entire gait cycle will be considered.

The figure 4.9 shows the different via poses for the stepping motion of NUSTBOT-3.

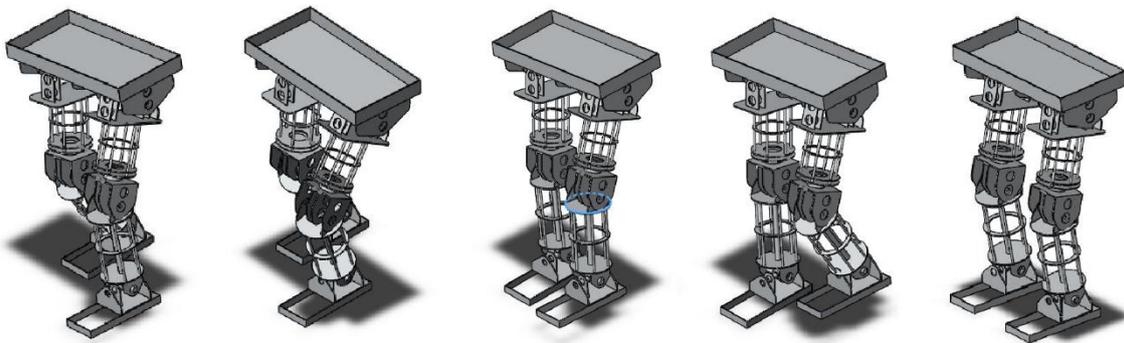


Figure 4.9 NUSTBOT-3 Walking Via Poses

The ZMP trajectories for the stepping motion generated by the CPG are shown below in figure 4.10, and the corresponding ZMP trajectories for the stepping motion generated by the DPG are shown in figure 4.11.

A prominent difference between the two ZMP profiles is that the DPG generates a far smoother ZMP profile as opposed to the CPG.

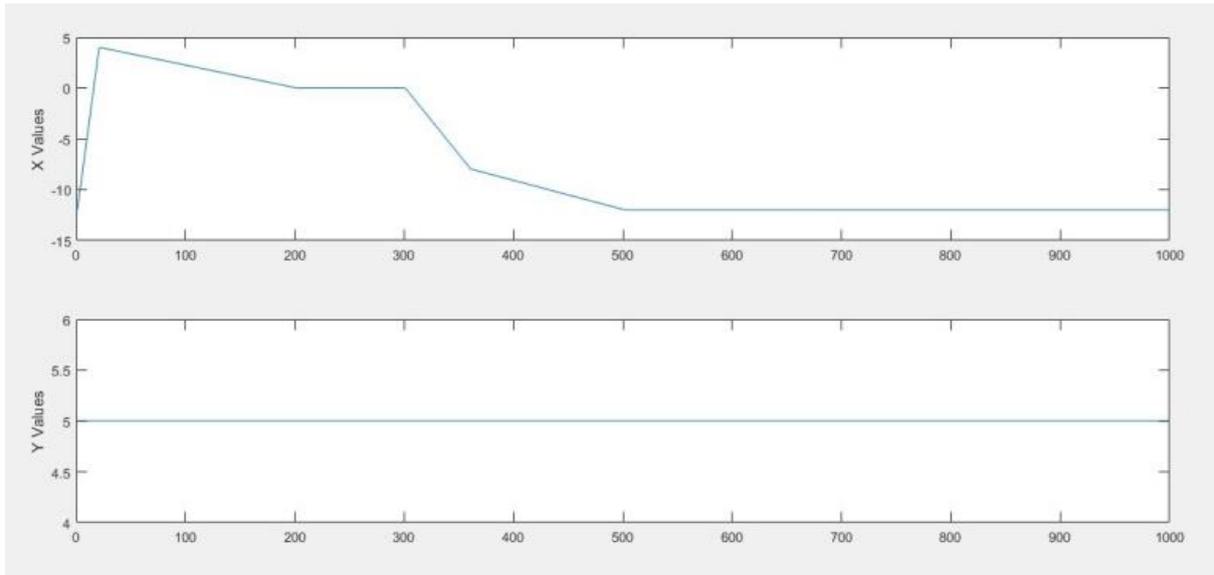


Figure 4.10 ZMP trajectory during step for fully-actuated NUSTBOT-3 by CPG

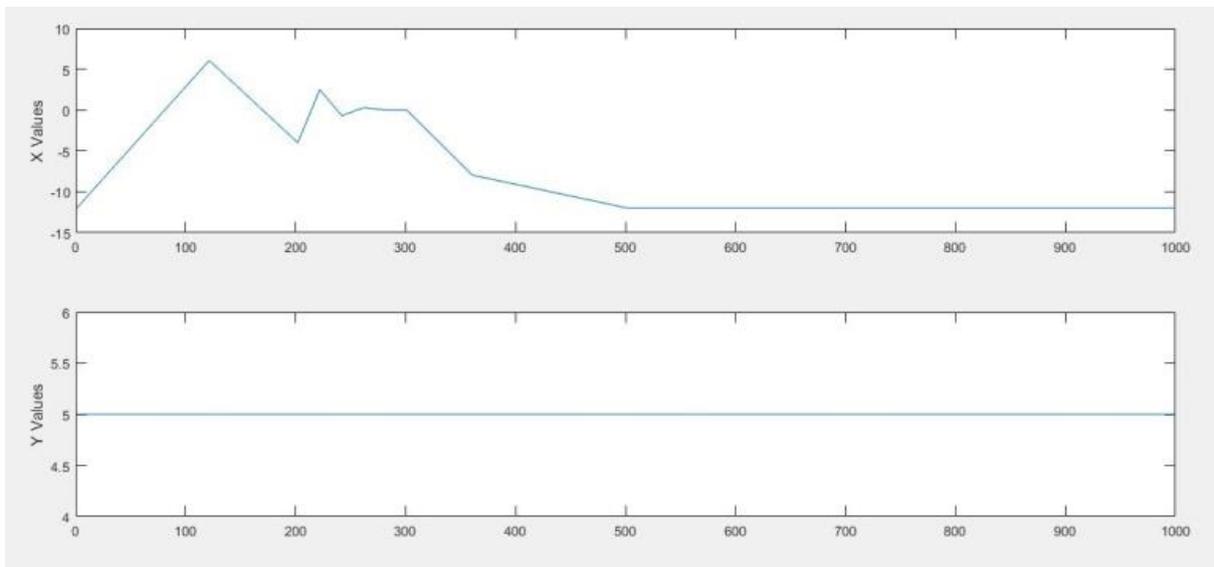


Figure 4.11 ZMP trajectory during step for fully-actuated NUSTBOT-3 by DPG

Another important difference is the ankle torque applied by the supporting leg during the stepping phase. The right ankle pitch during the stepping motion in case of the motion generation by a CPG is shown in figure 4.12 below.

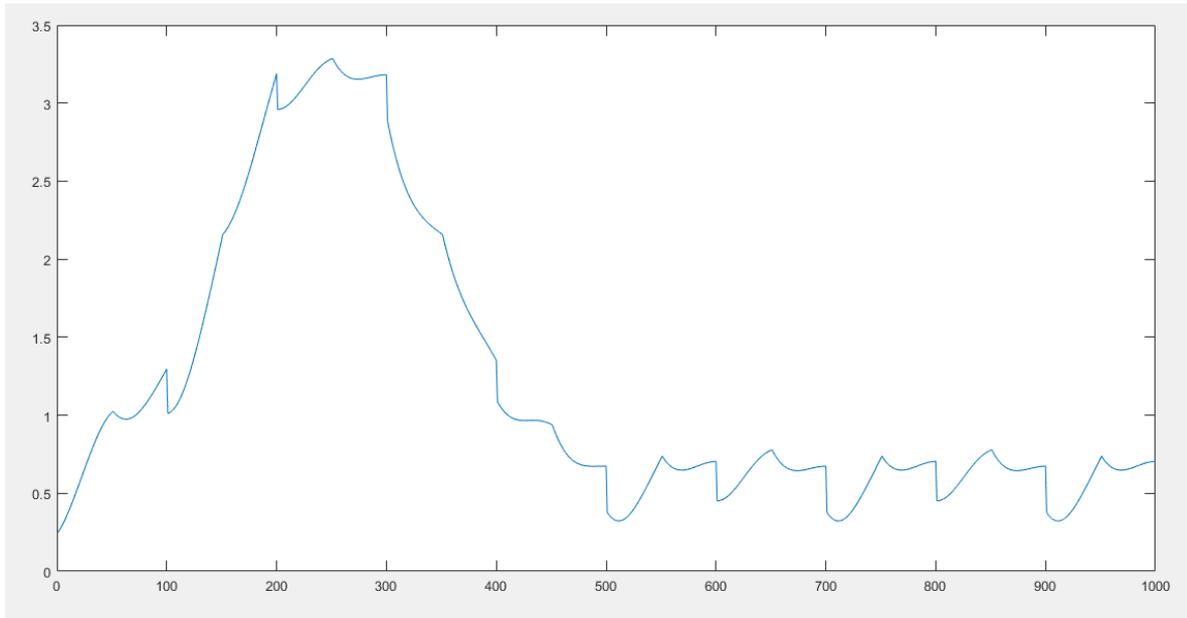


Figure 4.12 Right ankle pitch torque, generated by CPG, during support phase

The equivalent actuator torque profile for the right ankle pitch rotation during support phase generated by the DPG is shown in figure 4.13 below.

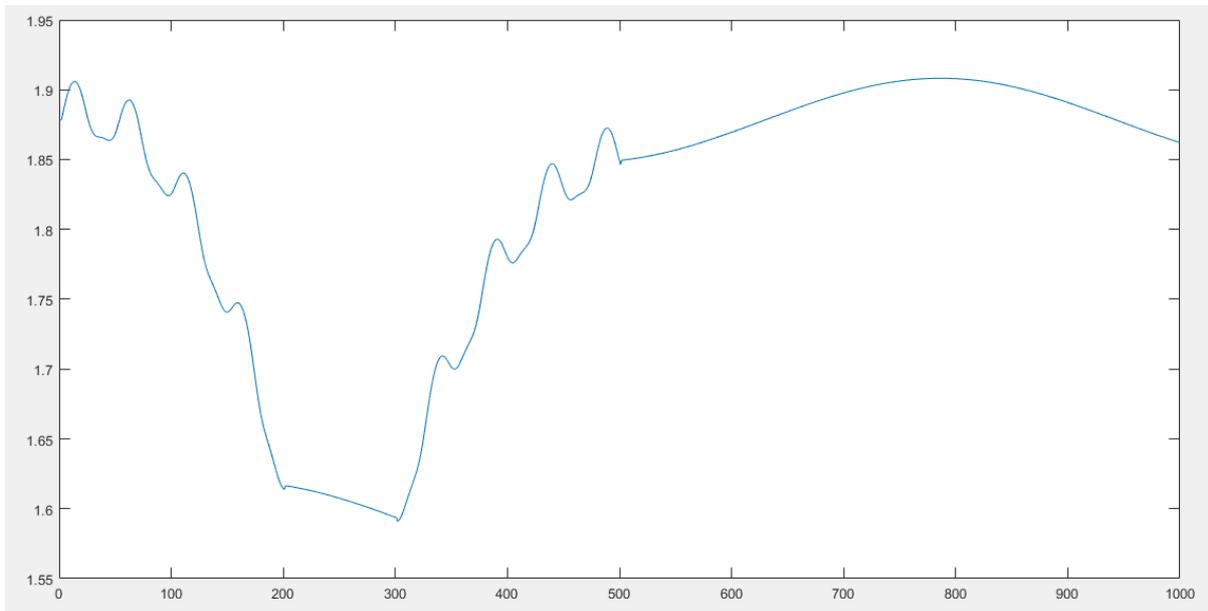


Figure 4.13 Right ankle pitch torque, generated by DPG, during support phase

Two very important observations can be drawn directly from the torque plots in figure 4.12 and 4.13. These will be elaborated further in the next chapter.

- 1- The average ankle pitch torque in case of DPG is significantly higher than in case of CPG.

2- The peak ankle pitch torque in case of DPG is much lower than CPG.

4.7 CASE V: Under-actuated 12 DOF Humanoid Robot - NUSTBOT-3

The final case that was simulated and analyzed for this study is the walk generated for the under NUSTBOT-3 model with reduced actuation. The joints for the ankle pitch and roll in both legs are un-torqued in this case. The trajectory for two steps is shown in the frames in figure 4.14 below. The via-poses start from the top right in this figure.

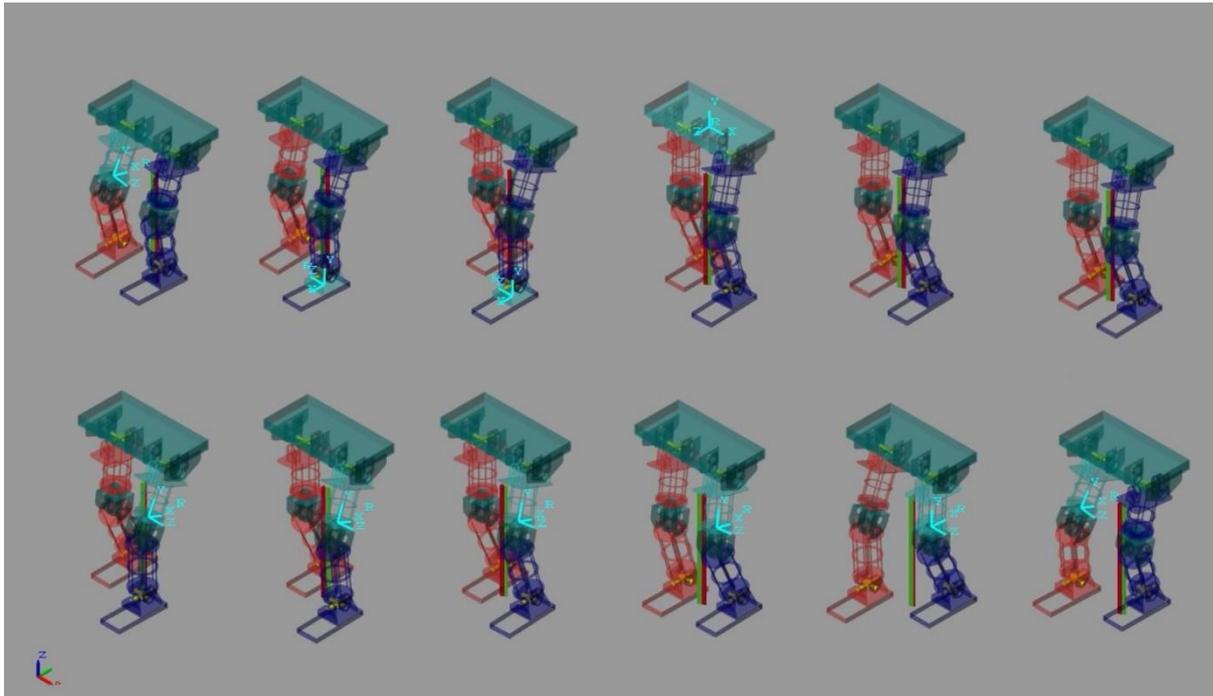


Figure 4.14 Via-Poses for the NUSTBOT-3 walking trajectory for the DPG

The motion generation results in a dynamic gait of the under-actuated NUSTBOT-3. In figure 4.14, the ZMP projection is shown by the red line whereas the COM projection is shown by the green line.

Another interesting observation here is that the motion generated by the DPG does not necessarily confine the ZMP within the contact polygon of the supporting foot. The COM and ZMP profiles are shown in the figures 4.15 and 4.16 respectively.

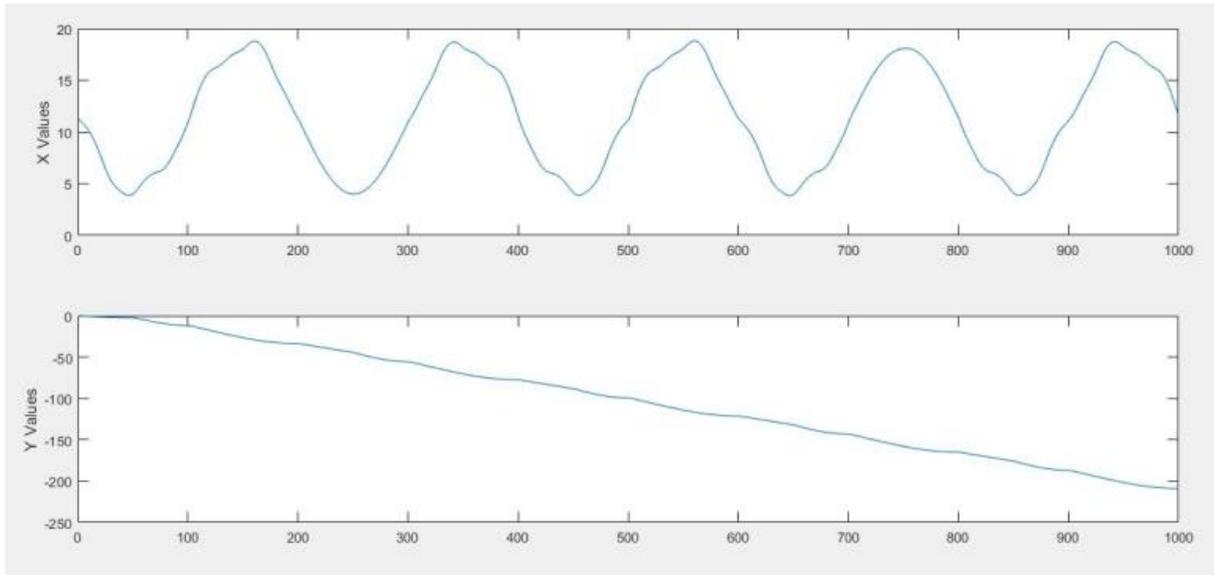


Figure 4.15 COM Profile for NUSTBOT-3 walking motion generated by a DPG

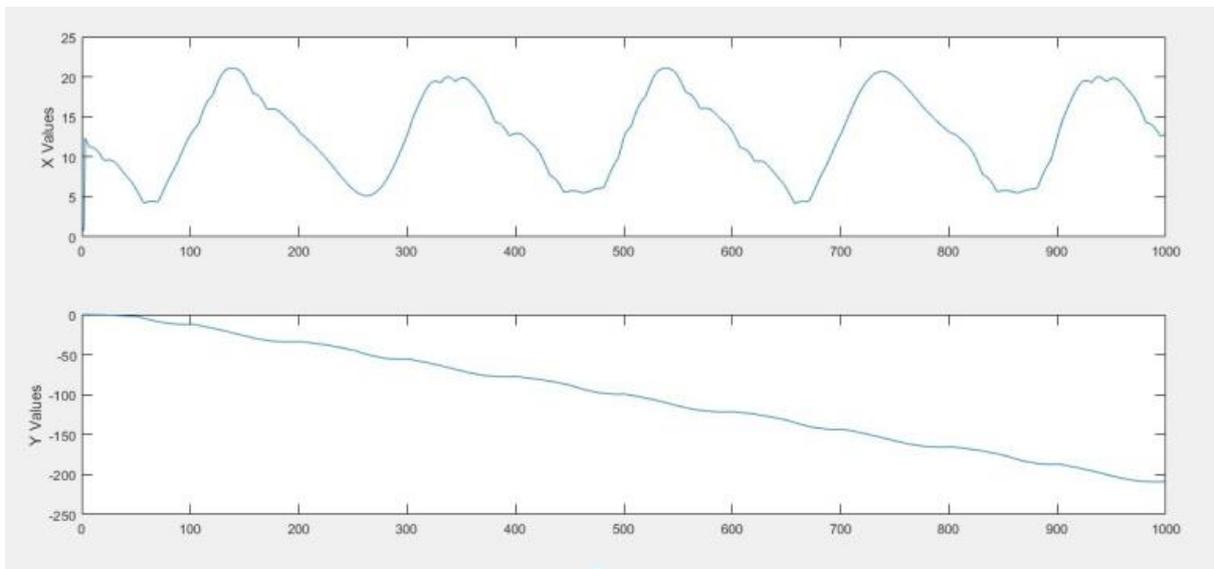


Figure 4.16 ZMP Profile for NUSTBOT-3 walking motion generated by a DPG

Notice how the DPG does not treat the ZMP-Support Polygon stability criteria, instead it uses the pose-recoverability as a metric for the selection or rejection of a trajectory spline for the robot. This will be discussed in the next chapter.

4.8 Comparison DPG & CPG

The case in which both DPG and CPG are applicable is the fully actuated model of NUSTBOT-3. Same trajectory via-points were used for both pattern generators and the result was a significant difference in generated ZMP profiles and the actuator torque requirements for the ankles. This is an important comparison result since the DPG generated a motion pattern which required less peak torque, but more average torque in order to execute the desired trajectory.

The CPG cannot solve the motion generation problem for an under-actuated robot, this is because the high gain feedback (high stiffness) approach used over-simplifies the dynamics of the robot, thereby reducing the system model significantly. The DPG on the other hand has very high computational cost, on accounts of two-fold checks of position and velocity constraints and the iterative equation of motion solver.

CPGs cannot be applied on under-actuated or reduced-actuated systems whereas the DPG, based on multi-body dynamics, can solve the motion generation problems of under-actuated or reduced-actuated systems in similar manner as for the fully-actuated systems.

These insights will be further expanded upon in the next chapter, where important conclusions will be drawn from the simulation results and analyses presented in this chapter.

4.9 Stability vs Recoverability

A final remark regarding the stability of a robot is that the pose-recoverability cannot comment about the stability of the system, it can give insights about both the choice of actuators and the type of operation the robot can or cannot perform. The stability criteria utilized by CPGs is effectively an over-estimation of the internal dynamics of the system, whereas the recoverability criteria utilized by DPG is a check influenced by the stable pose of the robot. Interesting a robot may or may not have satisfactory recoverability regardless of stability.

This means that a robot can be stable in sense of support polygons whereas it may fail the pose-recoverability check, and the opposite also holds true, that a robot which satisfies the pose-recoverability check to arrive at the stable pose, may or may not satisfy the support polygon stability criteria. The stability criteria and the recoverability criteria are two different metrics used to estimate two different quantities and will be discussed as such in the next chapter.

5 Conclusions

The previous chapter detailed the analyses & results of the DPG based motion generation of a number of robot systems. A few prominent features observed in the DPG performance are shared in this chapter.

5.1 Decentralized Pattern Generators

The decentralized pattern generator divides the motion pattern generation problem of a manipulator type robot into a problem of multi-rigid-body dynamics. The DPG based solution is for all the segments (links) of the robot instead of the representative single point like COM or ZMP as in the case of CPGs. Using the DPG approach described in this study complex motion sequences can be generated for walking robots. Also one of the advantages of DPG over CPG is the capability of solving under-actuated or reduced-actuated dynamical systems, as the multi-body dynamics allows for lesser number of external actuation forces than the joints of the manipulator of the robot.

The DPG incorporates the robot dynamics within its solution architecture whereas the CPG overrides the system dynamics to generate a high stiffness motion sequence. The DPG can execute the trajectories for complex spatial manoeuvres incorporating diverse movements like walking, sitting, crawling and climbing stairs in the same general solution, for fully-actuated and under-actuated robots alike, whereas the CPG is constrained to the solution of particular cases of walking, or stair climbing independently, one at a time.

The DPG requires a lot of computation resources as opposed to CPG and an online system will require extensive optimization of system dynamical equations or a very high end micro-processor as the on-board PC whereas the CPG can be computed during runtime. In order to assist with the readability of the algorithm the DPG variables are stored in the Matrix format which are used to reduce the number of variables, in CPG however the overall number of variables is significantly lower and with the preview controller it requires no structure for storing and computing data besides the fundamental ZMP and contact polygon (support polygon) equations and forward and inverse kinematics to drive and compute the ZMP. The DPG has twice as many continuity checks within its iterative loops as compared to ZMP preview control based CPG which has only one continuity check for the target trajectory prior to the iterative solver, this makes the CPG significantly faster than the DPG without any dedicated processing hardware for DPG computation.

5.2 Robot Design Consideration

The pose recoverability criteria of the DPG utilizes the position constraints based on the mechanical hardware of the robot and the velocity constraints based on the choice of actuators used at the joints of the robot. The pose recoverability margin varies depending upon the construction of the robot and hence a pattern generator applied on the CAD model of the robot can be used to analyze the hardware prior to mechanical fabrication.

It is possible to invert the DPG solution to give the pose recoverability of the robot in terms of a numerical solution which can be used to give significant insights on the robot hardware performance during the CAD design phase of the robot. Since the DPG is dependent upon the robot hardware constraints it can be used to optimize the robot hardware for a few specified tasks, based on the robot applications, within the workspace of the robot.

The action subspace of the actuated joints is also an important feature of the DPG, which can assist during the robot design phase. The action subspace is constrained by the type and the performance of the actuator used at the particular joint of the robot. As observed in the analysis and simulation of the fully-actuated NUSTBOT-3 model, the DPG solution is less restricting on the robot ankle joint torque whereas the CPG solution requires a very high peak torque by the ankle joint. As such since the DPG does not generate high stiffness motion patterns unless the position constraints are very stringent. Thus the DPG allows a trade-off in the mechanical design of the robot based on the constraints of the robot operation.

5.3 Target Trajectory & Motion Execution

The target trajectory given as input to the DPG in terms of via poses, is readjusted by the DPG and the spline generator can generate an over-smoothed position profile in order to satisfy the pose recoverability check for the trajectory generated by the spline generator. Although there isn't any limitation in regards to the via poses but in terms of the application of the robot the target trajectory may need to identify a further task performance in the 6D spatial vectors.

For instance if the robot is required to manipulate an object in space by interacting it with through forces the DPG may need a larger number of via poses to perform the task in a better manner, otherwise over-smoothing the target trajectory through the via poses may result in incorrect or in certain cases unacceptable trajectory tracking. A via-pose planner which generates the capture points for the robot prior to pattern generation may reduce the error in the spline generator, but this will greatly increase the computation required for the pattern generation.

The motion executed by the robot is a continuous function solved numerically through iterative computation of the DPG equations. The multi-body dynamics equations are nonlinear and can have multiple convergence points when it comes to numerical solutions (the error converges at many actuator force/torque sequences), thus the actual motion executed by the robot may have significant variation as compared to the target trajectory generated by the spline generator. A Lagrangian heuristic which optimizes the solution for either the potential energy of the system or the cost heuristic with a quadratic cost function can ensure that the motion generated by the DPG is closer to the expected solution.

5.4 Stability Criteria

The DPG does not address the robot stability in terms of the contact polygon stability utilized by the CPG with preview control of ZMP. Instead the DPG utilizes the pose recoverability criteria which is measure of the ability of the robot to recover to its stable pose after executing a motion sequence. Pose recoverability margin is a numerical Boolean function which determines if the robot segments can return to the stable pose for all segments and it is dynamically computed by the DPG in order to fine tune the velocity spline of the robot segment. The main difference between the support polygon stability criteria and the pose recoverability criteria lies in the utility. The pose recoverability criteria cannot give any insight about the robot stability in terms of the Zero-Moment Point. However the pose recoverability is a necessary condition for the controlled motion generation of the robot.

The ZMP based contact polygon criteria for stability of the robot is an over-estimation criteria (it is a sufficient condition of robot stability) and hence robots which do not satisfy the contact polygon stability can still satisfy robot pose recoverability. This is a major difference between CPG and DPG, since the DPG addresses the robot recoverability instead of the robot stability.

The robot pose recoverability and robot stability are not comparable quantities, since both have different applications, the contact polygon stability is utilized for determining the viability of the ZMP (or COM) trajectory executed by the CPG whereas the pose recoverability determines the viability of the force/torque profiles of the actuators as well as the motion profiles of the robot segments.

6 Future Work

In the closing chapter of this dissertation future recommendations are given to improve on the research work conducted during this study. Since the DPG is a new approach for solving the manipulator chains it can benefit significantly from the existing techniques applied on such robots. The major upgrades which can be developed for the DPG are briefly described in the following sections.

6.1 Pattern Generation for Online Walking

The decentralized pattern generator formulated in this study was simulated, tested and analyzed for the SimMechanics model of robots. ODE along with SimMechanics is used for dynamical simulation of multi-body systems however in order to develop a practical system which employs physical hardware of the robot an online solution must be developed which performs fast computation and converges the target trajectories fairly quickly.

One technique to facilitate the online pattern generation for walking robots is to unfold the iterative numerical equations onto a dedicated parallel processing hardware. Since the DPG is based on the structure of rigid robot segments, which remains constant by and large, the parallel processing hardware (such as an FPGA) will directly map the inverse multi-rigid-body dynamics onto the programmable array. This will significantly reduce the processing time and allow live systems to generate DPG on the run-time.

A capture points or via-pose generator such as a footstep planner based on the environment sensing and feedback system will facilitate the DPG by making the robot independent of human operation.

6.2 Generalized Pattern Generation

The DPG is by definition a generalized pattern generator, hence it incorporates the robot hardware design and actuator models within its equations. Diverse test cases for the DPG expanding onto flying, swimming and even cross terrain robots will refine the DPG process even further. However there is a catch within this process, the air-resistance is negligible for walking machines, whereas the viscosity of water for under-water and buoyancy and surface tension for surface swimming robots is an important parameter. Similarly for flying robots the air drag plays a similar role for providing the lift force/torque to the flying machine.

Thus redefining the external forces to include the environmental parameters is applicable works for ground friction in case of walking robots and air resistance in case of flying robots,

however for under-water and water surface robots a broader approach for the DPG incorporating fluid dynamics along with the multi-rigid-body dynamics may be required.

In addition this dissertation discussed robots which comprised of rigid body segments, for a flexible link robot the rigid-body dynamics is replaced by deformable or non-rigid body dynamics and the DPG can include inelastic collisions in the constraint force computation.

6.3 Stability for DPG

One of the major differences between a CPG and DPG is the stability and recoverability criterion, respectively. A stability criteria based on the pose-recoverability of the robot which determines the necessary condition for stability can be formulated which describes the ZMP stability in terms of pose-recoverability. Support polygon - ZMP based stability criteria is not a universal stability criteria, however the robot which satisfies the ZMP stability over a smooth surface is also generally stable. Thus a robot ZMP description based on the inverse multi-body dynamics and pose-recoverability can be generated which will effectively describe ZMP stability as a subset of pose-recoverability.

6.4 Hardware Experimentation

The hardware development of NUSTBOT-3 is underway at the Robotics & Intelligent Systems Engineering Laboratories and Research Center at NUST. The robot hardware is similar to the simulation model described in the test case V of this study. Hardware experimentation of the DPG on the robot hardware is planned in the future and further refinement of the DPG process will ensure a complete solution instead of a particular solution.

References

- [1] M. Vukobratovic, V. Potkonjak, and S. G. Tzafestas, "Human and Humanoid Dynamics," *J. Intell. Robot. Syst.*, vol. 41, no. 1, pp. 65–84, 2004.
- [2] R. Brooks, *Humanoid robots*, vol. 45, no. 3. 2002.
- [3] R. Zaier, *THE FUTURE OF HUMANOID ROBOTS – RESEARCH AND Edited by Riadh Zaier*. 2011.
- [4] H. Iba, *Frontiers in Evolutionary Robotics*, no. 1. 2014.
- [5] M. Eaton, *SPRINGER BRIEFS IN INTELLIGENT SYSTEMS Evolutionary Humanoid Robotics*. .
- [6] M. Hirose and K. Ogawa, "Honda humanoid robots development," *Philos. Trans. R. Soc. A Math. Phys. Eng. Sci.*, vol. 365, no. 1850, pp. 11–19, 2007.
- [7] G. Metta, G. Sandini, D. Vernon, L. Natale, and F. Nori, "The iCub humanoid robot," *Proc. 8th Work. Perform. Metrics Intell. Syst. - Permis '08*, vol. 1, p. 50, 2008.
- [8] H. Hirukawa, S. Kajita, F. Kanehiro, K. Kaneko, and T. Isozumi, "The human-size humanoid robot that can walk, lie down and get up," *Int. J. Rob. Res.*, vol. 24, no. 9, pp. 755–769, 2005.
- [9] H. Dai, A. Valenzuela, and R. Tedrake, "Whole-body Motion Planning with Simple Dynamics and Full Kinematics," *Int. Conf. Humanoid Robot.*, no. JANUARY 2014, pp. 295–302, 2014.
- [10] L. Righetti and A. J. Ijspeert, "Design methodologies for central pattern generators: an application to crawling humanoids," *Proc. Robot. Sci. Syst.*, vol. 191, pp. 191–198, 2006.
- [11] J. C. Arevalo, J. Pestana, F. Sanchez, J. F. Sarria, and E. Garcia, "Impedance Control of an Agile-Locomotion Robotic Leg," *Emerg. Trends Mob. Robot. - Proc. 13th Int. Conf. Climbing Walk. Robot. Support Technol. Mob. Mach.*, pp. 631–638, 2010.
- [12] K. Hauser, "Motion planning for legged and humanoid robots," *PhD - Stanford*, no. December, p. 151, 2008.
- [13] J.J. Kuffner, K. Nishiwaki, S. Kagami, M. Inaba, and H. Inoue, "Motion planning for humanoid robots," *Proc. Int'l Symp. Robot. Res.*, 2003.
- [14] A. J. Ijspeert, "Central pattern generators for locomotion control in animals and robots: A review," *Neural Networks*, vol. 21, no. 4, pp. 642–653, 2008.
- [15] M. H. P. Dekker, "Zero-Moment Point Method for Stable Biped Walking," *Report*, no. July, p. 62, 2009.

- [16] H. Hirukawa *et al.*, “A universal stability criterion of the foot contact of legged robots - Adios ZMP,” *Proc. - IEEE Int. Conf. Robot. Autom.*, vol. 2006, no. May, pp. 1976–1983, 2006.
- [17] H. -o. Lim and A. Takanishi, “Biped walking robots created at Waseda University: WL and WABIAN family,” *Philos. Trans. R. Soc. A Math. Phys. Eng. Sci.*, vol. 365, no. 1850, pp. 49–64, 2007.
- [18] I. I. P. Murphy, “Modeling and Control of Flapping Wing Robots,” 2013.
- [19] A. A. Paranjape, M. R. Dorothy, S. J. Chung, and K. D. Lee, “A flight mechanics-centric review of bird-scale flapping flight,” *Int. J. Aeronaut. Sp. Sci.*, vol. 13, no. 3, pp. 267–282, 2012.
- [20] M. Khan, “Quadcopter Flight Dynamics,” *Int. J. Sci. Technol. Res.*, vol. 3, no. 8, pp. 130–135, 2014.
- [21] A. Abate, “Oregon State’s C ASSIE biped - Mechanical design for dynamic locomotion,” p. 4524, 2016.
- [22] H. K. Maki, *Bioinspiration and Robotics: Walking and Climbing Robots*. 2007.
- [23] J.-A. Meyer and A. Guillot, “Biologically-inspired robots,” *Handb. Robot.*, pp. 1–38, 2008.
- [24] S. Kim, “Bio-inspired robot design for legged locomotion,” *Robotics*, 2008.
- [25] M. Wahde, *Biologically Inspired Optimization Methods: An Introduction*. 2008.
- [26] C. G. Atkeson *et al.*, “What Happened at the DARPA Robotics Challenge and Why,” *Unkown*, 2015.
- [27] D. Case, “Rules of the DRC Finals,” pp. 1–15, 2014.
- [28] R. T. Committee, “RoboCup Standard Platform League (NAO) Rule Book,” pp. 1–29, 2013.
- [29] R. Tedrake, “Underactuated Robotics: Learning, Planning, and Control for Efficient and Agile Machines,” *Course Notes MIT 6.832*, 2009.
- [30] J. Chestnutt, “Navigation Planning for Legged Robots,” no. December, p. 37, 2007.
- [31] B. Ugurlu, J. A. Saglia, N. G. Tsagarakis, and D. G. Caldwell, “Hopping at the resonance frequency: A trajectory generation technique for bipedal robots with elastic joints,” *Proc. - IEEE Int. Conf. Robot. Autom.*, pp. 1436–1443, 2012.
- [32] J. Zhang, C. Chien Chern, and S. Collins, “Torque control in legged locomotion,” *Bio-Inspired Legged Locomot. Concepts, Control Implement.*, 2016.
- [33] J. E. Pratt and B. T. Krupp, “Series elastic actuators for legged robots [5422-17],”

- Proceedings- Spie Int. Soc. Opt. Eng.*, vol. 5422, pp. 135–144, 2004.
- [34] J. Xiong *et al.*, “A Modified Gait Planning Method for Biped Robot Based on Central Pattern Generators *,” no. 51305436, pp. 1551–1555, 2015.
- [35] J. Shan, C. Junshi, and C. Jiapin, “Design of central pattern generator for humanoid robot walking based on multi-objective GA,” *Intell. Robot. Syst. 2000. (IROS 2000). Proceedings. 2000 IEEE/RSJ Int. Conf.*, vol. 3, pp. 1930–1935 vol.3, 2000.
- [36] A. G. Baydin, “Evolution of central pattern generators for the control of a five-link bipedal walking mechanism,” *Paladyn, J. Behav. Robot.*, vol. 3, no. 1, p. 10, 2012.
- [37] S. Kajita *et al.*, “Biped walking pattern generation by using preview control of zero-moment point,” *2003 IEEE Int. Conf. Robot. Autom. (Cat. No.03CH37422)*, pp. 1620–1626, 2003.
- [38] S. Kajita, H. Hirukawa, K. Harada, and K. Yokoi, *Introduction to Humanoid Robotics*, vol. 101. 2014.
- [39] J. W. Grizzle, C. H. Moog, and C. Chevallereau, “Nonlinear control of mechanical systems with an unactuated cyclic variable,” *IEEE Trans. Automat. Contr.*, vol. 50, no. 5, pp. 559–576, 2005.
- [40] M. N. ElBsat, “Finite-time control and estimation of nonlinear systems with disturbance attenuation,” no. August, 2012.
- [41] J.-J. Slotine and W. Li, *Applied Nonlinear Control*. 1991.
- [42] M. W. Spong, “Partial feedback linearization of underactuated mechanical systems,” *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, vol. 1, pp. 314–321.
- [43] M. J. Powell, W. L. Ma, E. R. Ambrose, and A. D. Ames, “Mechanics-based design of underactuated robotic walking gaits: Initial experimental realization,” *IEEE-RAS Int. Conf. Humanoid Robot.*, pp. 981–986, 2016.
- [44] M. Gajamohan, M. Muehlebach, T. Widmer, and R. D’Andrea, “The Cubli: A Reaction Wheel Based 3D Inverted Pendulum,” *Proc. Eur. Control Conf.*, pp. 268–274, 2013.
- [45] M. Muehlebach, G. Mohanarajah, and R. D’Andrea, “Nonlinear Analysis and Control of a Reaction Wheel-based 3D Inverted Pendulum: BT - Proc. Conference on Decision and Control,” pp. 1283–1288, 2013.
- [46] M. Gajamohan, M. Merz, I. Thommen, and R. D’Andrea, “The Cubli: A cube that can jump up and balance,” *IEEE Int. Conf. Intell. Robot. Syst.*, pp. 3722–3727, 2012.
- [47] H.-K. Shin and B. K. Kim, “Energy-Efficient Gait Planning and Control for Biped

- Robot Utilizing the Allowable ZMP region,” *Ieee Trans. Robot.*, vol. 30, no. 4, pp. 986–933, 2014.
- [48] M. VUKOBRATOVIĆ and B. BOROVIAC, “Zero-Moment Point — Thirty Five Years of Its Life,” *Int. J. Humanoid Robot.*, vol. 1, no. 1, pp. 157–173, 2004.
- [49] W. Suleiman, F. Kanehiro, K. Miura, and E. Yoshida, “Enhancing zero moment point-based control model: System identification approach,” *Adv. Robot.*, vol. 25, no. 3, pp. 427–446, 2011.
- [50] Z. A. Shah, “NUSTBOT-3.” [Online]. Available: <http://rise.smme.nust.edu.pk/projects/NUSTBOT-3/>. [Accessed: 13-Aug-2017].
- [51] R. Featherstone, *Rigid Body Dynamics Algorithms*. .
- [52] R. Featherstone and D. Stewart, “Notes to Accompany the Slides,” pp. 1–4, 2008.
- [53] *Springer handbook of robotics*, vol. 46, no. 6. 2009.
- [54] S. Traversaro and A. Saccon, “Multibody Dynamics Notation,” 2016.
- [55] H. Lim and A. Takanishi, “A Biped Humanoid Robot for Human-robot Coexistence,” 1996.
- [56] J. Pratt, C. M. Chew, A. Torres, P. Dilworth, and G. Pratt, “Virtual model control: an intuitive approach for bipedal locomotion,” *Int. J. Rob. Res.*, vol. 20, no. 2, pp. 129–143, 2001.
- [57] J. Buchli and a J. Ijspeert, “Distributed Central Pattern Generator Model for Robotics Application Based on Phase Sensitivity Analysis,” *Biol. Inspired Approaches to Adv. Inf. Technol. First Int. Work. {BioADIT 2004}*, vol. 3141, no. c, pp. 333–349, 2004.
- [58] K. Kaneko *et al.*, “Design of prototype humanoid robotics platform for HRP,” *IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, vol. 3, no. October, pp. 2431–2436, 2002.
- [59] S. Shirata, A. Konno, and M. Uchiyama, “Development of a light-weight biped humanoid robot,” *Proc. 2004 IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, pp. 148–153, 2004.
- [60] S. H. Collins and A. Ruina, “A Three-Dimensional Passive Dynamic Walking Robot with Two Legs and Knees,” *Int. J. Rob. Res.*, vol. 20, no. 7, pp. 607–615, 2001.
- [61] “Robots at Sano, Nagoya Institute of Technology.” [Online]. Available: <http://drei.mech.nitech.ac.jp/~sano/theme.html>. [Accessed: 13-Aug-2017].
- [62] M. Raibert, “Dynamic Legged Robots for Rough Terrain,” p. 1995.
- [63] M. H. Raibert, “Hopping in Legged Systems-Modeling and Simulation for the Two-Dimensional One-Legged Case,” *IEEE Trans. Syst. Man Cybern.*, vol. 14, no. 3, pp.

- 451–463, 1984.
- [64] J. G. Cham and M. R. Cutkosky, “Dynamic Stability of Open-Loop Hopping,” vol. 129, no. May, pp. 275–284, 2007.
- [65] D. Wooden, M. Malchano, K. Blankespoor, A. Howard, A. A. Rizzi, and M. Raibert, “Autonomous Navigation for BigDog,” pp. 4736–4741, 2010.
- [66] J. K. Hodgins and H. Raibert, “Adjusting Step Length for Rough Terrain Locomotion,” vol. 7, no. 3, 1991.
- [67] G. Aguirre-ollinger, J. E. Colgate, and M. A. Peshkin, “A 1-DOF Assistive Exoskeleton with Inertia Compensation : Effects on the Agility of Leg Swing Motion,” pp. 1–44.
- [68] J. W. Grizzle, J. Hurst, B. Morris, H. Park, and K. Sreenath, “MABEL , A New Robotic Bipedal Walker and Runner,” in *American Control Conference*, 2009, pp. 2030–2036.
- [69] A. Ramezani, J. W. Hurst, K. A. Hamed, and J. W. Grizzle, “Performance Analysis and Feedback Control of ATRIAS , A Three-Dimensional Bipedal Robot,” *J. Dyn. Syst. Meas. Control*, vol. 136, no. March 2014, 2016.
- [70] K. Yokoi, F. Kanehiro, K. Kaneko, and K. Fujiwara, “Experimental Study of Biped Locomotion of Humanoid Robot HRP-1S,” pp. 1–10.
- [71] K. Kaneko *et al.*, “Humanoid Robot HRP-2,” in *International Conference on Robotics and Automation*, 2004, no. April, pp. 1083–1090.
- [72] K. Kaneko, K. Harada, F. Kanehiro, G. Miyamori, and K. Akachi, “Humanoid Robot HRP-3,” in *International Conference on Intelligent Robots and Systems*, 2008, no. April, pp. 2471–2478.
- [73] D. V. Knudson, *Fundamentals of Biomechanics*, 2nd ed. New York, New York: Springer US, 2007.
- [74] S. Kajita and B. Espiau, “Legged Robots,” *Springer Handb. Robot.*, pp. 361–389, 2008.
- [75] E. Garcia and P. Gonzalez de Santos, “On the improvement of walking performance in natural environments by a compliant adaptive gait,” *IEEE Trans. Robot.*, vol. 22, no. 6, pp. 1240–1253, 2006.
- [76] E. Garcia, J. Estremera, and P. G. De Santos, “A Comparative Study of Stability Margins for Walking Machines.”
- [77] van P. W. M. Zutven, *Control and identification of bipedal humanoid robots: stability*

- analysis and experiments*. 2014.
- [78] J. Strom, Slavov George, and Chown Eric, “Omnidirectional walking using ZMP and preview control for the NAO humanoid robot,” *Rob. 2009 Robot Soccer World Cup XIII, ser. Lect. Notes Comput. Sci.*, 2010.
- [79] J. Park and Y. Youm, “General ZMP preview control for bipedal walking,” *Proc. - IEEE Int. Conf. Robot. Autom.*, no. April, pp. 2682–2687, 2007.
- [80] S. Kajita *et al.*, “Biped walking pattern generator allowing auxiliary ZMP control,” *IEEE Int. Conf. Intell. Robot. Syst.*, vol. 2, pp. 2993–2999, 2006.
- [81] J. Melorose, R. Perroy, and S. Careas, *Nonlinear Control of Dynamics Network*, vol. 1. 2015.
- [82] H. Yabuno, K. Goto, and N. Aoshima, “Swing-up and stabilization of an underactuated manipulator without state feedback of free joint,” *IEEE Trans. Robot. Autom.*, vol. 20, no. 2, pp. 359–365, 2004.
- [83] G. Vilmart, “Rigid Body Dynamics,” *Offshore Hydromechanics*, no. January, pp. 1–17, 2001.
- [84] R. Featherstone and R. Featherstone, “Spatial Vector Algebra,” *Rigid Body Dyn. Algorithms*, pp. 7–38, 2008.
- [85] Y. H. Amengonu and Y. P. Kakad, “Dynamics and control of constrained multibody systems modeled with Maggi’s equation: Application to differential mobile robots part i,” *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 65, no. 1, 2014.
- [86] K. S. Shankar and T. L. Harmon, “Introduction to Robotics,” *IEEE Expert*, vol. 1, no. 2, pp. 108–108, 1986.
- [87] D. K. Pai, U. M. Ascher, and P. G. Kry, “Forward dynamics algorithms for multibody chains and contact,” *Proc. 2000 ICRA. Millenn. Conf. IEEE Int. Conf. Robot. Autom. Symp. Proc. (Cat. No.00CH37065)*, vol. 1, pp. 857–863, 2000.
- [88] J. Lieh, “Computer-aided modeling and control of multibody systems for robotics application,” *Robot. Autom. 1991. Proceedings., 1991 IEEE Int. Conf.*, no. April, pp. 2385–2390, 1991.
- [89] M. Quasem, S. Uhlar, and P. Betsch, “Inverse dynamics of underactuated multibody systems,” vol. 120, no. 5, pp. 5–6, 2009.
- [90] A. Jain, “Unified formulation of dynamics for serial rigid multibody systems,” *J. Guid. Control. Dyn.*, vol. 14, no. 3, pp. 531–542, 1991.
- [91] G. Jr, B. Robert, and S. Olivier, “Inverse dynamics of serial and parallel underactuated

- multibody systems using a DAE optimal control approach,” no. July, pp. 359–376, 2013.
- [92] D. Baraff, “Fast contact force computation for nonpenetrating rigid bodies,” *Proc. 21st Annu. Conf. Comput. Graph. Interact. Tech. - SIGGRAPH '94*, no. May, pp. 23–34, 1994.
- [93] P. Flores and H. M. Lankarani, *Solid Mechanics and Its Applications Contact Force Models for Multibody Dynamics*, no. January. 2016.
- [94] M. Machado, P. Flores, D. Dopico, and J. Cuadrado, “Dynamic response of multibody systems with 3D contact-impact events : influence of the contact force model Contact-impact force models,” pp. 1–2, 2012.
- [95] P. Song, “Modeling, Analysis and Simulation of Multibody Systems with Contact and Friction,” pp. 1–164, 2002.
- [96] M. J. Powell and A. D. Ames, “Mechanics-based control of underactuated 3D robotic walking: Dynamic gait generation under torque constraints,” *IEEE Int. Conf. Intell. Robot. Syst.*, vol. 2016–Novem, pp. 555–560, 2016.
- [97] D. Auckly, L. Kapitanski, and W. White, “Control of nonlinear underactuated systems,” *Commun. Pure Appl. Math.*, vol. 53, no. 3, pp. 354–369, 2000.
- [98] “Only in Japan” 2014 [Online]. Available: <http://onlyinjapan.tv/japans-amazing-robots-hondas-asimo-uni-cub-and-the-robot-culture/> [Accessed: 13-Aug-2017]
- [99] “DARPA Robotics Challenge” 2015 [Online]. Available: <https://www.darpa.mil/program/darpa-robotics-challenge> [Accessed: 13-Aug-2017]
- [100] “Cubli” [Online]. Available: <http://www.idsc.ethz.ch/research-dandrea/research-projects/cubli.html> [Accessed: 13-Aug-2017]
- [101] “Cornell Robotics Research, Andy Ruina” [Online]. Available: <http://ruina.tam.cornell.edu/research/topics/robots/> [Accessed: 13-Aug-2017]
- [102] “3D One-Leg Hopper - MIT, Leg Lab” [Online]. Available: http://www.ai.mit.edu/projects/leglab/robots/3D_hopper/3D_hopper.html [Accessed: 13-Aug-2017]
- [103] “PETMAN by Boston Dynamics” [Online]. Available: <http://www.extremetech.com/extreme/102545-petman-the-us-armys-latest-lifelike-robotic-recruit> [Accessed: 13-Aug-2017]

- [104] “Big Dog by Boston Dynamics” [Online]. Available:
<https://www.engadget.com/2013/12/14/google-bought-boston-dynamics-its-over-for-humans/> [Accessed: 13-Aug-2017]
- [105] “Atrias by Oregon State University” [Online]. Available:
<http://mime.oregonstate.edu/research/drl/atrias/> [Accessed: 13-Aug-2017]
- [106] “MABEL by University of Michigan” [Online]. Available:
<https://spectrum.ieee.org/automaton/robotics/humanoids/mabel-bipedal-robot-is-now-fast-enough-to-catch-you> [Accessed: 13-Aug-2017]
- [107] “Cassie by Agility Robotics” [Online]. Available:
<https://spectrum.ieee.org/automaton/robotics/industrial-robots/agility-robotics-introduces-cassie-a-dynamic-and-talented-robot-delivery-ostrich> [Accessed: 13-Aug-2017]
- [108] “HRP Robot Series” [Online]. Available: <http://www.roboticstoday.com/robots/robots-a-to-z/h> [Accessed: 13-Aug-2017]