Negotiating Large Obstacles with a Humanoid Robot via Multi-Contact Motion Planning

Pavan Kanajar

Supervisors:
Dr. Petar Kormushev
Prof. Darwin Caldwell

A thesis submitted in partial fulfillment
of the requirements for the degree of

Doctor of Philosophy
in
Bioengineering and Robotics

Department of Advanced Robotics
Istituto Italiano Di Tecnologia
&
Department of Informatics, Bioengineering, Robotics and Systems Engineering
Università Di Genova

February 2018

Abstract

Incremental progress in humanoid robot locomotion over the years has achieved essential capabilities such as navigation over flat or uneven terrain, stepping over small obstacles and climbing stairs. However, the locomotion research has mostly been limited to using only bipedal gait and only foot contacts with the environment, using the upper body for balancing without considering additional external contacts. As a result, challenging locomotion tasks like climbing over large obstacles relative to the size of the robot have remained unsolved. In this paper, we address this class of open problems with an approach based on multi-contact motion planning, guided by physical human demonstrations. Our goal is to make humanoid locomotion problem more tractable by taking advantage of objects in the surrounding environment instead of avoiding them. We propose a multi-contact motion planning algorithm for humanoid robot locomotion which exploits the multi-contacts at the upper and lower body limbs. We propose a contact stability measure, which simplifies the contact search from demonstration and contact transition motion generation for the multi-contact motion planning algorithm. The algorithm uses the whole-body motions generated via Quadratic Programming (QP) based solver methods. The multi-contact motion planning algorithm is applied for a challenging task of climbing over a relatively larger obstacle compared to the robot. We validate our planning approach with simulations and experiments for climbing over a large wooden obstacle with COMAN, which is a complaint humanoid robot with 23 degrees of freedom (DOF). We also propose a generalization method, the "Policy-Contraction Learning Method" to extend the algorithm for generating new multi-contact plans for our multi-contact motion planner, that can adapt to changes in the environment. The method learns a general policy and the multi-contact behavior from the human demonstrations, for generating new multi-contact plans for the obstacle-negotiation.

Acknowledgements

I would like to thank my supervising professors Dr. Petar Kormushev and Prof. Darwin Caldwell for constantly motivating and encouraging me, and also for their invaluable advice during my doctoral studies. I would also like to thank Dr. Nikolaos Tsagarakis and Dr. Roy Featherstone for taking interest in my research and helping me to address the challenges throughout my Ph.D. Many thanks to Dr. Serena Ivaldi and Dr. Emmanuel Spyrakos for showing interest and taking time to review my thesis and also serve on the thesis committee. Many thanks to Dr. James Kuffner and Dr. Atsuo Takanishi for providing permissions to reuse their experimental images and robot pictures in this thesis. Thankful to Enrico Mingo, Alessio Rocchi, Corrado Pavan, Alessandro Settimi and Luca Muratore for helping me with the robot hardware and associated software used for the robotic systems at ADVR. Hugely thankful to Gianluca Pane for helping me plan and set up the physical experiments in a short time. Also, thanks to Phil Edward Hudson for helping me fix the robot numerous times and providing useful tips to run the experiments and Riccardo Sepe for providing the necessary computer accessories. Extending my heartfelt thanks to Przemek Kryczka, Dimitrios Kanoulas, Rajesh Subburaman, João Silvério, Leonel Rozo, Martijn Zeestraten, Jinoh Lee, Ferdinando Cannella, Francesca Negrello, Yangwei You, Brian Delhaisse, Domingo Esteban, Songyan Xin, Fares Abu Dakka, Rajeshkumar Mupparapu, Nahian Rahman, Stefano Toxiri, Vishnu Dev Amara, Sandeep Keshavan, Subbu, Manish, Prachi, Chengxu Zhou, Ioannis Sarakoglou, Nikhil Deshpande, Alperen Acemoglu, Lorenzo Saccares, Stefano Toxiri, Wesley Roozing, Yifu Gao, Zhuoqi Cheng, Mariacarla Memeo, Maria Laura DAngelo, Milan, Anh Nguyen and all other friends who helped to make this journey more fruitful. Thanks to Silvia Ivaldi for arranging the conference and summer school trips and Floriana Sardi for supporting in administrative tasks. I would also like to mention my friends at UT Arlington, i.e., Isura Ranatunga, Isao Saito, Avinash Bharadwaj, Harsha Doreswamy, Sujay Bagi, Abhishek Bangalore, Bharat Venugopal and Kiran Bhatta who helped me in many ways to get here. Finally, I would like to express my deepest gratitude to my dear wife, Akshata who has encouraged me throughout this process and motivated me towards achieving this Ph.D. I am also grateful to my mother, father, and sister for their sacrifice, encouragement, and patience during all this time. I also thank all my teachers and professors who have helped me throughout my career.

To my wife, Akshata

Contents

CHAPTER 1

Introduction

Humanoid robots can perform human-like locomotion and manipulation tasks to assist humans. The goal of this thesis is to develop advanced planning algorithms to improve the multi-contact motion planning for locomotion of humanoid robots in challenging environments.

1.1 Motivation

Every night when I get out of bed in the dark, I use my arms to quickly feel the walls to support myself and walk towards the refrigerator to fetch myself a glass of water. We make numerous such supporting contacts during our day-to-day tasks with the surrounding objects in our environment without even thinking about the process. For instance, when we trip and lose balance during walking we make a few more supporting contacts with our arms to quickly grab on to any near-by objects to avoid potential falls. We regularly see that by making additional contacts, humans manage to move around their surrounding environment with safety at least most of the times.

Having robotic systems that can precisely display such advanced levels of contact planning, would be an essential milestone in bringing the humanoid robots to every home. Like for example, when a humanoid robot trips while walking we would expect it to prevent the fall by planning to and making additional contacts with its arms to balance itself in the process. Another good example of such planning for multiple contacts is usually seen on a crowded bus for transport at an airport, where numerous people who are standing in the bus take the support of the handrails or hand-straps in the bus to stabilize themselves against any driving disturbances. Similarly, in wall climbing sports, people make support contacts with their arms and feet on the climbing holds to accomplish the climbing task. We can see that the same ability helps us in rock climbing or any sort of adventurous sports as in Figure 1.1. In disaster sites, where stable footholds are not present, it is essential for the humanoids to have the capability to plan for multiple contacts for locomotion in such environments. Human beings take due advantage of living in the familiar environment by making use of the objects in our surroundings while planning for the movements around them.

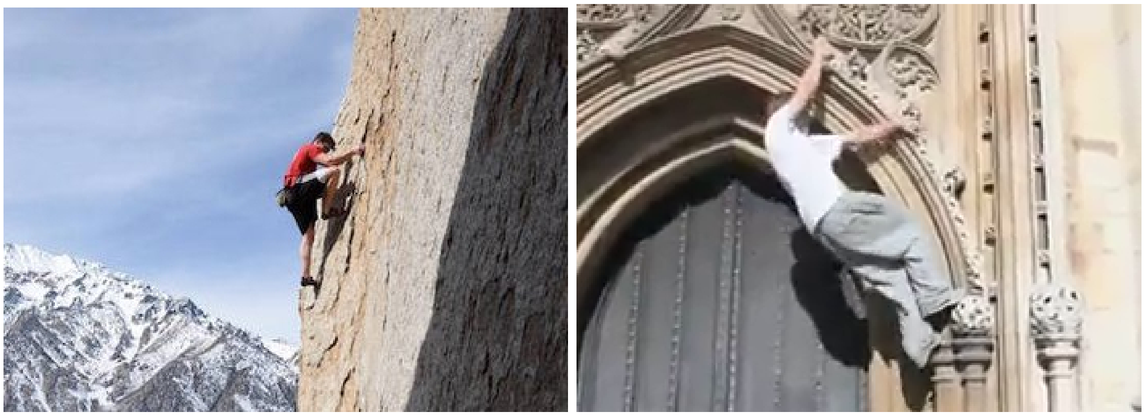

Fig. 1.1: A person climbing a rocky mountain without any gears is shown on left side. Another person is shown clinging on to the church wall decorations in the right picture.

Imagine if the robots could also do the same with all the comparable human capabilities it now has. With the advancements in technology, the present humanoid robots' sensing and actuating capabilities are comparable to that of humans, yet they do not portray the same levels of multiple contact planning and motion behaviors. I wish that robots could one day achieve similar degrees of multi-contact behaviors, and plan for motions in challenging environments, eventually allowing these robots to be incorporated into human-centered environments seamlessly, from their usual restrictive lab settings.

Devising a planning algorithm that can accomplish a sophisticated level of Multi-Contact Motion Planning for humanoid robots, as described above, is complex and nontrivial. For decades, the research in locomotion for humanoid robots has mainly focused on motion pattern generators for bipedal walking gaits (Nagasaka et al. (1999), Kajita et al. (2003a), Kryczka et al. (2015)). Using state-of-the-art bipedal motion pattern generators, humanoid robots can navigate across complex scenarios, including obstacle negotiation mainly through stepping-over methods or avoiding obstacles altogether by circumnavigating around them and also climbing stairs. However, these approaches avoid any additional contact with the surrounding objects in the environment and the motion planners for humanoid robots explicitly generate planned paths that incorporate safety distance margins to avoid such collisions. Due to this, the humanoid robots tend to fall short of the many prerequisites needed to execute tasks as well as humans do.

1.2 Objectives and Approach

This thesis aims at developing a multi-contact motion planning algorithm to generate multi-contact plans that can adapt its behavior to changing scenarios. Specifically, our aim is to develop an all-encompassing multi-contact motion planning algorithm that fulfills the following desiderata: 1) It should be easy to specify a goal, 2) It must be able to adapt the contact-plans to the environment changes, 3) It must be able to incorporate new contact plans and must select an optimal plan for the given scenario, 4) It must be able to make use of the static objects via multi-contacts in the environment.

The first criterion is essential as it provides an easy and intuitive means to provide goals for the robots. In addition, it enables the system to generate motions that mimic human behavior (and also look natural for human interaction). The other two criteria help the planner to adapt the planning scheme conceived in the initial stage, according to the changes in the environment and finally optimizes it to the robot. The final criteria is central to our multi-contact approach, i.e. making use of the surrounding objects effectively to aid humanoid locomotion. This brings us to our hypothesis:

> *Hypothesis: Humanoid robots can negotiate large obstacles by planning for multiple contacts on the obstacles. These multi-contacts can increase the overall support region for the humanoid robot thereby improving its ability to navigate in challenging environments.*

The main scope of this thesis is to develop a planning algorithm that fulfills the objective of multi-contact motion planning for a humanoid robot, especially for the case of negotiating larger obstacles in its surrounding environment. Eventually to improve the state-of-the-art in locomotion by allowing multiple contacts not only limited at feet but also over elbows, hands, and knees of the humanoid robot. We also show an extended application of the multi-contact motion planning algorithm. The objectives of this thesis are the following:

1. Obtain an initial solution for negotiating obstacle with multi-contacts which captures the requirements and challenges of the task.

2. Develop a multi-contact motion planning method using this initial solution and adapting the solution to demonstrate the negotiation of a large obstacle via multi-contact on a humanoid robot.

3. Generalize the multi-contact motion plans through the “Policy Contraction Learning” method and adapting the contact plans to different experimental scenarios.

4. Generate new multi-contact plans through the “Multi-contact Replanning” algorithm for generating new contact-plans for changes in the dimension of the obstacle.

1.3 Chapter Summary

This thesis is organized as follows:

Chapter 2: Background and Research
In this chapter, we provide a brief introduction to the state-of-the-art robots. We discuss the challenges faced by the humanoids robot at the DARPA robotics challenge and how this thesis addresses some of those problems with the multi-contacts planning approach. We also review the state-of-the-art planning algorithms, whole-body control and multi-contacts for the humanoid robots.

Chapter 3: Initial Solutions for Negotiating Obstacles
In this chapter, we introduce the research problem addressed by this thesis. We discuss the reasons for obtaining an initial solution for the problem. Several methods like kinesthetic teaching, tele-operation and sampling based planners were explored to get a preliminary solution for the multi-contact motion task. We show that human demonstration is a good way to obtain initial solution for humanoid robot for negotiating obstacle. We explore two methods of obtaining the demonstration in the chapter, and select the best demonstrated method.

Chapter 4: Multi-Contact Motion Planning
In this chapter, we present the multi-contact planning algorithm for humanoid robots from the demonstrations of the task. We show that the planning algorithm can formulate the multi-contact tasks directly from the human demonstration. We also introduce a contact stability metric for the contact optimization and generation of the transition postures, to simplify the multi-contact planning in humanoid locomotion. The motion planning algorithm also addresses the collision-free planning for the humanoid robots. The multi-contact motion planning is validated in both simulation and experiments for climbing over an obstacle task.

Chapter 5: Generalization of Multi-Contact Motion Planning
In this chapter, we present two methods to adapt and generalize the multi-contact plans from the demonstrations. The first method, "Policy-Contraction Learning" method builds a general policy from the contact plan of the demonstrations. It is shown that the learned general policy can generate several multi-contact steps necessary for the task of overcoming an obstacle. We also introduce notion of costs, to evaluate the general policy the multi-contact task. We propose another method, "Mutli-Contact Replanning" method, which addresses the multi-contact planning to generate new contact plan for changes in the obstacle dimensions.

Chapter 6: Conclusion
This chapter concludes the thesis by summarizing its technical contributions, discussing its assumptions and limitations, and providing possible research extensions.

1.4 Publications

Conference Papers

P. Kanajar, D. G. Caldwell and P. Kormushev, Climbing over Large Obstacles with a Humanoid Robot via Multi-Contact Motion Planning, 2017 26th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN), Lisbon, 2017.

Workshop Papers

P. Kanajar, D. G. Caldwell and P. Kormushev, Obstacle Climbing with a Humanoid Robot guided by Human Demonstrations, In electronic proceedings of the Workshop on Human-Humanoid collaboration: the next industrial revolution, International Conference on Humanoid Robots, Birmingham, 2017.

Publication note

Most of the material presented in Chapters 4 and 5 have been published or under submission in peer-reviewed conference proceedings and scientific journals.

CHAPTER 2

Background and Research

Humanoid robots are designed based on the general structure of humans beings, enabling such robots to walk on two legs and also have an upper body consisting of two arms and a head with sensory inputs. Such advanced robots which are designed based on similar human embodiments, are suitable for incorporation into our environments, since these have been designed to facilitate our own mobility with flat floors, stairs or ramps, and corridors.

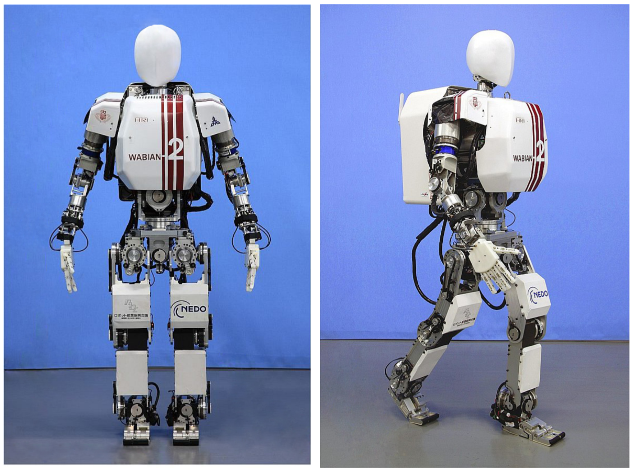

Fig. 2.1: Humanoid robot WABIAN-2R[1] front picture (left). WABIAN-2R robot walks with heel-contact toe-off motion (right).

Also, their dimensions are tailor-made to enable ease of human access to the environment. The presence of similar human characteristics in the humanoid robots makes them better suited for the human-centered environments. One of the earliest humanoid robots was prototyped by Ichiro Kato et al. at the Waseda University. Their latest humanoid robot WABIAN-2R with 41 DOFs, which is 150 cm in height and 64 kg in weight is shown in Figure 2.1. By exploiting

[1] Courtesy of Atsuo Takanishi Lab., Waseda Univ., Tokyo, Japan.

the research on the human gait analysis and redundancy in the foot mechanism, the robot performs stretched-knee walking with heel-contact and toe-off motion, as shown in Figure 2.1.

The field of humanoid robotics was drastically changed when Honda Co. publicly revealed their first humanoid robot P2 in 1996, that can walk on biped legs with sufficient stability. Honda had then challenged the robotics community by performing these astonishing feats on a full humanoid robot. Their latest humanoid robot ASIMO has 34 DOFs, is 48 kg in weight and 130 cm in height, is shown in Figure 2.2. ASIMO can stably walk at speeds of around 3 km per hour and also run, achieving speeds up-to 9 km per hour. ASIMO has also demonstrated stair ascending and descending capabilities. It also can coordinate the upper body movements to perform realistic gaits and balance during locomotion. Honda's ASIMO robot performing running and climbing stairs at a special presentation in Ontario Science Centre are shown in the picture below.

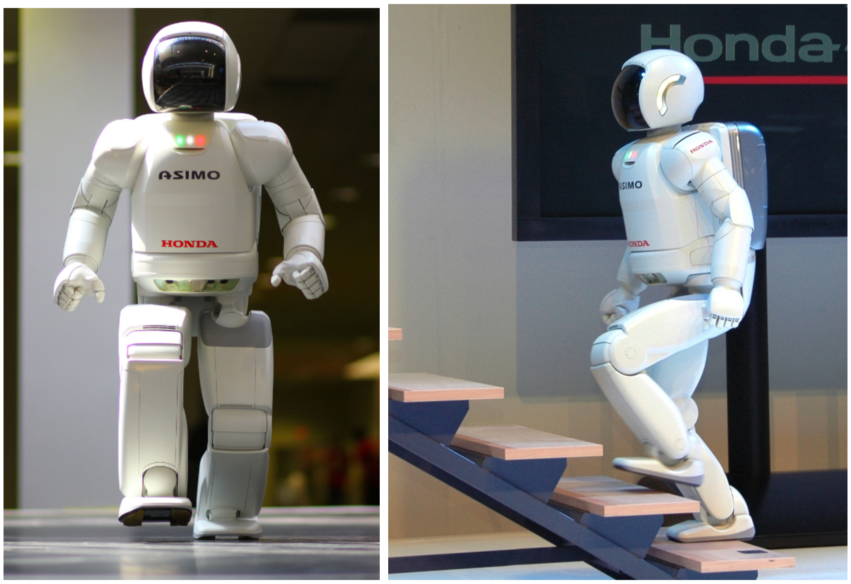

Fig. 2.2: The humanoid robot ASIMO by Honda Co. is shown in running stance. On the right side the robot ASIMO is climbing the stairs.

Research in humanoid robot locomotion has mainly focused on periodic walking pattern generators for humanoid walking gaits. For dynamically stable humanoid robot walking, Zero Moment Point (ZMP) is widely used to determine the stability (Vukobratovi et al. (1990)). It guarantees stability when the computed ZMP is inside the support polygon. The advanced approaches used for the realization of biped walking in humanoid robots are off-line pattern generation with on-line feedback compensation or on-line pattern generation with on-line feedback control (Kajita et al. (2003b)).

The off-line pattern generation with on-line feedback compensation does overcome the stability and the robustness problems. This approach uses ZMP based walking patterns and compensates them to keep the balance by feedback control. It can increase the walking stability of the robot by reducing the instability factors induced by un-modeled dynamics, ground conditions, etc., controller, although the complexity of stable walking pattern generation still remains. The enticing feature of the described methods, is that they use an oversimplified linear inverted pendulum model, which allows for a dissociation of the ZMP along the various axes. The inverted pendulum model is a simple model for generating walking gaits to mimic the human walking gaits. In the on-line pattern generation, on-line feedback approach, the walking pattern is obtained by kinematically generating the position commands for the joints (Park et al. (2006)). It is constructed by observing the humans behavior, and the walking stability is controlled by adapting the feedback controllers (Kajita et al. (2003a), Kagami et al. (2002)).

Now, with the state-of-the-art bipedal motion pattern generators, it is possible to navigate complex environment scenarios with obstacles, uneven terrains (rubbles) and stairs. The humanoid robot can avoid the obstacles by either circumnavigating around them as shown in (Stilman et al. (2006)) or by stepping over them up-to obstacle height of 15cm (Guan et al. (2006), Verrelst et al. (2006)). Although there have been tremendous advancements in pattern generators and ZMP controllers for navigating in environments for humanoid robots, uncertainty in the environment can invalidate the assumption made for simplified controllers causing instabilities in humanoid robot.

2.1 DARPA Robotics Challenge for Disaster Response

Humanoid robot locomotion has been challenging, but it is very relevant for performing human-like tasks in human-centered environments. These include walking, climbing stairs, stepping over obstacles and walking across uneven terrain without the constant fear of falling over, especially in disaster-like situations, where human intervention is prohibitive due to potentially hazardous and fatal reasons.

The Fukushima Daiichi nuclear disaster was caused primarily by the tsunami following the Tōhoku earthquake on 11 March 2011. During the Fukushima nuclear plant meltdown, even the most advanced military robots had difficulties intervening and failed to contain the fallout from the nuclear disaster in a timely fashion, which grievously impacted thousands of local residents and plant workers. In response, DARPA had called for a robotics challenge for the disaster response by assimilating the similar conditions and challenges in the disaster.

Several tasks relevant to the disaster scenario were identified and many humanoid robots from around the world took part in the finals challenge. The challenge objective was exploring the robot's ability to contain a contaminated nuclear reactor, where the robot would have to conquer not only piles of rubble in the facility, but also be able to open doors, turn valves and climb stairs. The graphic in Figure 2.3 demonstrates the tasks the robots must successfully

[2] Courtesy of IEEE Spectrum.

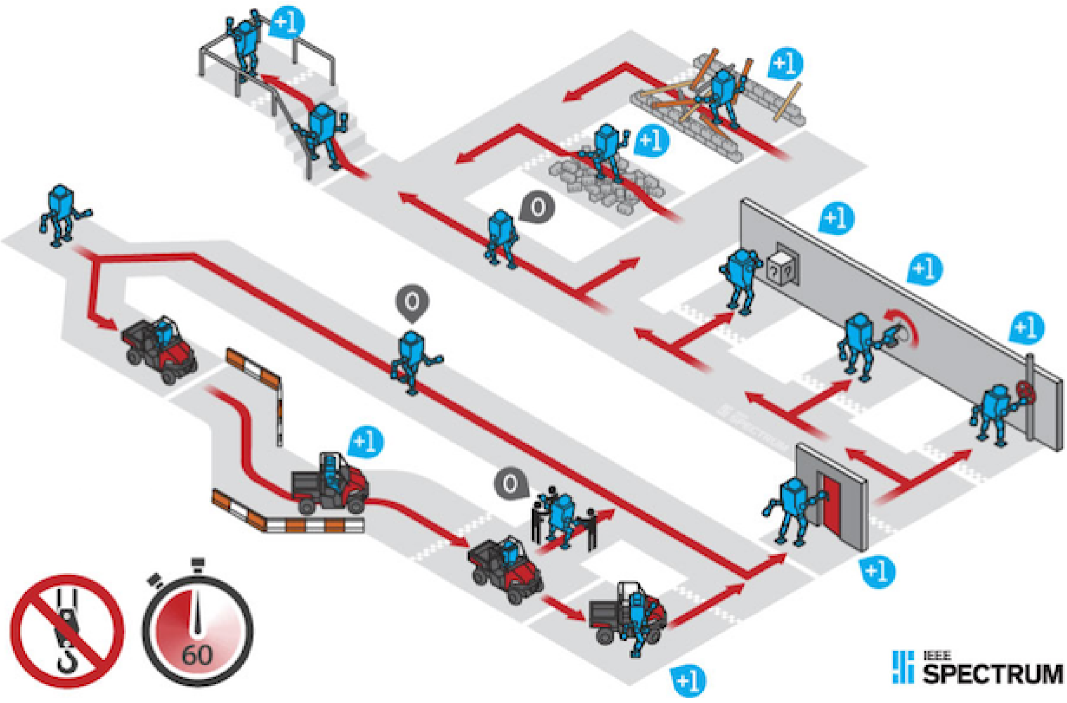

Fig. 2.3: DARPA disaster response tasks description graphics[2]

perform in the disaster response challenge.

There were 8 tasks in total for the robots in the challenge: driving a utility vehicle from the start area to the hazardous area, to egress from the vehicle, opening a door and entering into the hazardous area, opening a valve, cutting a wall with a circular shape using a drill tool that the robot had to grasp, a manipulation surprise task, passing over rubbles or debris area to exit the hazardous area and finally climbing stairs. All these tasks were performed inside the hazardous area with some degraded communication between the robot and the operator. The robots were allowed to be semi-autonomously controlled over a wireless communication network (i.e., human operators in the loop still do much of the control in the challenge). The robots were able to successfully pass some impressive tests, including the driving task.

In **door task**, the robot has to open the door and enter the disaster site. Figure 2.4 on the left shows the KAIST DRC-HUBO humanoid robot successfully performing the door opening task. Here the robot performs the manipulation task for opening the door after walking towards it. Few humanoid robots failed at the door task as shown in Figure 2.4 on the right since they did not manage to balance the disturbances induced by the manipulation motions. Planning for the whole body motions with multi-contacts as demonstrated by KAIST DRC-HUBO could have handled these disturbances to provide stability during the manipulation. These problems were frequently encountered in all the tasks which needed manipulation in addition to the locomotion.

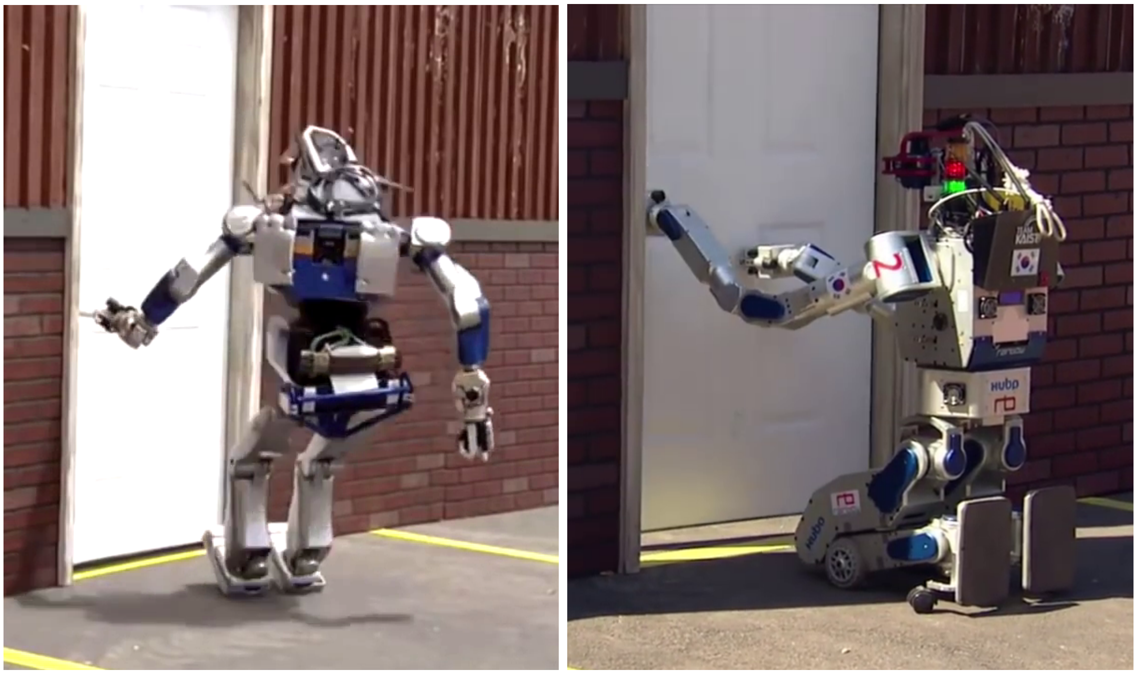

Fig. 2.4: Team HRP2-Tokyo (left) failed in door opening task shown. KAIST DRC-HUBO (right) successfully performs door opening task.

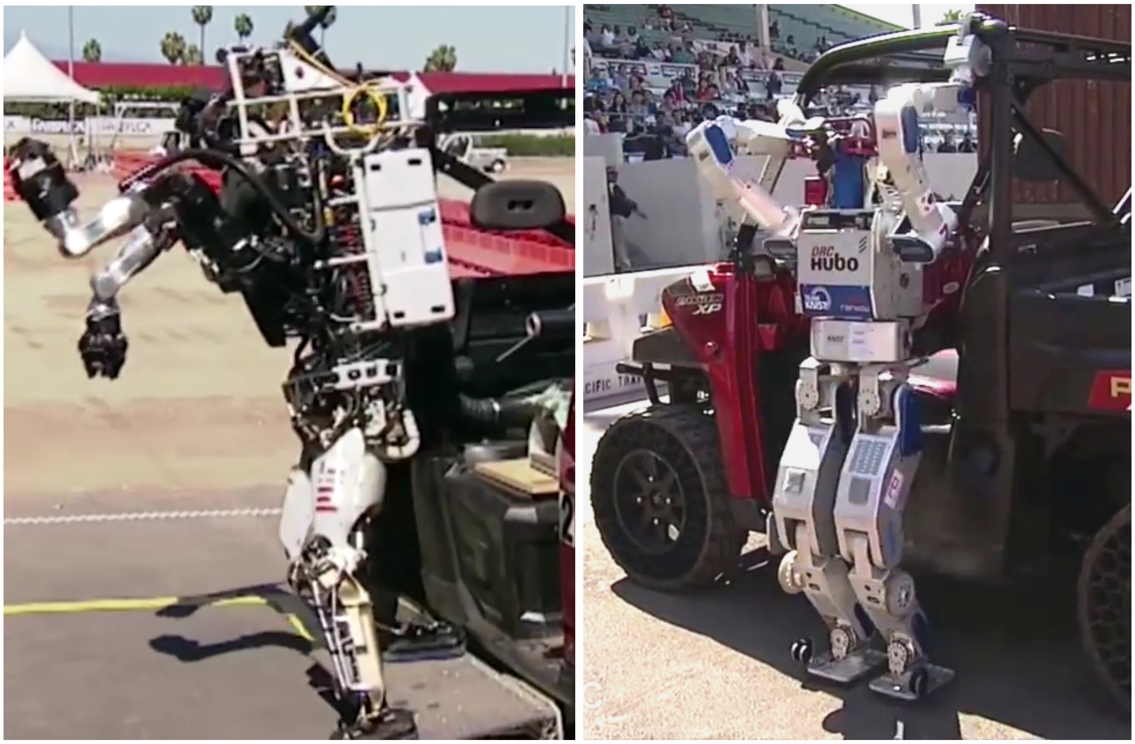

Fig. 2.5: MIT ATLAS (left) spills while performing the car egress task. KAIST DRC-HUBO (right) successfully performs the egress task.

In **egress task**, the robot has to move out of the vehicle after the driving task to a marked region. This was probably the most difficult task in the entire challenge, because the robot had to extricate itself from a complicated driving position all the while balancing itself and the robot body movements without any collisions. Many robots had opted to be taken out manually with the human assistance, while only few robots took up the challenge. The MIT ATLAS humanoid robot failed to balance itself using only its feet and it lost its balance while taking its first steps and took a spill, as shown in Figure 2.5 on the left. However, KAIST DRC-HUBO robot was successful in the egress task, as shown in Figure 2.5 on the right. The multi-contact planning approach helped the robot to execute the egress task successfully without taking a spill. The robot was able to maintain its balance while being guided by human operators with motion plans to use the arm grippers to grip on additional support holds mounted on the vehicle.

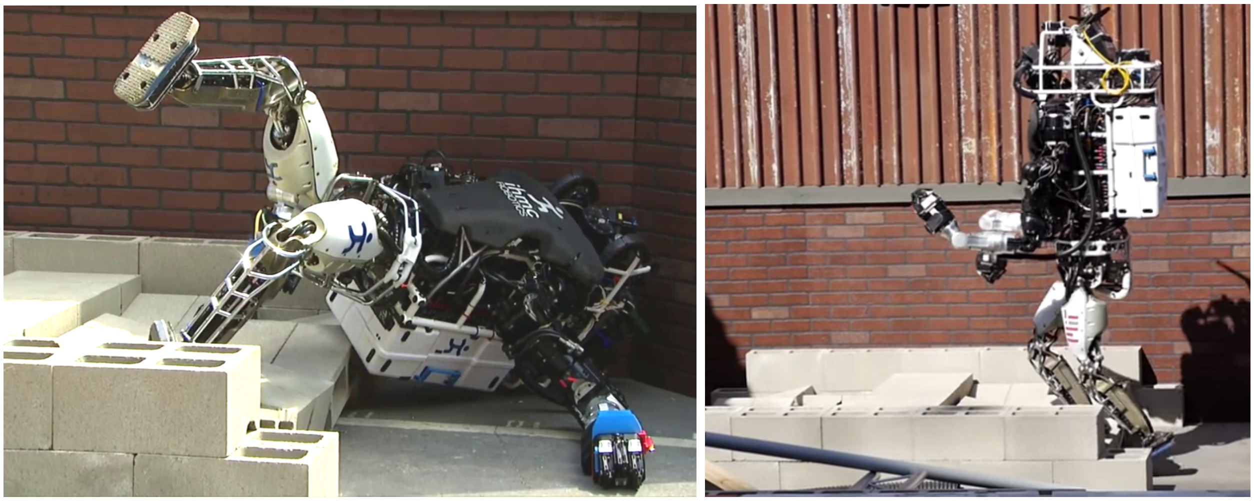

Fig. 2.6: IHMC ATLAS (left) takes spill while performing uneven terrain task. MIT ATLAS (right) successfully performs the uneven terrain crossing task.

In **rubble task**, there were both the uneven terrain crossing and debris removal tasks. Robots could choose whether they wanted to deal with the uneven terrain or the debris removal task. The MIT ATLAS robot was successful in completing the uneven terrain task as shown in Figure 2.6 on right. Although, most of the robots which attempted the uneven terrain task had taken spills while climbing on it, due to minute uncertainties in the cinder blocks used as an uneven terrain. Figure 2.6 on the left shows the IHMC ATLAS robot tasking a spill during the uneven terrain.

Although, we contend that a multi-contact approach with contacts on the adjacent wall can increase the robot support and improve its balance leading to a successful execution of the task. The probability of the humanoid robots falling increased since whole-body multi-contact motions were not planned for the tasks. Eventually, this resulted in many humanoid robots taking spills during the course of the challenge. Also, any additional contacts with the surrounding environment were strictly avoided during the process and motion planners explicitly generated paths that incorporate safety distance margins to avoid such collisions. Due to this, the humanoid robots did fall short of the many expectations to execute tasks as well as humans, in a disaster like situations at the DARPA challenge.

We see that these humanoid robots with multiple DOFs were difficult to be semi-autonomously controlled, even with human-in-the-loop for guidance throughout the course of the task. Also, balancing with only the foot contacts for the robot was not enough to keep the robot stable. We propose that the robot could have taken due advantage with additional contacts over the surrounding objects in the environment to support itself during the tasks. Also, due to multiple DOFs in a humanoid robot, the complexity of controlling is increased and it is further compounded by the difficulty of planning for such high-DOF systems involving external collisions and ground contacts. In all of the above experiments, the planning for locomotion in humanoid robots was restricted to mostly footstep generation on flat surfaces with minute disturbances.

2.2 Path Planning for Humanoid Robots

Path planning for locomotion of humanoid robots in human-centered environments consisting of obstacles needs complex algorithms. Path planning is an essential primitive for autonomous robots, and it lets the robots find the shortest distance or the optimal path between two locations. This optimal path could be one that minimizes some path cost like the amount of turning, braking or whatever the specific application requirements may be. Thus, the path planning algorithms deal with the problem of searching these sequence of actions that will cause the robot to achieve a goal state from its initial state (LaValle (2006)). This path may contain discrete or continuous elements, such as point coordinates or curves, respectively. In other words, path-planning finds a path that travels through the representation related to the robot's state in the environment.

Graph-based search algorithms are widely used for path planning in robots like A* algorithm. A* algorithm finds optimal paths as long as an admissible cost-to-go heuristic is selected for the algorithm. Particularly the planning problem for humanoid robot locomotion was simplified by restricting the planning dimensions by only planning for the footstep placements in the environment. A 2D map is used for the representation of the robot environment.

Another graph-based algorithm like Anytime algorithm, which implies that the algorithm can search quickly sub-optimal solutions and upon several runs refines that solution (Hansen and Zhou (2007)). Such algorithm can allow for efficient re-planning in case of changes in the environment or the initial solution. The algorithm modifies data in the search-tree instead of performing a new search from scratch. This often leads to faster re-planning calculations when map information changes but the start and goal location remain the same. Anytime algorithms that can be applied in the context of A* algorithms are called Anytime Repairing A* algorithms. Footstep planning for the humanoid robot using Anytime Repairing A* algorithm was applied for faster re-planning in changing environments (Hornung et al. (2012)).

A graph-search algorithm can be used to find a solution that is optimal with respect to the graph model of the world, but that may not be optimal with respect to the real-world. Graph-search based algorithms are currently the most common technique for robots operating in three or fewer dimensions. Complete solutions are often unfeasible however, when the possible state space is large. This is the case for robots with multiple degrees of freedom such as humanoid robots. In practice, most algorithms are only resolution complete, i.e., only complete if the resolution is fine-grained enough, as the state-space needs to be somewhat discretized for them to operate (e.g., into a grid) and some solutions might be missed as a function of the resolution of the discretization. The method of choice in more than three dimensions is random-trees.

The configuration of a robot is a specification of position and orientation of the robot w.r.t a fixed frame in the workspace. A set of all allowable configuration forms a configuration space (C-space). Random-tree algorithms are used in complex C-spaces when other methods are computationally prohibitive. Random trees perform graph construction simultaneously with graph-search. New nodes, created by sampling continuous points in C-space, are linked to old nodes using a predetermined procedure. Paths provided by random trees can be far from optimal. However, they are often used for problems so complex that finding any solution is considered an achievement. They can be implemented in many different frameworks, including those described as 'Anytime'.

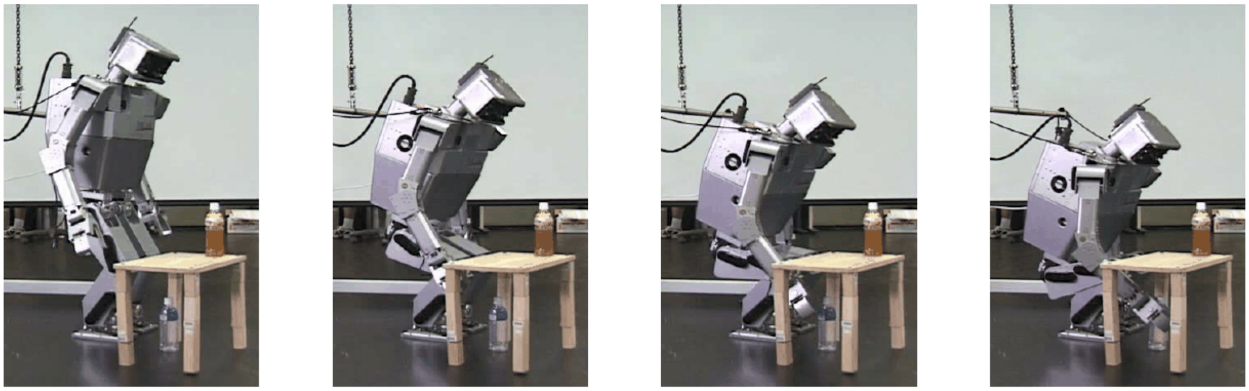

Fig. 2.7: Humanoid robot H6 performs full-body motions to retrieve an object[3]

Planning in configuration spaces is akin to the motion planning problem which is complex in its general form (Reif (1979)). So, these planning algorithms have been limited to planning problems in the lower dimensions problems of C-spaces. This has led to use of heuristic-

[3] Reprint by permission of authors: Kuffner et al. (2000).

based algorithms, many of them employed randomization for searching the C-space e.g., (Barraquand and Latombe (1991), Amato and Wu (1996), Horsch et al. (1994), Bohlin and Kavraki (2000), Hsu et al. (1997), Mazer et al. (1998), Boor et al. (1999), Kuffner and LaValle (2000), LaValle and Kuffner (1999)). Although these planning algorithms have no guarantee or completeness, with enough samples in high-dimensional configuration spaces many are shown to find solutions.

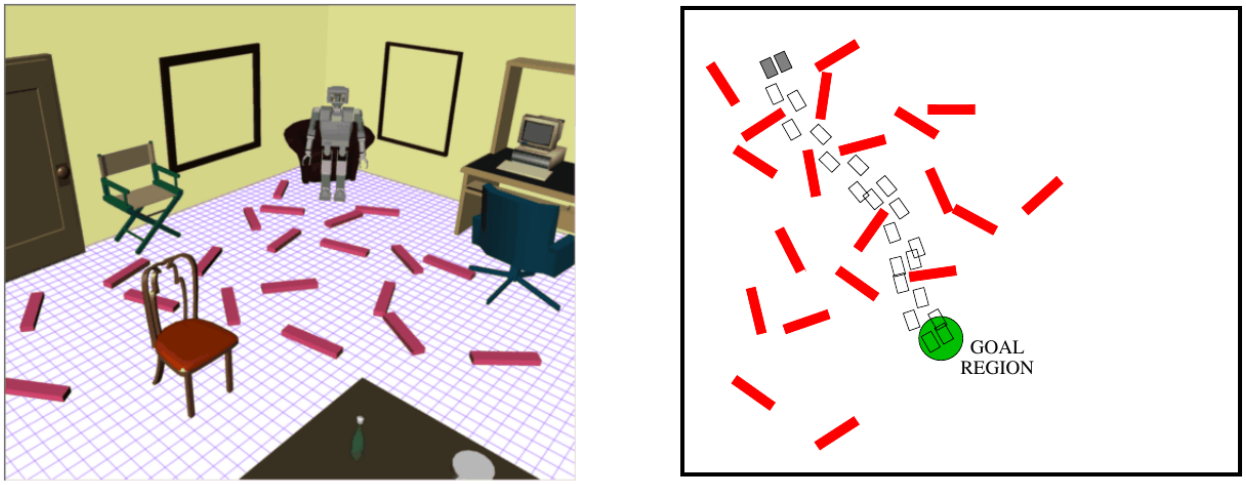

Fig. 2.8: Navigation for humanoid robot in cluttered environment (left). Right image shows Footsteps planned for the robot to reach the goal[4]

Rapidly-exploring random trees (RRTs) based RRT-Connect algorithm was used to plan dynamically-stable motions for the humanoid robots in the C-space (Lavalle et al. (2000), Steven M. LaValle and James J. Kuffner (2001)). The full-body motion planning for a humanoid robot in the C-space to retrieve an object below a small table was demonstrated in (Kuffner et al. (2000)) as shown in Figure 2.7. The RRT-Connect planning method utilizes the Rapidly-exploring Random Trees to connect two search trees, one from an initial configuration and the other from the goal configuration. The random point samples in C-space with single or dual leg support configurations were used in the tree search exploration. In the second phase, the collision checker was used in conjunction with a dynamics filter to generate a final collision-free dynamically-stable trajectory. Footstep planning for the humanoid robot H6 with A* algorithm along with RRT-Connect algorithm to generate a collision-free statically stable solution for locomotion was demonstrated by (Chestnutt et al. (2003)) as shown in Figure 2.8.

The problem of footstep planning for humanoid robots, or searching for a list of step locations to reach the desired goal, has been solved using two classes of techniques: discrete searches and continuous optimizations. In general, discrete searches require some methods to estimate the possible displacements from one step to the other (Eilering et al. (2014), Bouyarmane and Kheddar (2012)). For instance, through approximations to the reachable space for the feet (Perrin et al. (2011)), or with a predefined set of possible footstep locations (Kuffner et al. (2003)). The problem can also be formulated as an optimization problem on the poses of the footsteps (Deits and Tedrake (2014), Herdt et al. (2010)), but also using some sort of geometric approximation to the reachable regions for the footstep locations.

A randomized motion planner for a robot to avoid collision with moving obstacles under kinematic, dynamic constraints and reach a specified goal state was presented by (Kindel et al. (2000)). Here the planner samples the state space of a robot by selecting control inputs at random in order to compute a roadmap that captures the connectivity of the space. Instead of precomputing a roadmap as most PRM planners do (Kavraki et al. (1996)), for each planning query, it generates on the fly, a small roadmap that connects the given initial and the goal state.

Imitation Learning (also known as programming by demonstrations or learning from demonstration) is one of the fundamental learning mechanisms in humans' daily lives. In robotics, imitation learning (IL) had started attracting attention at the beginning of the 1980s. Through the years, IL has been used as a powerful means to bootstrap robot learning (Ijspeert et al. (2002)). IL provides an intuitive way to transmit skills to robots without explicitly programming them. IL-based approaches have proved to be an exciting alternative to classical control and planning methods in different applications such as learning tennis swings (Ijspeert et al. (2002)), locomotion (Ratliff et al. (2007)) and control of acrobatic helicopters (Coates et al. (2008)).

2.3 Whole-Body Control for Humanoid Robots

A new control methodology for whole-body control was put forward for controlling full-body motions in humanoid robots. Whole-body control unlocks the full capability of controlling humanoid robots to perform various human-like tasks and behaviors. A whole-body control approach utilizing an optimization of contact forces in combination with Model Predictive Control was used for balancing and posture stabilization of the humanoid robot TORO (Henze et al. (2014)).

Another entirely different approach to whole-body control presented in (Sentis and Khatib (2006), Sentis et al. (2010)), based on the prioritization of different tasks using the framework of operational space control. The facets of the humanoid robot control with motions including contacts, and constraints were unified with a full integration of task-space control and posture control (Khatib et al. (2008), Sentis and Khatib (2005), Park and Khatib (2006b)). Whole-body control was achieved through a development of a task consistent posture Jacobian and a model for inducing the dynamic behavior in the posture space. There are many frameworks including those based on Inverse Dynamics, implemented on top of a pure low-level torque

[4] Reprint by permission of authors: Chestnutt et al. (2003).

control (Sentis et al. (2010)), it is difficult to find hardware platforms mature enough to implement control schemes of such frameworks and up to now no complex tasks have been experimentally demonstrated yet in humanoid bipedal robots. The control library called OpenSoT implements the idea of decoupling task descriptions and solvers to execute multiple Cartesian tasks to achieve complex motion behaviors for humanoid robots in position or velocity space, which is realizable to demonstrate on most robot platforms (Rocchi et al. (2015)).

A whole body motion trajectory can be generated for the humanoid bipedal robot based on a decoupling task space which is solved using Quadratic Programming optimizations. OpenSoT employs a QP solver (Ferreau et al. (2014)) to implement a cascade of Quadratic Programming (QP) problems for a set of tasks and constraints in velocity space, in order to solve a hierarchical inverse kinematics problem on a humanoid robot. We describe the OpenSoT formulation in detail, which we use extensively in this thesis for generating the whole body motions from a kinematic perspective. The robot can execute n tasks T_i and for each of these tasks, an error function $e_i(q,t)$ describing the Cartesian error in the tasks is provided. We can compute the derivative of the error w.r.t. time t as in (2.1), which can be further deduced to be associated with the task Jacobians, J_i, as in (2.2).

$$\dot{e_i} = \frac{\partial e_i}{\partial q}\dot{q} + \frac{\partial e_i}{\partial t} \tag{2.1}$$

$$\dot{e_i} = J_i\dot{q} + \frac{\partial e_i}{\partial t} \tag{2.2}$$

where q is joint positions and $\dot{q}$ is joint velocities.

By representing the Cartesian error derivatives as Jacobians, we can obtain the equation which combines the joint positions, velocities as the reference error changes. Rearranging the terms of the equation in(2.2), we can force the task errors to converge to zero by applying an exponential dynamic as

$$J_i\dot{q} - \frac{\partial e_i}{\partial t} = -\lambda e_i \tag{2.3}$$

The whole-body IK solver uses QP optimization with the possibility to specify hard and soft priorities between tasks as well as with linear constraints, $A_{c,n}$ and bounds, $b_{c,n}$. The solver is based on a framework from LAAS (Mansard et al. (2009)). The Cartesian tasks described by (2.3) are formulated in a QP form to solve for the solutions as

$$\begin{aligned} \dot{q}_1 = \quad & \arg\min_{\dot{q}} \quad \|J_i\dot{q} - \dot{e}_i^*\| \\ & s.t. \qquad A_{c,1}\dot{q} \leq \mathrm{b}_{c,1} \end{aligned} \tag{2.4}$$

Since a humanoid robot is redundant with respect to task, secondary Cartesian tasks can also be included in the equation without affecting the performance of a primary task, which allows the robot to execute tasks by utilizing its whole-body motion capabilities. Therefore, to generate whole-body motions, we can have a series of QP problems in cascade to be solved (Kanoun et al. (2011)). This is a well known method to derive motions by executing tasks adding bilateral linear constraints to the inverse kinematics problem (Escande et al. (2014)). The formulation used in (2.4) for the constraints can be profitably used to express lower and upper bounds for the variable value as well as equality constraints. In general, the n$^{\text{th}}$ task will then be written as:

$$\begin{aligned} \dot{q}_d = \arg\min_{\dot{q}} \quad & \|J_n\dot{q} + \lambda e_n + \frac{\partial e_n}{\partial t}\| \\ s.t. \quad & A_1\dot{q} = A_1\dot{q}_1 \\ & \vdots \\ & A_{n-1}\dot{q} = A_{n-1}\dot{q}_{n-1} \\ & A_{c,1}\dot{q} \leq \mathrm{b}_{c,1} \\ & \vdots \\ & A_{c,n}\dot{q} \leq \mathrm{b}_{c,n} \end{aligned} \tag{2.5}$$

where $\dot{q}_d$ is the desired velocity.

In (2.5) the previous solutions $\dot{q}_i$ are taken into account with constraints of the type $A_1\dot{q} = A_1\dot{q}_1 \quad i\forall < n$; so that the optimality of all higher priority tasks is not changed by the current solution. While in (2.5) the first task has a relationship of hard priority with respect to the second, and so on, for each level of priority, a soft priority relationship between tasks can be imposed introducing the relative weights β_i, so that the augmented Jacobians and the error vectors can be written as

$$\begin{aligned} J_{\text{aug}} &= [\beta_1 J_1^T \quad \dots \quad \beta_n J_n^T]^T \\ e_{\text{aug}} &= [\beta_1 e_1^T \quad \dots \quad \beta_n e_n^T]^T \end{aligned} \tag{2.6}$$

where J_{aug} are augmented Jacobians, e_{aug} is error vectors associated with it.

The soft priorities between tasks is altered by tuning the relative weights β_i, with higher priority tasks having larger β_i. As already mentioned, in this case, the tasks can still influence each other's performance. Therefore, a combination of hard and soft priorities are needed to describe a stack of tasks. The solution obtained can then be sent directly to a velocity controlled robot or integrated in a position controlled robot as

$$q_d = q + \dot{q}\delta t \tag{2.7}$$

where δt is the control loop period, q_d is the obtained joint solution.

2.4 Multi-Contacts for Humanoid Locomotion

Current approaches for the multi-contacts in humanoid systems address the contact planning through user-defined plans/hints, global graph-search methods and other planning methods discussed in Section 2.2. In contrast, our multi-contact motion planner uses a demonstration of the tasks, to obtain the sets of contacts, which is used for initializing the contact optimization. Our planner uses a QP solver for IK, i.e., searching the C-space, and also incorporating self-collision, unintended collision avoidance along with task prioritization for the supporting contacts of the goal.

In comparison, Escande et al. (2013) used adapted best first planning (BFP) with potential functions defined over C-space using the CoM and target goal position, along with the user-defined rough key-postures describing the task and user-selected contact points on the robot. The BFP planning process searched for witness postures attesting to the feasibility of a set of contacts. The witness posture is constructed using a set of contact tasks, which satisfies the joint limits, auto-collision, collision constraints to builds a set of C-space. The BFP planning approach is applied for scenarios involving multiple surfaces, and contact candidates are searched with good coverage of the surface area. A total of 497 nodes were generated for the case of table scenario experiment with multi-contacts. In contrast, our generalized contact planner introduced in Chapter 5, learns a general policy from the demonstrated task, to map the candidate contacts and select only a subset of contact actions for the multi-contact behavior using the robot environment state information.

Mordatch et al. (2012) presented a motion synthesis based on Contact Invariant Optimization(CIO) framework for producing a wide variety multi-contact behaviors similar to human behaviors such as getting up, crawling, climbing, etc. The methods exploited the set of active contacts which remain invariant movements phases of multi-contact motion. It was necessary for their approach to predefine the contact patches on the robot. Similar to our multi-contact motion planning approach they optimize the contact positions using the Limited-memory BFGS (L-BFGS) method. However, the contact sequences are neither defined nor obtained with any motion capture tools. In contrast to our approach of using stability costs for contact optimization, they formulate the contact optimization costs as an error violation cost with contact position, angle, and velocity, along with auxiliary variables which allow for activating the contacts during optimization. Also the costs for dynamics, ZMP and task goal were added in optimization. The task goals were encoded at a higher level by specifying the target position and velocities of the movement. Besides, it was necessary to define the number of movement phases essential for the task in the optimization step, whereas our generalized contact planner can find the necessary movement phases/key-frames with the help of a running cost-to-go metric.

Lengagne et al. (2013) generated whole-body motions in multi-contact motions for HRP-2 using semi-infinite programming formulation. The motion planning formulation considered the balancing constraints for contact forces, which is minimized using B-spline parameterization to contour the dynamics effects, along with collision constraints for unintended external collisions. Experimental validation of a sitting motion with HRP-2 was shown, where user-defined contact positions i.e., alternating contacts with both the left and right hand and feet contacts were specified.

Englsberger et al. (2014) presented a multi-contact passivity-based controller for climbing stairs with the humanoid robot TORO using the handrails as support contacts. Their proposed planner uses different methods for footsteps and hand contacts. A complete pipeline was used to generate quasi-static motions from either perception input using graph-based search approach for footstep planning. A Constrained Rapidly-exploring Random Trees (CBiRRT) for the planning hand contacts along with gradient-based optimization to obtain IK for the robot. Climbing stairs at the height of 0.05m, while using a handrail was used to experimentally validate their approach. Chung and Khatib (2015) presented a Contact-consistent Elastic Strips (CES) framework for humanoid locomotion in unstructured environments, where the environment scene was scanned for contacts to choose candidate support regions. In their approach, a global planner was used to guide the initial solution for the contact planning framework. Here also they restrict the multi-body contact with the environment to only wrists and feet to increase support during locomotion, whereas we try not to limit the contact to only a few robot body parts.

A multi-contact approach for ladder climbing was presented in (Vaillant et al. (2016)). This approach makes use of both a multi-contact planner along with multi-objective closed-loop control formulated as a QP. The set of contacts to climb the ladder was planned off-line by the user. The planning approach used a greedy search behavior to seeks for all possible contacts. The controller provided the desired states in terms of joint accelerations and contact forces to be tracked by the low-level motor controllers.

Advanced techniques for multi-contact humanoid locomotion in unstructured environments using support hiking poles was solved in the SupraPed framework, which builds on a whole-body framework by adding the friction constraints (Khatib and Chung (2014)). The supporting forces and moments are obtained by mapping the contact forces using a contact force grasp matrix (Ott et al. (2011)) and moments using the virtual linkage model for balancing the internal forces (Sentis et al. (2009)). The robot design incorporates a pair of actuated smart poles with vision and force sensing that transforms the biped humanoids into tripeds or quadrupeds. The multi-contacts are handled with a contact-consistent Jacobian to ensure that the tasks will not interfere the contact states of the robot by projecting the tasks to the null space of contacts (Park and Khatib (2006a)). The multi-contacts presented were confined to the robot feet and external support poles grasped with hands. In comparison, our work in this thesis extends this variable number of point contacts in the obstacle negotiating task, using bipedal, quadrupedal or tripedal stances but critically it also uses the limbs of the robot (hands, arms, legs, feet) to execute the task.

2.5 Conclusion

In this Chapter, we discussed the challenges faced by humanoid robots at the DARPA robotics challenge and how they could have prevented spills during the tasks. We also reviewed the state-of-the-art planning algorithms for the humanoid robots like graph-based algorithms such as A*, Anytime Repairing ARA* and sampling-based algorithms such as RRT-connect and PRM methods. RRT-connect algorithms are used for planning humanoid robot motions in the C-space while making use of the whole-body, and A* or graph-based search algorithms can give an optimal path for navigating in the environment. We also reviewed the capabilities of imitation learning in humanoid robots. We also discussed the state-of-the-art whole-body control methods for humanoid robots. The approaches used were model predictive control, task prioritization in operational space control, inverse dynamics, etc. We described OpenSOT for generating whole-body motions based on a decoupling task space which is solved using QP. We also discussed the current approaches used for planning the multi-contacts in humanoid systems using the adapted BFP method, CIO method, semi-infinite programming, CBiRRT planning, CES framework, SupraPed framework, etc.

CHAPTER 3

Initial Solutions for Negotiating Obstacles

Tasks for humanoid robot systems have traditionally been hand-designed by human experts by specifying the goal plans explicitly. Research in humanoid robots is challenging due to the numerous degrees of freedom in the system, that makes it harder to specify the task goals. Finding initial solutions for the task provides critical insights into the problem often leading to optimal solutions. Identifying such initial solutions can determine the limitations of the system and the task variables necessary for successful execution of the task. To begin with, we start with the problem statement of the thesis which builds on the hypothesis.

Problem statement: *Enable humanoid robots to climb large over obstacles with multiple contacts over the obstacle. The robot must use the obstacle to support itself during the process of negotiating the obstacle.*

We make a few assumptions for the problem of negotiating i.e., either climbing or overcoming the obstacle, regarding the shape of the obstacle used or the type of the contacts allowed in the simulations and experiments. We list the assumptions for the problem as follows:

1. A simple cuboid is used for the obstacle.

2. Unilateral contacts are allowed.

3. Obstacle height does not exceed the robot's reach.

4. Any robot body link can be in contact with the obstacle.

We consider a simple cuboid as an obstacle for the negotiating task, as shown in Figure 3.1. The robot can only make unilateral contacts with the obstacle, i.e., the robot can support at

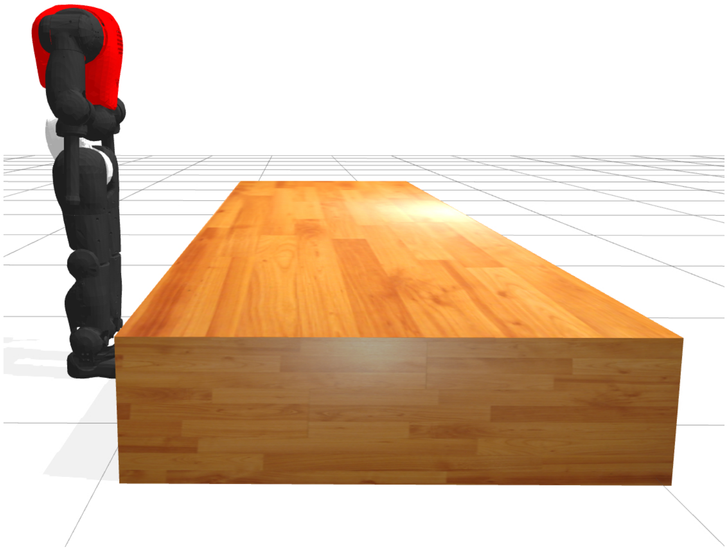

Fig. 3.1: The humanoid robot COMAN for negotiating over a large wooden obstacle via multi-contact motion planning is shown. The virtual world and simulations are in the Gazebo simulator environment.

contacts by by solely pushing against them. The obstacle height selected for the task must not be too high for the robot and it should be physically able to climb on the obstacle. Also, we allow any robot body parts to be in contact with the obstacle during the execution of the task.

The humanoid robot COMAN (shown in Figure 3.2) is used in our experiments. COMAN has 23 degrees of freedom (DOFs) and is a medium-sized robot that roughly approximates the dimensions of a 4-year-old child. The height of the humanoid robot COMAN from the foot to the center of the neck is 945 mm. The total weight of the humanoid robot is 31.2 kg, with the legs and the waist module weighing 18.5 kg and the torso and the arms weighing 12.7 kg. One unique feature of the robot COMAN is its passive compliance in the legs and arms, which makes it more robust but also more difficult to control. Special care needs to be taken to adapt the conventional ZMP-based walking pattern generator for the compliance in the legs (Kryczka et al. (2013)). Initial solution for the obstacle negotiating problem can determine the limitations and the task variables necessary for successful execution of the task as follows:

1. Feasibility of the task
2. Size of obstacle possible to negotiate
3. Strategy needed for the task
4. Static or dynamic motion

The goal here is to obtain a good initial solution in the simulation environment. Most state-of-the-art algorithms use simulation as a first step in the experimental phase. Here we

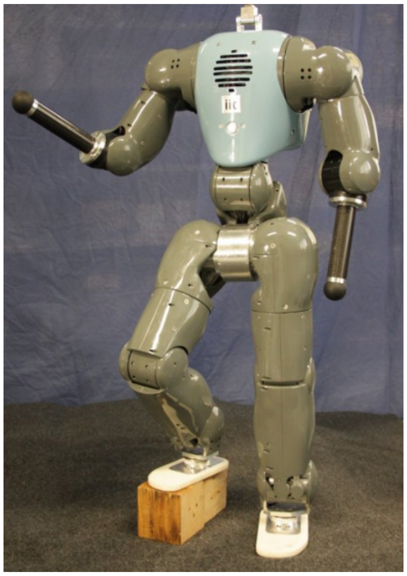

Fig. 3.2: The humanoid robot COMAN used for the experiments.

also resort to simulations, before experiment on the actual robotic system. The goal here is to obtain a good initial solution in simulation to start with, then to analyze the core challenges at each stage of the problem before further experimentation on the robot.

For simulation purposes, we use the Gazebo robot simulator. A virtual world is set up with a large wooden obstacle relative to the size of the robot in the environment. The obstacle is set as a static object with certain frictional and other material properties. A Unified Robot Description Format (urdf) model of COMAN is added to the scene for simulation. A combination of both Yarp and ROS middle-ware is used to send control signals and collect data from the simulations, while the same middleware can be readily used to control the actual robotic system (Metta et al. (2006), Quigley et al. (2009)). We use Yarp-Gazebo plugins to enable sensing encoders, IMU and control the simulated joints of the robot in the Gazebo simulator (Mingo Hoffman et al. (2014)).

3.1 Explored Methods for Initial Solution

Identifying the necessary task variables enables us to design a suitable obstacle in the simulation environment for the multi-contact task. We explore several methods to obtain an initial solution for the task.

3.1.1 Kinesthetic Teaching Method

The kinesthetic teaching approach provides demonstrations to a robot whereby a human guides a robot physically to perform a skill. The robots trajectory are recorded during the kinesthetic teaching process from start to end. This provides an easy way to enable the programming of a robot with the skills by showing successful examples. Kinesthetic teaching have been very successful in robot arms with 6/7 DOFs to collect skill demonstrations, while the robot arm is in gravity compensated zero torque mode. Although, in humanoid robots, it is necessary to have active balancing controllers enabled to counter the disturbances during such interactions. For example, a humanoid robot was taught to complete a board cleaning task, during quiet standing, through kinesthetic demonstration of the task as presented in (Kormushev et al. (2011)).

Although for the case of overcoming an obstacle with contacts, it is not possible to have an active controller since with every new contacts, the balancing constraints for the robot changes. Also, the high DOFs in the system makes it difficult to teach the robot. Hence, kinesthetic teaching methods are not a feasible means of demonstrating a climbing task using a humanoid robot.

3.1.2 Teleoperation Methods

Teleoperation methods are used to control the robots in a semi-autonomous mode from a distance. Usually, the mobile robots are easy to control in semi-autonomous modes for navigation due to the fewer DOFs in the system. Here we try to make use of a pseudo teleoperation method i.e., to control the robot in simulation, to get insight into the obstacle negotiation problem. Initially, we set up a virtual environment consisting of an obstacle (cuboid shaped) along with simulation version of COMAN in Gazebo simulator, as shown in Figure 3.1. The robot is operated in a joint position PID controlled mode.

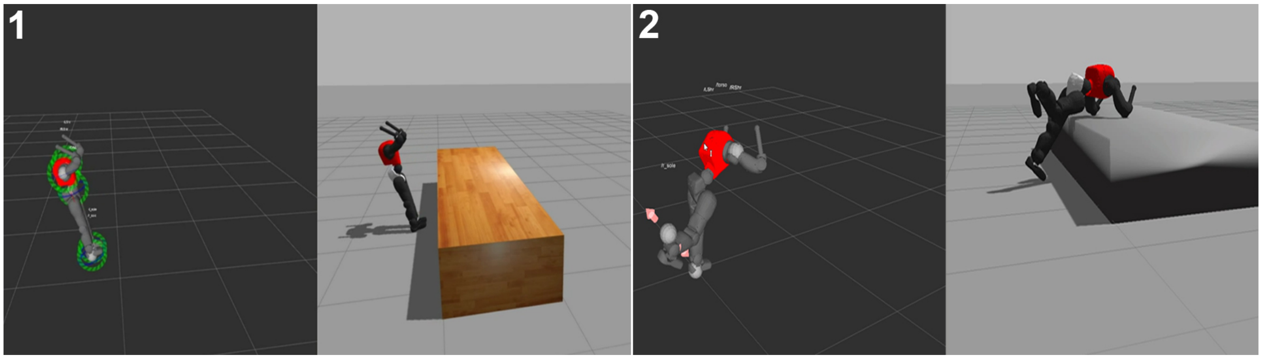

Fig. 3.3: RVIZ visualization of the robot with interactive markers are shown on left-side of images and on right of images is the Gazebo simulation environment with the robot and obstacle. In image 1, the robot falls backwards due to instability induced due to high joint velocities during the psuedo teleoperation. Image 2 shows the robot configuration with elbows in contact and right knee almost in contact with the obstacle.

We use the ROS visualization (RVIZ) tools to visualize the robot and provide interactive control inputs to the robot in the simulator. 6D interactive markers are employed in RVIZ, to provide the interaction inputs to control the robot in semi-autonomous mode. These markers allow us to set the position and orientation references for the desired robot links, selected to be controlled in semi-autonomous mode. COMAN has 23 DOFs distributed across its five kinematic chains i.e., right leg, left leg, left arm, right arm, and the torso. We assign Cartesian tasks for the robot links such as the left foot, right foot, right elbow and left elbow using their associated Jacobians and assign each of these tasks with an interactive marker in RVIZ, as shown in Figure 3.3 (see 1). Now, we can interactively control these 6D markers in RVIZ, which sends the desired position/orientation for the Cartesian tasks to control the humanoid robot motions.

We reduce the max velocity of the position controllers to avoid the robot from falling down due to the disturbances induced by the dynamic motions. Initially, we try to move the elbows closer to the obstacle. We were able to reach the obstacle surface with the current set height of $0.40m$ for the obstacle. After successfully moving both the elbow contacts over the obstacle, we send position command for the left leg such that the left knee can make contacts with the obstacle surface as shown in the Figure 3.3 (see 2).

At this point the obstacle height was too high and it was necessary to reduce the obstacle's height until the knee contact can occur on the obstacle surface. After reaching the robot configuration as shown in the figure, it was not possible to figure out the next viable movements for the tasks or which joints had to be actuated for the robot to climb on the obstacle. Due to the high DOF in the system, it was challenging to select the robot body links for the contact planning, specify the desired contact positions for them and to generate whole body coordinated motion behaviors, which were necessary for the multi-contact task.

3.1.3 Sampling-based Planning Methods

Most of the planning methods tend to avoid contacts with the obstacle in the planning process. Here, we need an inverse approach to find possible contacts for the humanoid robot. We use Gaussian sampling-based probabilistic roadmap methods (PRM) first proposed in (Boor et al. (1999)), for planning the multi-contacts for the obstacle negotiation task. Here, instead of randomly sampling the state space (Hsu et al. (1997)), we restrict the sampling space with a Gaussian function for the contact position of the robot body links on the obstacle, as in (3.1).

$$G(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(x-\mu)^2/2\sigma^2} \tag{3.1}$$

The Gaussian function, $G(x)$ with parameters sigma, σ and mean, μ are defined such that the contacts are sampled only in the reachable region for the particular robot link. For example, we show a sample set of possible contact positions for the left foot of COMAN. These samples are further filtered to eliminate the contacts penetrating the obstacle surface.

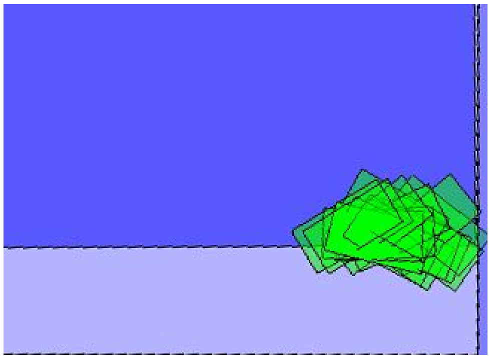

Fig. 3.4: The foothold configurations for the humanoid robot are shown with rectangles generated by probabilistic sampling of the reachable for space for the robot foot.

The resulting samples found for the foot contacts are shown in Figure 3.4. We see that there are many possible configurations for the contact positions at the left foot of robot along with different orientations. Moreover, if the task needs multiple contacts points at different robot link bodies, the search space of the samples increases. Along the problem of selecting which robot links to choose from for the contacts. Therefore, the contact planning problem with multiple contacts grows with the number of contacts needed for the task. Also, since only a stable posture must be searched in the planning process, the PRM methods will take a considerable amount of time to find a feasible solution depending on the complexity of the problem. The algorithms such as RRT, PRM methods require many heuristics and constraints to be defined for guiding the contact search efficiently. Thus, the problem to search a feasible solution for a multi-contact task cannot be computed in a finite time.

3.2 Initial Solution from Demonstrations

After several trials with the different contact planning methodologies, we use the motion capture system to directly obtain the solution in the motion space.

3.2.1 Motion Capture Suits for Demonstration

We use the Xsens MVN motion capture suit to directly record the demonstration for the task of overcoming an obstacle. A table is used as an obstacle to demonstrate the climbing task. The Xsens motion capture system is set up by providing human measurements, followed by sensor initialization routines for accurately recording the movements during the task. The Xsens movement data acquisition rate is set at 100 Hz. To perform the human demonstration of the obstacle-climbing task, we select the table with height $h = 0.35m$, such that the person can make contacts with the table obstacle using both the arms and legs in the process, illustrating that usage of additional body contacts can help in climbing large obstacles for a humanoid robot. Here the height of the table is a crucial determining factor such that both the arms and legs can reach it, as is the distance of the person in front of the obstacle. We then record the human demonstration of the climbing task with a motion capture Xsens capture suit. The demonstration begins with the person standing in front of the table obstacle and performing the climbing task as shown in Figure 3.5.

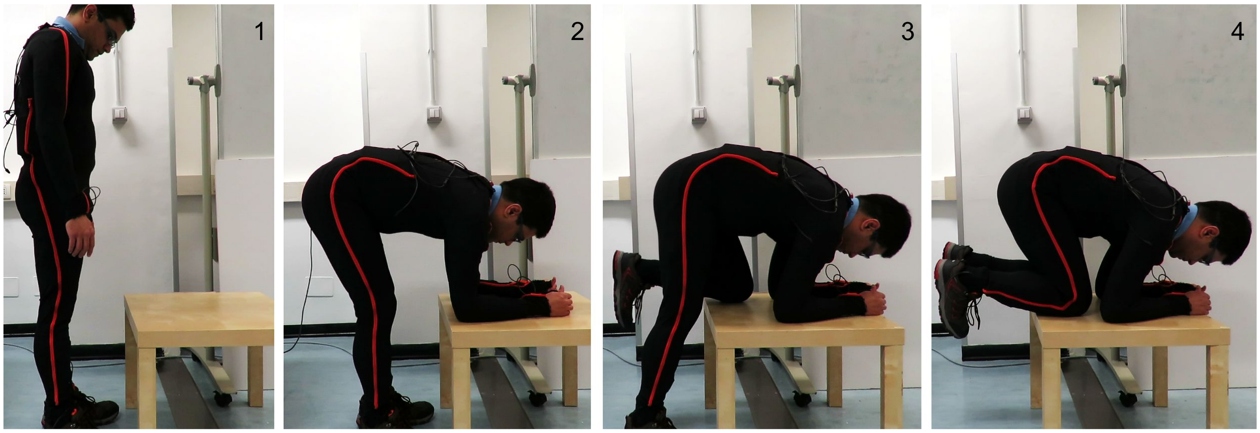

Fig. 3.5: The Xsens motion capture suit for recording the demonstrations of the climbing task. A demonstrator show the movements necessary to climb on the table obstacle for the multi-contact climbing task.

Motion mapping from suit to the robot: Xsens MVN motion capture suit consists of 18 inertial and magnetic sensor modules, each measuring 3 degrees-of-freedom at each body-based linkage spread over the motion capture suit. To transfer the recorded movements from the demonstrations to the humanoid robot COMAN we need to map by solving the joints correspondence problem. X-sens capture suit has 54 DOFs to capture the movement of the person wearing the suit; whereas the humanoid COMAN has only 23 DOF. We therefore have to significantly reduce the DOF of the motions to map the movements of the person to the humanoid robot.

For mapping, the Xsens motion data from the capture suit to the COMAN requires a reduction in the number of DOFs recorded, to 23 DOFs only. We ignore the 12 DOFs related to the hand because the robot only has support ends. We also ignore the 3 DOFs associated with the neck joint. Since we have a single 3 DOF joint in the pelvis, we reduce the 6 DOFs in the upper body to 3 DOFs at the pelvis joint, which reduces the number of DOFs to 30. We can ignore the additional DOFs in the capture suit at which the robot lacks DOFs over particular body regions, to reduce the total DOFs from 54 to 23. As a final step of the mapping, we constrain the motions to be within the robot joint motion limits as in (3.2). In addition to this, we have to compensate for the differences in the reference frames between the capture suit and the robot. Following this, we can directly map to transfer the motions recorded, to humanoid robot motions, in order to reproduce similar movements as in (3.3).

$$[\theta_{min}, \theta_{max}] \tag{3.2}$$

$$\theta_{Xsens} \rightarrow \theta_{COMAN} \tag{3.3}$$

Motion playback in simulations: We use the direct mapping of demonstrated movements for the humanoid robot given by (3.3) to playback the climbing task movements in the simulation environment. The environment consists of an obstacle (wooden cuboid) along with a simulation version of COMAN as shown in Figure 3.1. The robot is operated in a joint position controlled mode. The mapped joint positions from the demonstration are applied as joint position references to the joints of the humanoid robot. We see that the robot performs similar movements as demonstrated during the climbing task, although it fails to complete the task.

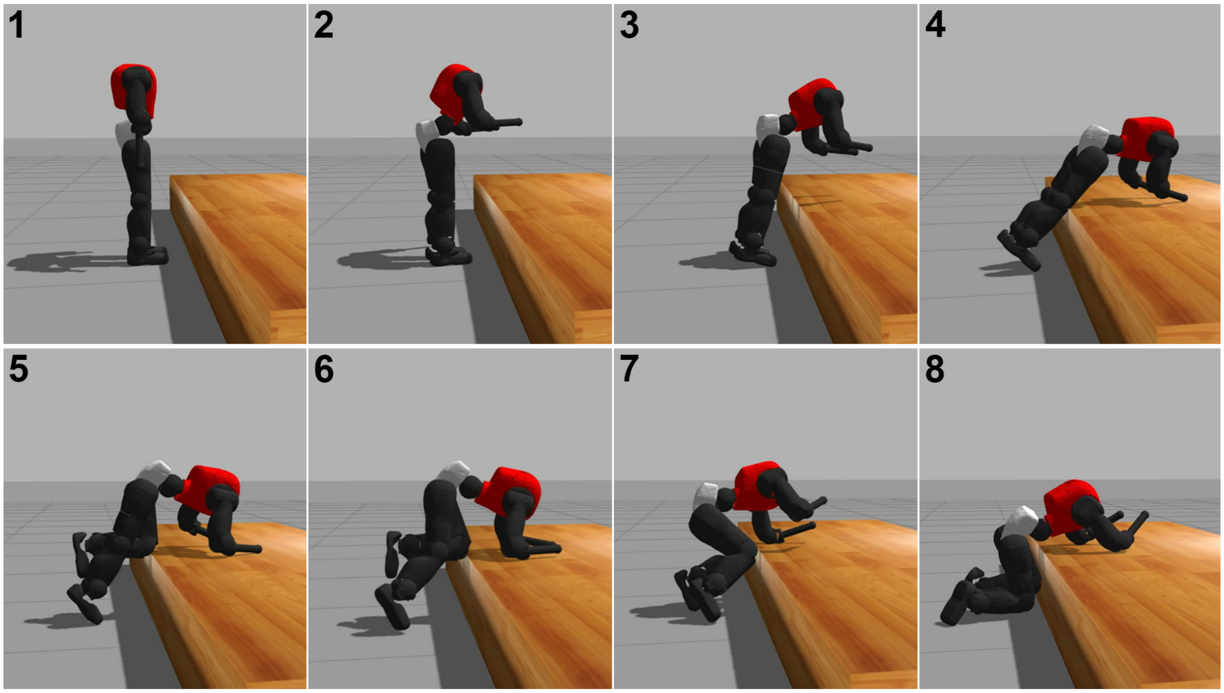

Fig. 3.6: Direct playback of the mapped movements from a human demonstration onto humanoid robot COMAN to perform the climb task over wooden obstacle in Gazebo simulation. The robot fails to complete the climbing task.

The differences in the physical size between the robot and the demonstrator causes the motion playback to fail. Thus, we adjusted the obstacle's height and robot's position in front of the obstacle, to make sure that both the robot arms and legs could make contact with it during the playback of movements. We tried to adjust these parameters several times during the playback of movements from the recorded motions to complete the climbing task on the robot in the simulation.

The robot nearly completes the task and manages to put one of its knees on top of the obstacle as shown in Figure 3.6. Although the robot nearly completes the task, there is instability, and the robot fails to finish the climbing just by playback of the human movements as shown in snapshot 6 of Figure 3.6. The playback gives crucial information, i.e., the obstacle height feasible for the climbing task, along with the distance at which the robot could be positioned from the obstacle.

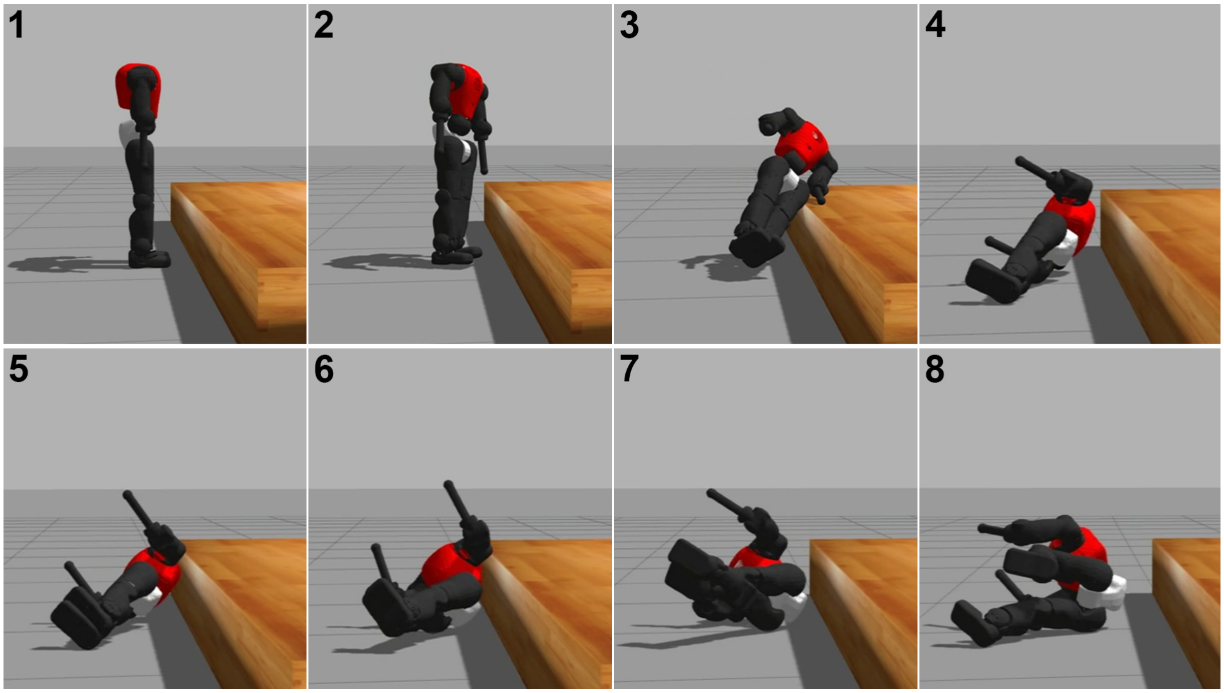

Fig. 3.7: During the direct playback of mapped movements from demonstration in Gazebo simulation, the robot falls on the ground, even failing to get any part of the robot on the obstacle.

Also, the outcome of each trail is slightly different from that of every other trial, resulting in separate results as shown in Figure 3.7. We see that there is instability at the beginning of the motion, with which the robot falls on the obstacle and causes impacts at the knees and hands of the robot. Although we see that the motion capture suit was able to capture the whole-body behavior with the recorded movements, the necessary contact interactions are not reproduced in the simulations. Hence, we have different contact interactions during each playback in the simulation, which results in different actions. We hereby propose the use of optical markers with an Opti-Track system, to better capture the contact interactions reliably, which is necessary for the multi-contact task during the demonstrations.

3.2.2 Optical Markers Tracking for Demonstration

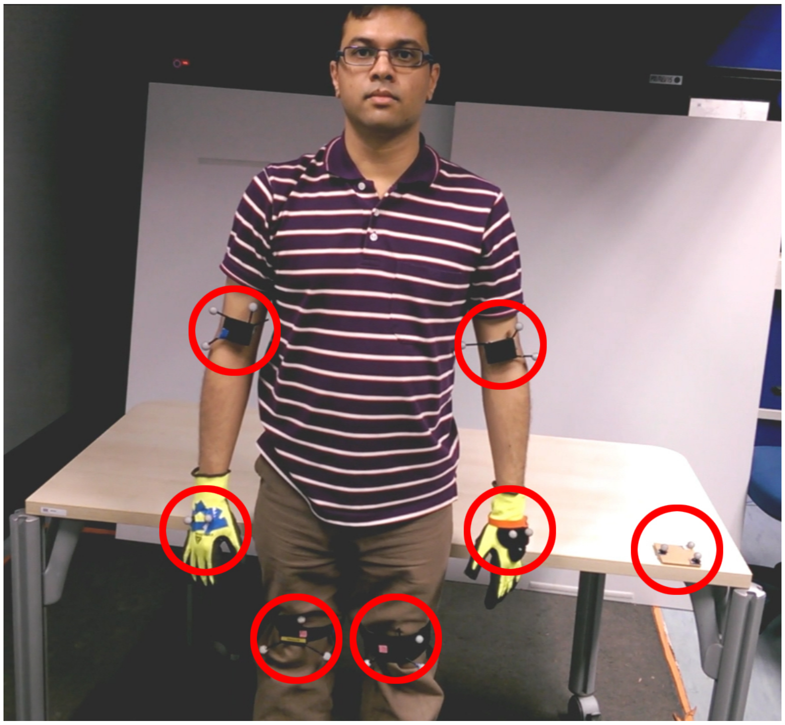

Fig. 3.8: Optical marker placements are indicated by red circles, which are used to track the marked limb positions such as both the left and right hands (with optical markers on the gloves), elbows and knees and also one on the table. The markers are tracked with an Opti-Track motion system.

We use optical markers along with an Opti-Track motion tracking system for recording the task demonstrations. These tracked markers can record the contact positions more accurately to capture the interactions with the obstacle. These markers are placed close to both the left and right knees, hands (using gloves with optical markers), elbows and one is also placed on the obstacle to mark the relative reference for the positions recorded, as shown in Figure 3.8.

3.2.3 Demonstrations for the Climbing Task

We record the task demonstrations with a human subject, to obtain a contact strategy for climbing an obstacle task. Here we show a demonstration involving a table used as an obstacle for the climbing task. The dimensions of the table, $l = 3.8m, b = 0.8m, h = 0.65m$ are such that the person can easily use both their arms and legs to climb on it. A demonstrator illustrates the climbing task to be performed by the humanoid robot, as shown in Figure 3.9.

The Opti-Track system tracks the marker motions at a fixed interval of 10ms, and it saves both the position and orientation information for all the markers. The demonstration starts with the person standing in front of the obstacle, then makes 2 contacts with hands (in

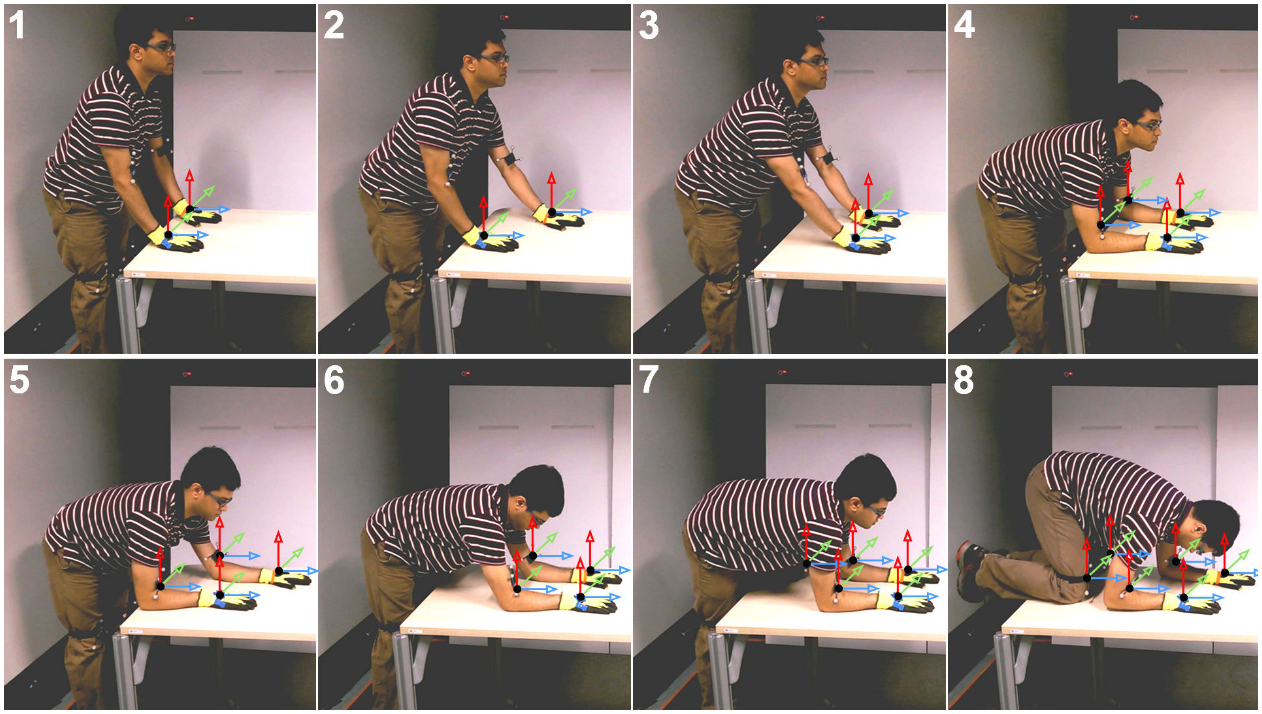

Fig. 3.9: The demonstration of climbing on a table obstacle at different contact stages. The demonstrator shows enough intermediate steps to accomplish the task.

pictures 1 to 3), then 4 contacts at both left and right elbows and hands (in pictures 4 to 6), then 5 contacts and later 6 contacts at both left and right hands, elbows, and knees with the obstacle. The recorded dataset are tagged off-line to indicate the contact states transpired during the task demonstration. This process of tagging the data allows us to discretize the recorded movements into different contact stages precisely as shown in the series of snapshots contained in Figure. 3.9. The contact sequences resulting from the data tagging enables us to devise a multi-contact strategy for a climbing task with full-body motions for humanoid robot. We use both the tracked position and orientation data of the contacts to represent the contact reference positions for the task. These contact reference positions are used to initialize our multi-contact motion planning algorithm to generate optimized contact positions for COMAN to complete the demonstrated climbing task.

3.3 Conclusion

We introduced the problem statement of our thesis in this chapter. We described the assumption for the multi-contact task of negotiating over the obstacle. Planning for humanoid robots with DOFs might take an unacceptably long time to solve. Although sampling-based path planners can drastically reduce the search space needed to find some solution, they are not optimal and struggle with specific situations such as contacts because, since in general, all planners tend to avoid obstacles as a default norm in their planning process. However, we are considering the planning process for the multi-contacts in a humanoid robot. We discovered that finding an initial solution for the multi-contact task does provide insight into parameters and the feasibility of the task itself. We explored pseudo teleoperation and sampling-based planning methods to obtain a suitable solution, although the outcomes did not give any initial solution. Human demonstrations through a motion capture suit provided the initial solution through which we were able to assess the task feasibility, although the solution was not good enough and did not suitably capture the contact interaction with an obstacle. Therefore, we used optical markers to track the movement of the contacts during the demonstration of the task. A demonstration for obstacle-climbing task was obtained using the optical markers with an Opti-Track motion system.

CHAPTER 4

Multi-Contact Motion Planning

Humanoid robots research on multi-contacts based on non-periodic locomotion has advanced relatively slowly in the past decade. However, in recent years the state-of-the-art has improved mainly through research on movement synthesis and character animation for the performance of various human-like actions, such as getting up, running, doing handstands etc. (Mordatch et al. (2012)), and motion planning through analysis of various multi-contact strategies like traditional forward facing, backward or sideways facing ladder climbing with hand grip contacts on the support rails (Vaillant et al. (2014)). The ability to plan for multi-contacts with the surrounding objects in the environment is crucial for humanoid robots to successfully egress from a car and climbing stairs, as demonstrated in the DARPA Robotics Challenge (Fallon et al. (2014), Johnson et al. (2015), Atkeson et al. (2015)).

Negotiating challenging environment needs better locomotion strategies like making use of obstacles as support points via multiple contacts, to get through inaccessible regions of the environment. In our proposed multi-contact motion planning, the humanoid robot is not restricted to only using its feet for support, but can also use other robot body parts like hands, elbows, and knees as supporting contacts. Furthermore, the availability of multiple degrees of freedom in the humanoid robots can be advantageous if exploited well to overcome the obstacle effectively. Essentially the humanoid robot can make transitions from 2 contact points at the feet, to 4 or 6 contacts at the hands and elbows of its arms, which may increase the overall robot stability and allow movements that are not possible.

In this chapter, we present a case of non-periodic locomotion where the COMAN climbs onto a large wooden obstacle with multiple contacts at various locations over the obstacle during the process of climbing it. The support polygon of the robot increases with every additional contact at robot foot, hand or any another body part of the kinematic chain that with the surrounding object. In such multi-contact tasks, the advantage is that the center of mass of the robot can move freely with the event of new contacts.

Generally, global planners such as A* search algorithm along with heuristics, RRT planning algorithms and PRM algorithms are employed for the multi-contact planning (Englsberger et al. (2014), Chung and Khatib (2015)). Due to the shortcoming of these global planners for the multi-contact motion planning as explored in the previous chapter, our multi-contact motion planner is initialized with a human demonstration of the task in the real world. The task demonstration gives us a proper strategy for the multi-contact planning and provides a clear way of specifying the goal for the climbing task. The planner uses the demonstrated contact plan to train with, as an initial set of contacts for the humanoid robot to interact with the obstacle for the multi-contact task.

4.1 Multi-Contact Planning Algorithm

We propose a multi-contact motion planner based on the whole-body control approach, which can directly work with the analyzed human demonstration data to formulate and guide the multi-contact search towards an optimal solution, while adapting the task demonstrations to the humanoid robot. The multi-contact motion planning algorithm exploits both the multi-contacts and the whole-body motions using both the upper and lower body of the robot in the task. The multi-contact approach allows any robot body surface to be in contact with the obstacle, as illustrated with a task demonstration for the robot. An overview of the multi-contact motion planner is shown in Figure 4.1.

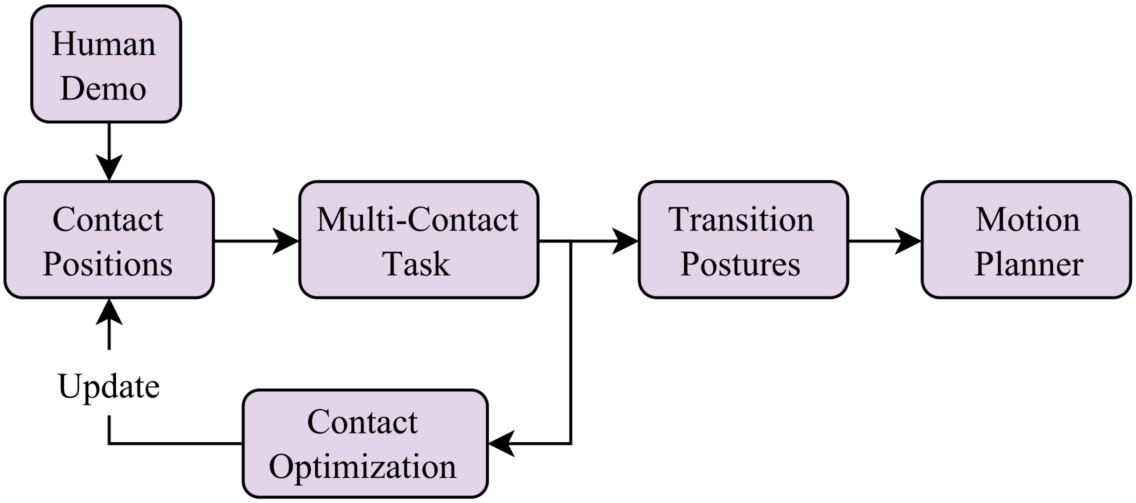

Fig. 4.1: Overview of Multi-Contact Motion Planner

The data collected from the Human Demo gives a contact plan for the multi-contact motion planner with an initial reference Contact Positions for the contact optimization. The multi-contact planner consists of two function blocks i.e., one is to formulate a Multi-Contact Task which solves for the IK solutions given the contact plan of the task and another is Contact Optimization which optimizes the reference contact position based on the contact stability costs to generate optimized solution postures. The Transition Postures generates the necessary contact transition postures between the optimized solution postures. Then a Motion Planner generates a collision-free trajectory i.e., the final optimized solutions for the multi-contact task. The steps involved in the multi-contact motion planning algorithm are listed below.

Input: Contact sequences $\mathcal{C}_i$ and initial solutions $\mathcal{Q}_i$: $(\mathcal{C}_0, \mathcal{Q}_0), (\mathcal{C}_1, \mathcal{Q}_1), \ldots, (\mathcal{C}_n, \mathcal{Q}_n)$ and contact positions.
Output: Optimized solutions $(\mathcal{C}_1, \mathcal{Q}_1^*), (\mathcal{C}_2, \mathcal{Q}_2^*), \ldots, (\mathcal{C}_n, \mathcal{Q}_n^*)$ and contact positions.

1. Pick the next contacts and positions from the contact sequence, $(\mathcal{C}_{i+1}, \mathcal{Q}_{i+1})$.
2. Formulate the multi-contact task to get IK solution for the next contacts, $\mathcal{C}_{i+1}$.
3. Optimize the contact positions over stability costs, contact constraints and collision constraints to find optimum contact positions and solution, $\mathcal{Q}_{i+1}^*$.
4. Update all the contact positions in the contact sequence by repeating steps 1 to 3.
5. Generate contact transition postures and plan collision-free motions.

Each time the algorithm is called with the next set of contact positions from the contact sequence as in step 1. After formulating the multi-contact task in step 2, an update process optimizes the contact reference positions while minimizing the stability costs and satisfying the contact constraints (i.e., to maintain contact positions on the obstacle) whereas collision constraints to avoid robot inter-body collisions. After repeating the process of optimization for all contact sequence, in step 5, it generates a final collision-free trajectory for the contact transition postures.

Next, we describe in depth the functionality of each block starting from the Multi-Contact Task block to the final output generated from the Motion Planner block. We present an application involving multi-contact locomotion, i.e., climbing over a large obstacle task, to describe our multi-contact motion planner. Finally, we show the simulation outputs followed by the experimental results obtained from the multi-contact motion planner.

4.2 Formulation of Multi-Contact Task

The multi-contact tasks are defined by the desired contact reference positions of the respective robot body links as demonstrated. The COMAN has multiple kinematic chains over the whole body such as legs and arms which also share the torso chain in the upper body. Hence, describing a multi-contact task in a humanoid robot requires multiple Cartesian tasks to be formulated and solved together to generate a whole-body solution. These multi-contact tasks are built with the whole-body control framework first presented in (Khatib et al. (2008)). The whole-body solutions for humanoid robot can be solved using Quadratic Programming QP-based solvers (Rocchi et al. (2015)). We use OpenSoT control library which employs stacks of such tasks to be solved, to generate whole-body inverse kinematic solutions for the robot.

We assign a Cartesian task T_i for each contact task, defined with their associated Jacobians J_i and the reference error functions e_i represented as in (4.1). Then, these Cartesian contact tasks are formulated in a QP form and optimized by minimizing the reference errors with the desired Cartesian reference positions and orientations to obtain the joint solutions for the humanoid robot.

$$T_i = (J_i, e_i) \tag{4.1}$$

We propose a multi-contact objective with QP optimization to generate a whole-body solution for the task as in (4.2). The multi-contact objectives represent the collection of Cartesian tasks for the contact points. The objective of the multi-contact is met by minimizing the task errors while being subject to bounds and constraints defined in (4.3). The multi-contact objectives are subject to joint limit bounds such that the joint solutions are bounded with the joint limits for the robot. Whereas constraints are enforced on the resulting contact positions for collision avoidance. Also objectives can be subjected to task prioritization with either soft or hard priorities.

$$\min_{\text{q}} \sum \text{multi-contact objectives} \tag{4.2}$$

$$\text{subject to} \begin{cases} \text{joint limits} \\ \text{task priorities} \\ \text{internal collisions} \end{cases} \tag{4.3}$$

The formulation of Multi-Contact Task consists of combining multiple tasks for the robot and then solve for desired contact inputs to generate the whole body solutions for humanoid robot. We select the multi-contact tasks for the respective robot body links, shown in the contact sequences (1 to 8 of Figure **??**) from the demonstrations. To explain the multi-contact task formulation in our algorithm, first we list a few contact sequence/sets denoted as $\mathcal{C}_n$, with their contact states from the demonstrations as follows:

1. $\mathcal{C}_0 = \{$LFoot, RFoot$\}$
2. $\mathcal{C}_2 = \{$LFoot, RFoot, LHand, RHand $\}$
3. $\mathcal{C}_4 = \{$LFoot, RFoot, LHand, RHand, LElbow, RElbow$\}$
4. $\mathcal{C}_7 = \{$RFoot, LHand, RHand, LElbow, RElbow, LKnee$\}$
5. $\mathcal{C}_8 = \{$LHand, RHand, LElbow, RElbow, LKnee, RKnee$\}$

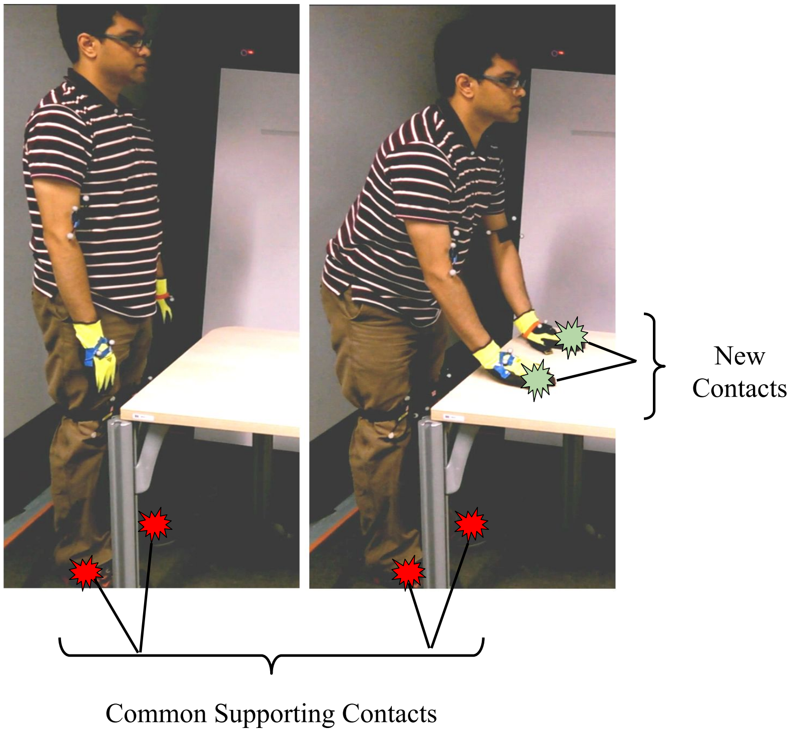

Fig. 4.2: The demonstrated contact sequence for climbing a table obstacle: initial state (left) and 2nd contact sequence (right).

Now, we need to identify the multi-contact tasks required to be defined from the contact sequences. The contact sequences representing the initial state and the next contact sequence are shown in Figure 4.2. The inter-connecting support links between the adjacent contact sequences are necessary for generating the transition phase and also feasible motions. The common supporting links, denoted as $\mathcal{C}_{sup}$, can be found using intersection of the two contact sets $\mathcal{C}_1$ and $\mathcal{C}_2$ as in (4.4).

$$\mathcal{C}_{sup} = \mathcal{C}_1 \cap \mathcal{C}_2 \tag{4.4}$$

The common supporting links between the contact sequences in Figure 4.2 are given by their contact sets intersection, $\mathcal{C}_{sup} = \{\text{LFoot, RFoot}\}$. We can select any link l_i from the supporting links (i.e., $l_i \in \mathcal{C}_{sup}$) as a reference frame link to express the remaining contact positions w.r.t. to it. For instance, if we select RFoot as the reference frame link l_i, the Cartesian contact tasks $\mathcal{C}_t$ to be formulated for the next contact sequence is given as in (4.5) and the resulting contact tasks are $\mathcal{C}_t = \{\text{LFoot, RHand, LHand}\}$. Thus, we have the desired contact positions at the left foot (LFoot), right hand (RHand) and left hand (LHand), and we denote the corresponding contact tasks as T_{LFoot}, T_{RHand} and T_{LHand}. The Cartesian contact tasks $\mathcal{C}_t$ are defined using their associated Jacobians described in the RFoot reference frame with the corresponding error vectors e_i as in (4.6), (4.7) and (4.8).

$$\mathcal{C}_t = \mathcal{C}_2 - l_i \tag{4.5}$$

$$T_{\text{LFoot}} = \left(J^{\text{LFoot}}_{\text{RFoot}}, e_1\right) \tag{4.6}$$

$$T_{\text{RHand}} = \left(J^{\text{RHand}}_{\text{RFoot}}, e_2\right) \tag{4.7}$$

$$T_{\text{LHand}} = \left(J^{\text{LHand}}_{\text{RFoot}}, e_3\right) \tag{4.8}$$

4.2.1 Combining the Multi-Contact Tasks

To formulate the multi-contact task, we need to combine the multiple tasks given by $\mathcal{C}_t$ in a meaningful way. Here, although some of the tasks are part of the unconnected kinematic chains which can be solved independently, it is necessary to have inter-dependencies in the task formulation to be defined between them, to enable searching whole-body solutions. Some of the tasks can be easily combined, such as the arms and the legs task. This is due to their symmetrical aspects and their strong inter-kinematic relationship. We combine the left and right hand tasks with a combination task as T_{comb} in (4.9) with the concatenation of the corresponding task Jacobians and error vectors as in (4.10).

$$T_{\text{comb}} = (J_{\text{comb}}, e_{\text{comb}}) \tag{4.9}$$

$$J_{\text{comb}} = \begin{bmatrix} J^{\text{LHand}}_{\text{RFoot}} & J^{\text{RHand}}_{\text{RFoot}} \end{bmatrix}, \quad e_{\text{comb}} = [e_3 \quad e_2] \tag{4.10}$$

In addition to this, the Cartesian tasks can be combined using prioritization in two different ways namely, hard priority and soft priority. On the other-side, we can create an individual task for the left foot to be solved independently as in (4.6). Since, the common supporting links, $\mathcal{C}_{sup}$ indicates that the legs support the person as verified in Figure 4.2, while including the leg task in the multi-task formulation, we assign it a higher priority than the hand tasks. Here, by selecting hard task prioritization to combine the leg task and hand tasks, with high priority for the leg task, we can force the solver to guarantee that the positions of the supporting leg contacts must be strictly satisfied. Similarly we can define priority tables as the new contacts occur during the task demonstration by computing their common support links $\mathcal{C}_{sup}$ as in (4.4). The resulting priority tables for new contacts occurring at the 4th, 7th and 8th contact sequences shown in Figures 4.3, 4.4 and 4.5 are listed in Tables 4.1, 4.2 and 4.3.

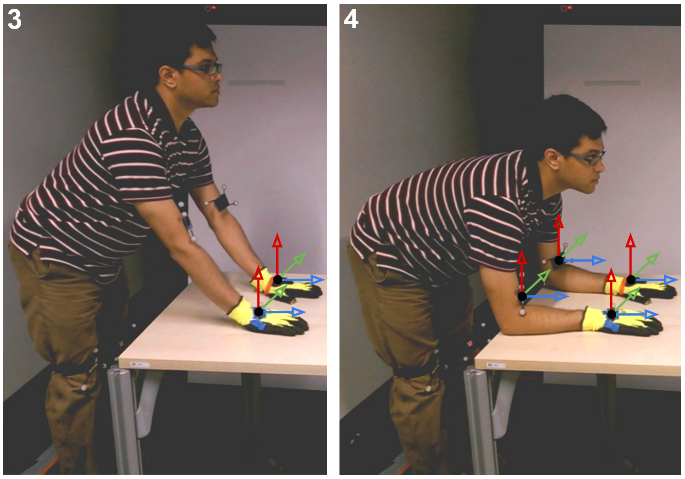

Fig. 4.3: Climbing a table obstacle, 3th contact sequence (left) and 4th contact sequence (right)

Table 4.1: Priority table for 4th contact sequence

Task	Priority
LFoot, RFoot	High
LElbow, RElbow	Low

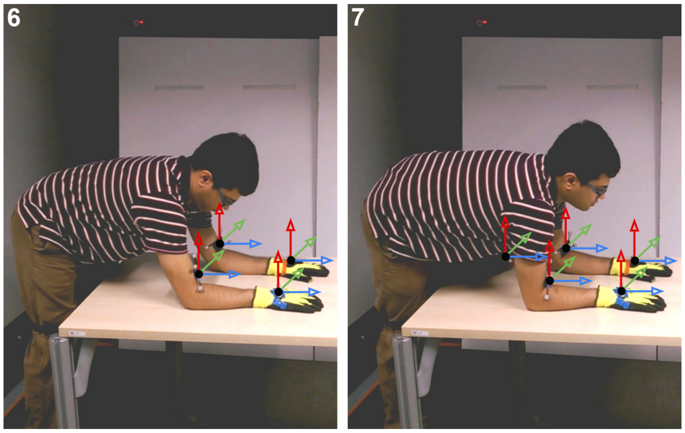

Fig. 4.4: Climbing a table obstacle, 6th contact sequence (left) and 7th contact sequence (right)

Table 4.2: Priority table for 7th contact sequence

Task	Priority
LElbow, RLelbow, RFoot	High
LKnee	Low

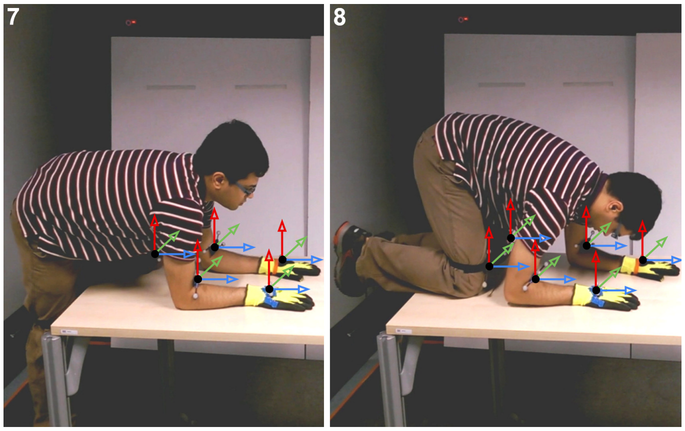

Fig. 4.5: Climbing a table obstacle, 7th contact sequence (left) and 8th contact sequence (right)

Table 4.3: Priority table for 8th contact sequence

Task	Priority
LElbow, RElbow, LKnee	High
RKnee	Low

We see that the top priority for the tasks moves from the feet to the elbows in the arms at the 7th contact stage of climbing on the obstacle as demonstrated. This is because the arms start to support the robot as it climbs on the obstacle. The higher priority for the tasks are assigned by the computing the common support links $\mathcal{C}_{sup}$ and lower to the remaining contact links. Finally, at the 8th contact stage shown in Figure 4.6, all the robot body links in contact have the high priority except for the RKnee given by Table. 4.3.

Multiple contacts in the same kinematic chain:

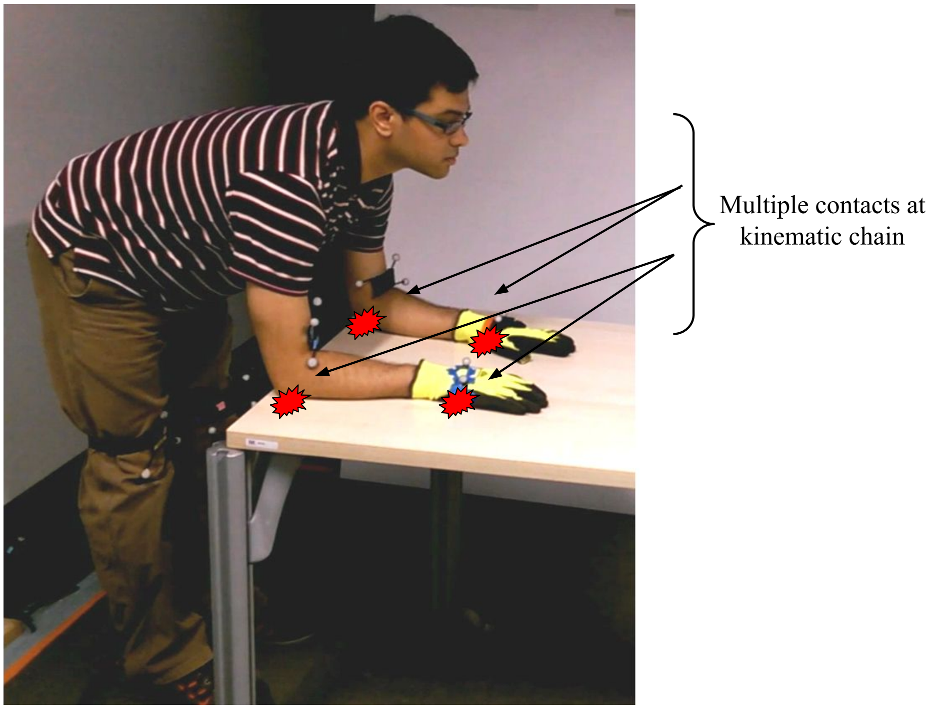

Fig. 4.6: Multiple contacts occurring at the elbows and hands of the same kinematic chain arm for the 4th contact sequence in the demonstration.

We can have instances of multiple contacts in the same kinematic chain. For example, in the 4^{th} demonstrated contact stage shown in Figure 4.6, we have contacts at both the elbow and hand which are parts of the same kinematic chain arm, i.e., both the left and right arms. If these multiple contact tasks in the same kinematic chain tasks are combined by concatenating their associated Jacobians to be solved as in (4.11), this forces the IK solver to satisfy both the references simultaneously. However, it is not possible for most of the cases to find such solutions.

In such cases, we identify the robot body link in the kinematic chain which plays a significant role in supporting the shoulder mass (i.e., for the arm kinematic chains). So to ensure that at least one of the contact tasks is satisfied, a higher priority must be assigned to it. However, assigning hard priorities for the contact tasks in the same kinematic chain results in a poorly formulated QP problem. Instead of combining the contact tasks with hard priorities, we can combine them with soft priorities. This can be achieved by weighting their task Jacobians according to the defined priority. The multi-contact tasks with soft priorities can be solved together as in (4.12).

$$J_{comb} = [J_1 \quad J_2] \tag{4.11}$$

$$J_A = \begin{bmatrix} W_1 J_1^T & W_2 J_2^T \end{bmatrix} \tag{4.12}$$

where W_1 and W_2 are diagonal weighting matrices through which soft priorities can be assigned to the tasks, by setting their diagonal elements to a positive value. The Jacobians for the elbow and hand contacts are represented as J_1 and J_2 respectively.

Our algorithm automatically chooses priorities for tasks in QP form to get the IK solutions. The key idea is that in general, the higher up in the kinematic chain we have a contact (i.e., if the contacts are closer to the waist/pelvis region of the robot), the more support exclusively falls under the contact link closer to the pelvis joint. We define a precedence which establishes the rules to assign soft priorities to the task when multiple robot body links of the same chains are in contact as in Table. 4.4.

Table 4.4: Precedence table for Kinematic chain

Precedence	Low		High
Kinematic Chains			
Arms	Hands	Elbows	Shoulders
Legs	Feet	Knees	Hips

We call the contacts with the highest precedence as a dominant contacts, since its position is essential for stability. In general, weighting coefficients for the elbow W_1 and wrist W_2 contact case can be set according to priority order as in (4.13).

$$W_1 > W_2 > 0 \tag{4.13}$$

Particularly due to the off-line nature of contact planning used, we use fixed weighting coefficients for the tasks. We set the diagonal elements to 1.0 for the contacts at the elbows and to 0.1 for the contacts at the hands.

Postural task: Infinite solutions exist in hyper-redundant robots like the humanoid robots with more than 6 degrees of freedom. Although, robots with more degrees of freedom (> 6 DOFs) has many advantages, but the most troubling disadvantage is the certainty of having infinite solutions to choose from. This uncertainty in finding an apt solution can impinge on robustness necessary for searching meaningful and consistent solutions for a humanoid robot. Here, we can use the observed motion data from the Xsens motion capture suit to be applied as a posture hint in the multi-contact task formulation. This also enables the robot to mimic some of the observed human movements during the demonstration. The posture task uses the mapped joint angles from the motion suit as a joint reference and forces the optimizer to find solutions close to it.

$$T_{posture} = (I, \quad \dot{q}_{posture} + K_q(q_{posture} - q)) \tag{4.14}$$

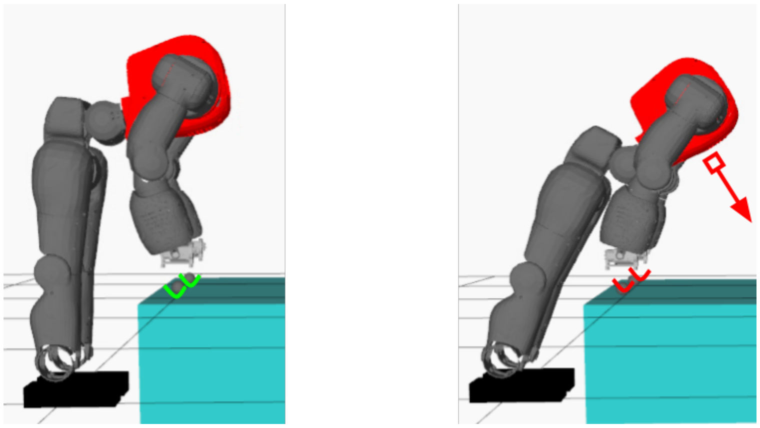

Fig. 4.7: In left, the postural task is applied for the multi-contacts at the hands with hints from motion capture data. The resulting new contacts at the left and right hands are stable and supports the shoulder masses completely. However, right picture shows with no postural task for the multi-contact tasks. The two contacts at the left and right hand are unstable and robot may fall in the direction shown.

A posture task helps to keep the IK solutions, q, generated for contact position references to stay close to the desired posture $q_{posture}$. The posture task, $T_{posture}$ is defined with a proportional gain, Kq and identity Jacobian, I as in (4.14). The postural task is incorporated into the multi-contact task formulation as a lower priority task. The postural task at low priority allows the solver to find unique solutions while staying close to the desired postures (hints from motion capture), while also satisfying the multi-contact task reference goal requirements.

4.2.2 Internal Collision Avoidance

Internal collision avoidance for the humanoid robots is vital for avoiding any unwanted internal collisions of the robot with itself. We approximate the humanoid robot COMAN body links with simple geometrical shapes like squares, cylinders, and spheres. Efficient approaches to limit the collision checking to only a few robot body links by pruning irrelevant pairs was first presented in (Kuffner et al. (2002)). To allow efficient internal collision avoidance, we select only a few robot link bodies as in Figure 4.8 for collision checking between

1. Identical robot link bodies in the left and right arms like elbows, forearms, upper arms.
2. Identical robot link bodies in the left and right legs like feet, knees and lower thighs.
3. Elbows and knees of arms and legs, respectively.

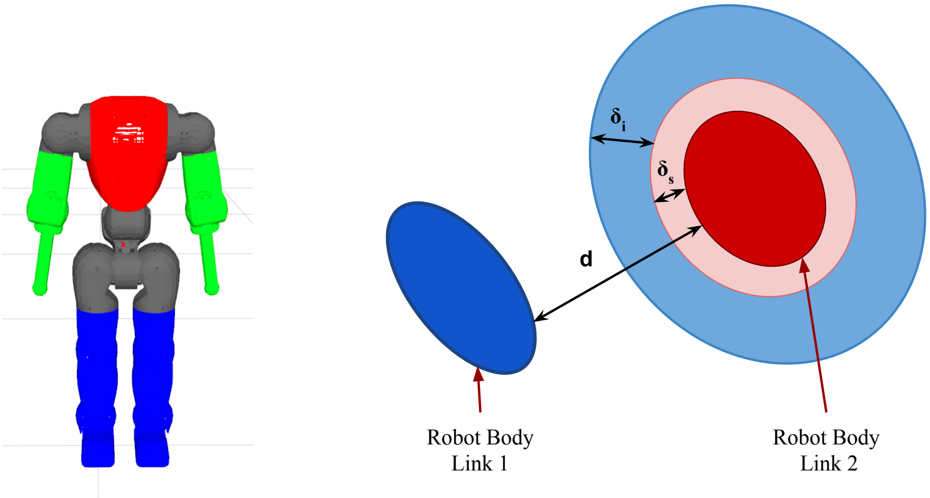

Fig. 4.8: Selected robot body links (right) in the arms: elbows, forearms are shaded as green and in the legs: feet, knee and thigh are shaded as blue. In left image, we depict the parameters necessary for collision avoidance between robot link body 1 shaded blue and robot link body 2 shaded red. The region of width δ_i (shaded light blue) close to robot body 2 is the region where the collision avoidance mechanism is initiated. However, the restricted region of width δ_s when the collision can occur at robot body is shaded as light red.

We use the off-the-shelf Flexible Collision Library (FCL) to detect collisions between two geometrical bodies. FCL is used for collision checking which outputs the closest points on the bodies against which collision checking takes place (Pan et al. (2012)). The collision checking between two robot link body shapes, detects for collisions between the closest points cp_1, cp_2

on these objects. We can measure the distance d between those closest points on the robot bodies as

$$d = ||cp_1 - cp_2|| \tag{4.15}$$

Consider the collision between the robot body link 1 and the robot body link 2 as in Figure 4.8 with d as the distance between the two bodies. The relative velocity of the one link with respect to the other is constrained along a direction connecting the closest points on the two robot body links, which is called velocity damping for the collision avoidance mechanism. We enforce a position constraint on the relative velocity along this direction vector $\vec{n}$ as

$$\vec{n}\left(J_{\mathrm{cp1}} - J_{\mathrm{cp2}}\right)\dot{q} < \epsilon \frac{\delta_i - \delta_s}{\Delta t} \tag{4.16}$$

where J_{cp1} and J_{cp2} are the Jacobians of the nearest points cp1 and cp2 on respective the robot body link 1 and robot body link 2. δ_s is the critical distance at which collision can occur and δ_i is distance at which the collision avoidance mechanism starts to oppose the collision. ϵ is the gain used to reduce the relative approaching velocity.

A low value of gains would jeopardize the robot's structural integrity, while a larger one could overly constrain its motion, which would reduce the task execution time and have an adverse effect on the negotiation of the different obstacles. The gains are tuned accordingly, to facilitate to smoothly reduce the relative approaching velocity at which the threshold distance is reached. We define position constraints for all the robot body pairs listed before, to enforce internal collision avoidance for the IK solution obtained.

The complete set of rules for the Multi-Contact task formulation from the contact sequence used by our multi-contact motion planning algorithm can be listed as follows:

1. Identify the tasks from the contact sequence given by $\mathcal{C}_t$ in (4.5), after selecting a common reference in $\mathcal{C}_{sup}$.
2. Impose whole-body behavior through the task Jacobians expressed in the common reference frame.
3. Combine the tasks at the same symmetrical level using (4.9): i.e. tasks in the arm chain or tasks in the leg chain can be combined.
4. Prioritize the tasks given by the common support links, $\mathcal{C}_{sup}$ using hard priority.
5. If multiple tasks are present in the same kinematic chain, we can assign soft priorities according to Table. 4.4
6. Add the postural task with solution hints obtained from the motions recorded.