

# Climbing over Large Obstacles with a Humanoid Robot via Multi-Contact Motion Planning

Pavan Kanajar<sup>1</sup>, Darwin G. Caldwell<sup>1</sup> and Petar Kormushev<sup>2</sup>

**Abstract**—Incremental progress in humanoid robot locomotion over the years has achieved important 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-body contact motion planning guided through physical human demonstrations. Our goal is to make the 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 whole-body motion and multi-body contacts including both the upper and lower body limbs. The proposed motion planning algorithm is applied to a challenging task of climbing over a large obstacle. We demonstrate successful execution of the climbing task in simulation using our multi-contact motion planning algorithm initialized via a transfer from real-world human demonstrations of the task and further optimized.

## I. INTRODUCTION

Research on humanoid robot locomotion has mainly focused on motion pattern generators for bipedal walking gaits [1]. Using state-of-the-art bipedal motion pattern generators, it is now possible to navigate complex scenarios including negotiate obstacles or rubble mainly through stepping-over methods, climbing stairs and avoiding obstacles altogether by circumnavigating them in the environment [2]. The current state-of-the-art methods only allow humanoids to cross over obstacles of height less than the knee length of the robot while avoiding any external contact with the obstacle in the process. Also, any other contact with the surrounding environment is strictly avoided during the process and motion planners explicitly generate paths that incorporate safety distance margins to avoid such collisions. Due to this, humanoid robots tend to fall short of the many expectations to execute tasks as well as humans in disaster situations, for example.

Through multi-body contact locomotion a humanoid robot can gain instantaneous increase in the stability region for balancing and manage to overcome large obstacles in the surrounding environment. The main novelty of our proposed approach is that the robot would be able to overcome larger



Fig. 1: The humanoid robot COMAN attempts a climbing task to get on top of a wooden obstacle.

obstacles than previously possible by climbing and crossing them over using multi-body contacts, as illustrated in Fig. 1.

Multi-body contact locomotion is a type of locomotion where the humanoid robot is not restricted to only using its feet for support, but also can use other body parts for contact such as hands, elbows and knees. The availability of multiple degrees of freedom in humanoid robots can be advantageous if exploited well to effectively overcome obstacles.

The support polygon of the robot increases with every additional robot foot, hand or other part of the kinematic chain that is in contact with environment. By defining novel stability measures for newly made contact similar to friction cone constraints, we develop a contact motion generation algorithm with several contact constraints that is able to overcome and cross a large obstacle in the path of the humanoid robot. Our approach is bootstrapped by a human demonstration of a similar climbing task in the real world.

A motion capture experimental setup is used to record the human motion data for the climbing task, and to specify and show the task goals for the humanoid robot to be performed. Through data analysis, rich data about the contact sequence and configuration information is extracted for solving the motion planning problem. We use the humanoid robot COMAN (shown in Fig. 2) which has 23 degrees of freedom (DoFs) for experiments in simulation. The size of the humanoid robot COMAN approximates the dimensions of a 4-year-old child. The height of 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

<sup>1</sup>Department of Advanced Robotics, Istituto Italiano di Tecnologia, Via Morego 30, 16163 Genova, Italy. pavan.kanajar@iit.it

<sup>2</sup>Robot Intelligence Lab, Dyson School of Design Engineering, Imperial College London, South Kensington, London SW7 2AZ, UK. p.kormushev@imperial.ac.uk

weighting 18.5 kg and the torso and the arms weighting 12.7 kg. One unique feature of COMAN is its passive compliance in the legs, which makes it more robust but also more difficult to control. Special care needs to be taken to adapt the conventional ZMP-based walking generator for the compliant legs [3].

We propose a multi-contact search algorithm based on the whole-body control approach which can directly work with the analyzed human motion data to formulate and guide the multi-contact search towards the optimal solution while adapting the human demonstration to the humanoid robot's body. This approach allows any robot body part to make contact with the obstacle, as it only depends on the recorded data observations.

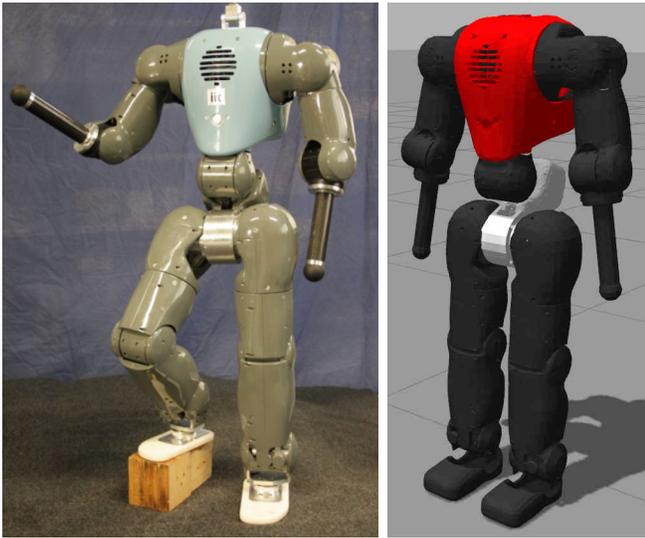


Fig. 2: The humanoid robot COMAN is shown on the left side, and on the right is its simulated version in the Gazebo simulation environment.

## II. RELATED WORK

OpenSoT presented a whole-body control library for humanoid robot to generate whole-body motion for the Cartesian space tasks. The library implemented a cascade of Quadratic Programming (QP) problems suitable for solving hierarchical IK problem on humanoid robot along with constraints and bounds [4].

Contact-consistent Elastic Strips (CES) framework was proposed for humanoid locomotion in unstructured environments, where the environment scene was scanned for contacts to choose candidate support regions [5]. 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 environment to only wrists and feet to increase support during locomotion, whereas we try to not limit the contact to only a few body parts.

A humanoid robot while maintaining balance on its feet is taught to complete a board cleaning task through a kinesthetic demonstration of the task was presented in [6]. The robot learns to produce appropriate interaction forces

with the arm to clean the board. An online optimization of the footsteps placement for gait generation for humanoid walking in the presence of external disturbances like pushes was presented in [1]. Real-time imitation of full-body motion of human demonstration on humanoid robot COMAN while maintaining stability was presented in [13]. The imitation motions of COMAN where constrained using prioritized task control to maintain stability.

Igor [7] introduced contact invariant optimization (CIO) method to synthesize different complex tasks on a humanoid like character which involved contact with hands and feet while generating human-like behaviors for getting up, crawling and climbing tasks via multi-contacts with the objects in the surroundings environment. Here they already predefine the surface patches on the humanoid character over which the contacts can appear or disappear over time during the tasks.

A humanoid robot learning to perform a forehand swing with a tennis racket from human demonstrations was presented in [8]. The demonstrated movement trajectories are encoded with control policies with a set of nonlinear dynamical system. Since these robot programming method by demonstration only learns control policies in the joint space, it fails to capture interactions in the Cartesian task space especially for floating base humanoid robot under multi-contact in locomotion tasks.

Kuffner [9] presented a dynamical stable motion planning for a humanoid robot placing foot above an obstacle while keeping balance of the robot. Here their motion planner used RRT algorithm to search for a solution path from a random set of statically-stable configurations which also lies within the obstacle free configuration space while checking for the balance of the humanoid robot. Whereas here we get our set of desired poses from human demonstrations of the task for searching the optimized solution. Although we only present a statically stable solution for the climb task over large obstacle.

Zucker presented a continuous trajectory optimization for opening a door with a humanoid robot HUBO+. Special emphasis was placed on generating smooth trajectories to minimize unnecessary motion of the robot [10].

Robust walking for humanoid robot Atlas through an online quadratic whole-body optimization generating joint torque for fast locomotion was presented in [11]. They incorporated a quadratic convex problem within planner which optimizes the future steps of the humanoid robot to generate robust footsteps against external pushes and terrain variations.

Autonomous planning and control framework for robust climbing of a general ladder-like structures with humanoid robot DRC-Hubo where the robot uses arms to support perturbations while climbing was presented in [12]. In this framework whole-body motion trajectories were generated through motion-primitive guided, sampling-based adaptive planner for climbing the ladder. Also with a comparison of different ladder climbing strategies like backward facing, sideway facing and traditional forward facing robot for task

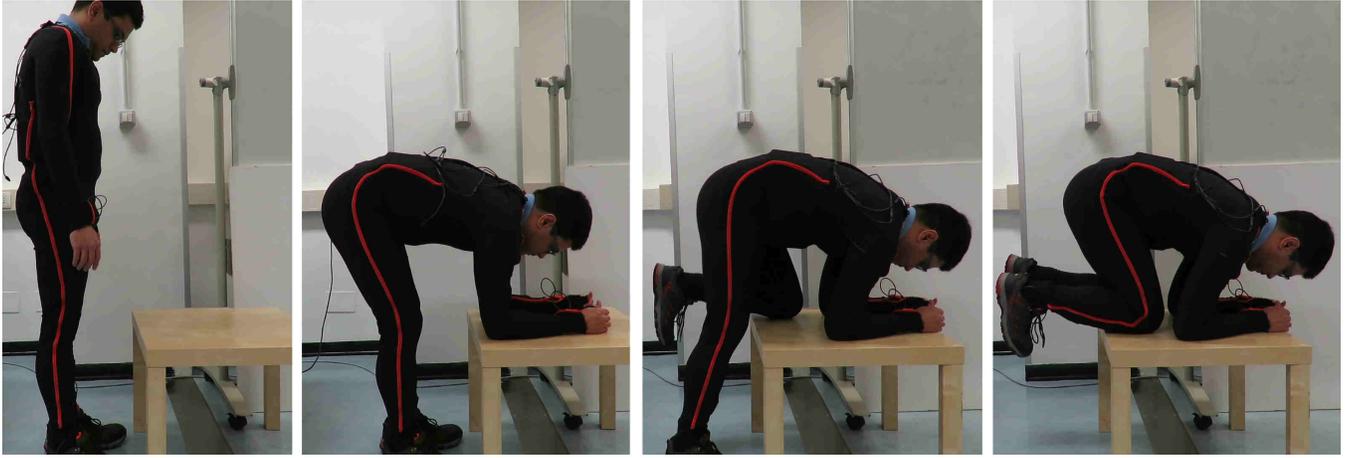


Fig. 3: Shows the demonstration of the climbing task with table as an obstacle. The demonstrator is wearing the Xsens bodysuit to record movements during the task. In the first picture we notice that the shoulder masses are completely supported by the elbows. Similarly in last picture we also notice that the upper thigh masses are completely supported by the knees.

execution are presented. In our climb approach over obstacle, through the demonstration of the task we are inherently specifying the best strategy to climb over the obstacle while also maximizing the support region for the robot through multi-contact during the task.

Kanajar [15] presented a mobile manipulator Neptune, to help motivate children with motor impairment to interact and encourage patients to work on their rehabilitation routines through interactive devices. This work was limited to a single arm demonstration while the current paper expands the interactive guidance to a full body humanoid robot.

### III. MOTION CAPTURE DATA COLLECTION AND MOVEMENT PLAYBACK ANALYSIS

We perform a human demonstration of the climbing obstacle task, where we choose a table as an obstacle to climb on. We select the table with height  $h = 0.35$  m, such that the person can easily make contact with the table obstacle using both the arms and legs in the process, in order to illustrate that usage of additional body contacts can help in climbing larger 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 and also the distance of the person in front of the obstacle. We then record the human demonstration of the climbing task with a full-body motion capture system like Xsens bodysuit. The demonstrations begin with the person standing in front of the table obstacle and performs the climb task as shown in Fig. 3.

#### A. Xsens Bodysuit

We use the Xsens bodysuit to record human demonstrations of the climb task. Xsens bodysuit has 18 motion sensors each measuring 3 degrees of freedom at each joint link spread over the bodysuit to capture the movements with a total of 54 DoFs. For mapping the Xsens motion data from the bodysuit to humanoid robot COMAN requires reduction in the number of DoFs recorded to 23 DoFs only. We here ignore the DoFs related to wrists, neck and we combine 9 DoFs in the upper

body as responsible for movement at the pelvis joint to 3 DoFs, which reduces the number of DoFs to 30. Also we can ignore the additional DoFs in the bodysuit sensors which the robot joints lack over certain body regions, bringing down the total recorded DoFs from 54 to 23.

The Xsens motion capture system is setup 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. We then compensate the Xsens movement data (joint angles) by applying the following computations.

1. Change reference from bodysuit frames to robot link frames according to sensor placements in the bodysuit.
2. Apply robot joint limits  $(\theta_{min}, \theta_{max})$  to the recorded motion angles.

We then use a direct mapping method to map human demonstrated movements to humanoid robot COMAN as a direct joint angle mapping.

$$\theta_{Xsens} \rightarrow \theta_{COMAN} \quad (1)$$

#### B. Playback Movements in Simulation

We use the direct mapping of demonstrated movements for humanoid robot given by (1) to playback the climbing task in simulation environment. The environment consists of an obstacle (wooden cuboid) along with a simulation version of humanoid robot COMAN as shown in Fig. 1. The robot is operated in a joint position control mode. The mapped joint positions from the demonstration is applied as joint position references to the humanoid robot COMAN. We see that the robot performs similar movements demonstrated during the climb task, but the robot does fail sometimes to complete the task.

This is due to the difference in the physical size between the humanoid robot and human demonstrator. So we had to adjust the obstacle height and position of the robot several times, in order to make sure that both the arms and legs could

make contact with the obstacle during playback of movements from the recorded motions of the climbing task. The successful attempt of climbing task over a wooden obstacle during playback of movements on humanoid robot COMAN in simulation is used to get references for describing the task. Although the robot nearly completes the task, there is instability and the robot fails to finish the task just by playback of the human movements as shown in Fig. 4. Thus, we need to optimize the contact references and motions for the task to guarantee successful execution of the climb task.

### C. Contact References for Climb Task

We use the data collected like robot positions, joint angles during the playback of movements in simulation to compute contact references. Instead of providing continuous joint angle references for the humanoid robot, we can discretize the movements in terms of contact position references for specifying the climb task now.

The humanoid robot links are approximated with geometrical shapes like square, cylinder and spheres, also the virtual environment objects are approximated with suitable geometrical shapes. We use off-the-shelf Flexible Collision Library [14] to detect collisions between two geometrical bodies.

We read the robot position and links positions at a fixed time interval 50 ms and compute collision detection for all robot links against the obstacle. Using this, a list of contact sequence is compiled with contact states, consisting set of active contacts and their corresponding contact positions for entire duration of the task demonstration. If there is a change in contact state only then we consider it as state in contact sequence.

We can build a contact sequence by adding active contacts or removing the inactive contacts as shown below.

| State1  | State2  | State3  | State4  | State5  | State6  |
|---------|---------|---------|---------|---------|---------|
| Lwrist  | Lwrist  | Lwrist  | Lwrist  | Lwrist  | Lwrist  |
| Rwrist  | Rwrist  | Rwrist  | Rwrist  | Rwrist  | Rwrist  |
| Lfoot   | Lelb    | Lelb    | Lelb    | Lelb    | Lelb    |
| Rfoot   | Relb    | Relb    | Relb    | Relb    | Relb    |
|         |         |         | Lknee   | Lknee   |         |
|         |         |         |         | Rknee   |         |
| $s > 0$ | $s > 0$ | $s = 0$ | $s > 0$ | $s = 0$ | $s = 0$ |

Here a couple of states are duplicated, because the position distance between contact states are not ignored and are encoded as a new state. Whereas here our intention is to encode only the effective movement responsible for the task, i.e., a unique set of contact state transitions and references (although discrete in nature). So here we make use of additional link (support) position to derive the information about the absolute stability at the contact state. We eliminate the duplicated intermediate state, by computing this stability measure described in the next section. The last row of the table show the states with stability measure(s) as  $s = 0$  for absolute stability or else as  $s > 0$ , we reduce the set of contact sequence with only states under absolute stability

to compile our effective contact sequence for executing the climbing task. Finally we obtain the reduced set of states containing the contact lists as shown below associated with contact position references. This reduced set of states are necessary to describe the climb task for humanoid robot with our multi-body motion planning algorithm.

| State1  | State2  | State3  |
|---------|---------|---------|
| Lwrist  | Lwrist  | Lwrist  |
| Rwrist  | Rwrist  | Rwrist  |
| Lelb    | Lelb    | Lelb    |
| Relb    | Relb    | Relb    |
|         | Lknee   | Lknee   |
|         |         | Rknee   |
| $s = 0$ | $s = 0$ | $s = 0$ |

## IV. MULTI-BODY MOTION PLANNING ALGORITHM

Here we propose a multi-body motion planning algorithm for optimizing the contact position references for a humanoid robot over stability and contact constraints for the climb task. The human motion data collected from observation serves as an initial solution for our algorithm, which is used to optimize the contact references for multi-contact motion planning of the task. Our motion planning algorithm computes IK solutions using a quadratic programming based solver (OpenSoT). The steps involved in searching contact position algorithm are listed below.

1. Inputs: Contact sequence and reference, Posture (joint configuration) from Xsens bodysuit, initial position of the humanoid robot.
2. Pick the next input in the list.
3. Task formulation to get IK solution.
4. Optimize contact references over stability costs, contact constraints and collision constraints.
5. Update all contact references in the state sequence by repeating steps from 2 to 4 to get

Each time, the algorithm is called with next of input, we optimize the contact reference positions while minimizing stability costs and satisfying contact constraint (i.e., to maintain contact positions on the top of obstacle ) whereas collision constraints at contact points on table sides (i.e., to avoid contact) through obstacle avoidance mechanisms.

### A. Task Formulation

The task formulation builds on whole-body control library [4]. The task formulation process consists of combining different parts of the robot links with desired contact and each task is applied with an operational reference position to satisfy. We focus only on the inverse kinematic solutions, where each Task  $T_i$  is represented with its associated Jacobian  $J_i$  and its reference error function  $e_i$ . The error in the task reference is minimized while optimizing the joint solutions.

$$T_i = (J_i, e_i) \quad (4)$$

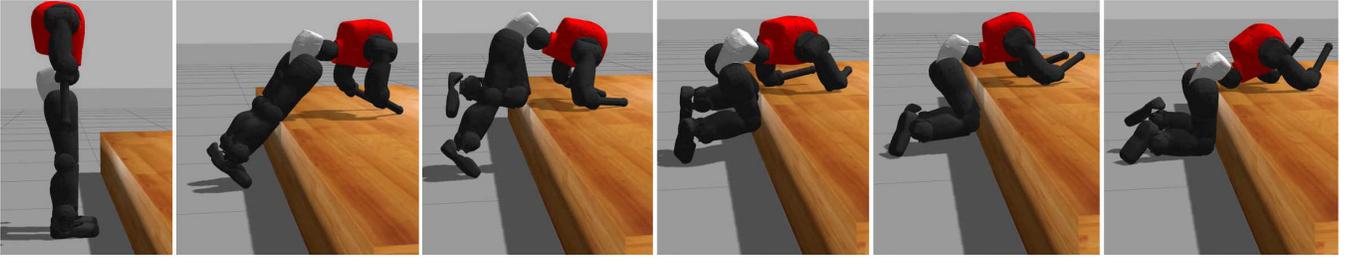


Fig. 4: Direct playback of the mapped movements from a human demonstration onto humanoid robot COMAN to perform the climb task over wooden obstacle in Gazebo simulation. The robot fails to climb on top of the wooden obstacle.

The observed contact sequence is used how to determine the tasks must be formulated. Using the recorded input contact references, we define a task for each contact reference as (4). For example if we have contact references at right wrist ( $R_{wrist}$ ) and left wrist ( $L_{wrist}$ ), we can denote the corresponding tasks as  $T_{R_{wrist}}$  and  $T_{L_{wrist}}$ . If we have contact references for both wrists at the same time we can combine the tasks as  $T_{comb}$  by concatenating their corresponding Jacobians  $J_i$  and errors  $e_i$  to solve both the tasks at the same time, which is given as

$$T_{comb} = T_{R_{wrist}} + T_{L_{wrist}} \quad (5)$$

For every input contact reference we consider current and previous list of active contacts to guarantee inter-connectivity between the two postures. We achieve this by computing the common support link between the adjacent contact sequences and by using this common support link as a frame of reference in the task definition. For example with right foot ( $R_{foot}$ ) as common support link, we modify tasks in (5) to represent the jacobians of the task with  $R_{foot}$  as reference frame.

Solving IK for a robot with more than 6 DoFs have many advantages but the most troubling disadvantage is the certainty of having infinite solutions to choose from, which can throw off robustness in searching solution for a humanoid robot. So we use the observed motion data from the Xsens suit to be applied as a posture hint in our task definition. This enables us to also mimic the human movements observed during the demonstrations. The posture task uses the mapped joint angles as a joint reference and forces the optimizer to find solution close to it. This can be achieved in two ways:

1. By forcing the posture task to be solved along with all tasks, where in particular the posture is strictly adhered.

$$T_{comb} = T_{R_{wrist}} + T_{Posture} \quad (6)$$

2. Prioritizing the tasks over posture allows us to find unique solutions while staying close to the desired postures but also meeting the other task reference requirements. Here the tasks  $T_{R_{wrist}}$  and  $T_{L_{wrist}}$  are given higher priority than posture task.

$$T_{comb} = (T_{R_{wrist}} + T_{L_{wrist}}) / T_{Posture} \quad (7)$$

Prioritizing the tasks in such ways can strictly guarantee task references solutions with minimal error, for example if

legs are in support its references would take higher priority over other tasks. We follow these strategies to automatically build the task according to these rules, with some assumptions like if multiple common supports are present; we side with some links for common support. Our algorithm generates task formulations using these set of rules to define task for any contact sequence list.

### B. Cost definition for optimization

In this section we define the costs over which we can optimize the contact references to efficiently mimic the human demonstrated multi-contact task, while achieving the task in a realizable way.

1) *Stability Cost*: We define stability for contacts in a geometrical way as shown in Fig. 5 and it can be related strongly to friction cone definition. For example, if wrist is in contact with the surface we can ascertain its position is good enough to support the robot using this cost. We know that the shoulder is completely supported by wrists only if its projected position on the inertial frame coincides with the wrist position. If the contact can support the mass, we call this as support phase. We use this projection difference as a

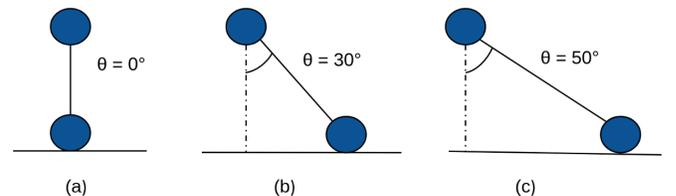


Fig. 5: The upper mass represents shoulder and lower mass is wrist. In (a) the shoulder is fully supported by the wrist, in (b) only a fraction of the shoulder mass is supported and in (c) the shoulder mass although supported can slide causing instability at the contact point if force is exerted.

measure of stability. And if a contact has non zero projection difference and without sliding if it can be made to support the shoulder mass, we call this reach-to-supportable phase. Here we select some body portions to define stability for the entire humanoid robot, we select shoulders for the upper body parts (in arms) and we select upper thigh region for lower body parts (in legs). Although these will not guarantee full stability of the robot unless we also consider support polygon to be used in conjunction to these definitions, to check if these

select parts are inside or on the support polygon region. We use projection of shoulder position over, when wrist or elbow are in contact with the obstacle and projection of upper thighs over, when knee or foot are in contact with the obstacle.

We define optimal value for the stability at any contact using its link length  $l$  up to the support mass i.e., either shoulders or upper thigh position as

$$l \sin(\theta) \quad (8)$$

where  $\theta$  is the angle measured as shown in the Fig. 5. The optimal range for the  $\theta$  to allow contact to support the mass without slipping is defined as  $35^\circ \geq \theta \geq 0^\circ$ . The absolute stability (full support) occurs at  $0^\circ$ .

2) *Collision Cost*: As we mention earlier, we approximate the robot links by simple geometrical objects like cylinders and spheres to enable quick collision checks. We use flexible collision library [14] for detecting these collisions, which outputs the closest points  $cp_1, cp_2$  on the objects and distance between them. We define position constraints on our task references to avoid collision in case of internal collisions by bounding with threshold  $\epsilon_{thres} < 0.05$ . The general constraint is defined as below

$$\|cp_1 - cp_2\| < \epsilon_{thres} \quad (9)$$

These collision constraints are enforced on the reference task formulations to find solutions in collision free space. We define only a minimal set of collision constraints using a group of selected link bodies for inter-body collisions in the humanoid robot as follows:

- (i) Identical link bodies in the left and right arms.
- (ii) Identical link bodies in the left and right legs.
- (iii) Elbows and knees on arms and legs.

The last group helps to avoid upper and lower body collisions sufficiently in our experiments. Collision avoidance with the obstacle in environments are obtained through surface bounds defined in the reference optimization section, which helps to keep limbs from penetrating the surface.

### C. Reference optimization

The contact reference positions must be optimized to find stable contact position references for the humanoid robot to climb on the obstacle. The contact position references are defined using configuration of the given robot link represented in the world frame. Then to impose the desired contact reference, we define as an equality constraint between two Cartesian frames,  $\mathbf{X}_i$  on the robot link  $i$  and the desired contact reference position  $\mathbf{X}_i^o$  on the obstacle, where  $\mathbf{X} \in \mathbf{R}^6$  represent the 6D contact vector containing both the position vector and orientation information as  $\mathbf{X} = [x, y, z, r, p, y]^T$ . Since we consider multi-contact motion-planning we allow contacts to take place over several robot links  $N$  indicated by  $i = \{1, \dots, N\}$ . We define our cost function  $L(\mathbf{X})$  for optimization of reference contact positions using constraints and costs defined before as

$$L(\mathbf{X})_i = w_1 \|\mathbf{X}_i - \mathbf{X}_i^o\|_b + w_2 \|\mathbf{X}_i - \mathbf{S}_i^o\|_b \quad (10)$$

$$\forall i = \{1, \dots, N\}$$

where  $\mathbf{X}_i^o$  is the initial reference contact position obtained from the playback analysis for the robot link  $i$ ,  $\mathbf{S}_i^o$  is the projected position of the shoulder or upper thigh robot link depending on the robot link under contact constraint. The first term imposes the reference contact position and the second term impose the support constraint which must be satisfied as equality constraints. Whereas the bounds  $b$  enforces the contact points to stay on obstacle surface and within the obstacle region while optimizing the references. We introduce weights for these terms to place more importance on each terms here using  $w_2 \geq 0$  and  $w_1 \geq 0$ .

To choose  $\mathbf{X}$  such as to minimize our cost function  $L$ , we consider the gradient descent algorithm that starts with an initial guess of reference positions  $\mathbf{X}_i^o$  obtained from playback of demonstrated data, and then repeatedly change this references  $\mathbf{X}$  to make our cost smaller, until we converge to a value that minimizes  $L$ . And perform the update on  $\mathbf{X}$  using the function defined as

$$\mathbf{X}_i := \mathbf{X}_i - \alpha \frac{\partial L(\mathbf{X})}{\partial \mathbf{X}_i} \quad (11)$$

Here when we start the search with an initial guess  $\mathbf{X}_i^o$ , we activate only the first term in  $L$  by setting  $w_2 = 0$ ,  $w_1 = 0.1$  to search only reachable references. Then we solve for IK solution with task formation as described before using the contact sequence associated with  $\mathbf{X}_i^o$ . We then check for solution error  $IK_{error}$  and perform the update function on  $\mathbf{X}$  until the error is minimized as  $IK_{error} < 0.01$ .

Using the joint solution we compute the  $\mathbf{S}_i^o$  support link position for the contact link  $i$  and plug into the cost function  $L$ . Next we activate only the second term in  $L$  by setting  $w_1 = 0$ ,  $w_2 = 0.1$  to search for stable contact references. We then update the  $\mathbf{X}$  and compute the IK solution until  $L$  is minimized sufficiently. We repeat this process of updating all the contact references in the current contact state. Then we repeat this for the next set of contact inputs in the sequence of movements.

### D. Search Optimization for intermediate postures

Once we have optimized the desired contact position references to obtain COMAN keyframes along with the contact equality constraints, inter-connectivity through common support link between adjacent contact references, stability costs optimized at the new set of contacts. But the set of postures (joint configurations) resulting from a mere trajectory interpolation of these adapted COMAN keyframes will not satisfy the static stability, resulting in unstable motions.

We propose a back-tracking search method which considers the contact position references in reverse order to search set of intermediate references in the process. This method reads the last contact reference as input and then apply contact references from its preceding state to begin the backward search. We relax the contact constraints at the input configuration only if those contacts do not appear at the final configuration. The stability costs are formulated as constraints and at these the input contact is bounded to stay in supportable mode. This allows the planner to find intermediate support location if needed.



Fig. 6: This is a sequence of screenshots taken while the humanoid robot COMAN is climbing on top of the wooden obstacle in the simulated world.

## V. EXPERIMENTAL RESULTS

The multi-contact motion planning algorithm is applied on the humanoid robot COMAN for the climbing task. In our experiments we use a Gazebo robot simulator for simulating COMAN in a virtual world environment. The virtual world includes a wooden obstacle of cuboid shape used for the climbing task and the simulated COMAN is placed in front of this wooden obstacle. The height of the obstacle in the world is  $h = 0.275$  m, whereas COMAN's knee height is slightly higher at 0.291 m. The mismatch in obstacle heights used in demonstration and simulation are optimized by our multi-contact motion planner with knowledge about obstacle height we use in the simulated world. The initial configuration of COMAN is obtained from simulator and is used in the search optimization process to eventually generate a set of statically stable solutions consisting of sequential robot postures to be executed on the robot. The stable solutions consists of joint angles positions at each robot posture; which are interpolated and applied for the humanoid robot COMAN. The robot uses PID based joint position controller to track the input joint angle references.

The climbing task was successfully attempted in simulations as shown in Fig. 6. The simulation and optimization were run on a i7 2.5 GHz processor with 16 GB RAM. The optimization of the references took less than 5 minutes with over 300 iterations for each posture generation. Search optimization for intermediate postures runs with over 50 iterations per posture generation and the entire process took 15 minutes, but it depends on the number of input keyframes to be converted into set of a statically stable poses. To draw comparisons in optimization time, for synthesizing various actions via CIO method needs 250 - 1000 optimization steps with a simulation time of 2 - 10 minutes [7]. CES Framework reports only search time for individual configuration poses as 3.4 s and 7.6 s for two simulation scenarios [5].

Since the solutions are computed off-line and then executed on the robot, we observe its influences on the task

execution like contact losses are encountered sometimes and it does not affect the overall stability of the robot nor impedes the execution performance due to multi-contact approach of the task. Our simulation of the robot climbing the obstacle experiment involves multitude of physical contacts with the obstacle. Although, due to static nature of the task execution on the robot the impacts are reduced at the physical contacts and it supports the robot.

During the simulation we record the robot positions w.r.t. the world, joint positions, velocities and torques. We compare these torques against max torque levels at the joints to assess the feasibility of running the motions. The max allowed joint torques on humanoid robot COMAN are in the range of 20 - 30 Nm at the peak performance level. We see that these max permissible torque limits are breached several times during the playback of movements as shown Fig. 7. These violations makes it impossible to execute these robot motions on the actual robot. We now take torque measurements when optimized solutions are used for the climbing task shown in Fig. 7. These torques are to ascertain the feasibility of running optimized solution on the actual robot COMAN. Here we only show a crucial portion of the torque measurements since the actual execution time is quite long because we produce a statically stable solution to the problem.

Here we see that the torque required are well within permissible level throughout the task execution time. The source code and supporting materials for the experiment can be obtained at <http://kormushev.com/goto/ROMAN-2017-Kanajar/>.

## VI. CONCLUSION

This paper has presented a novel way of climbing over large obstacles for a humanoid robot through multi-contact motion planning algorithm. Human demonstration was used to specify the goals for climbing the obstacle task. We proposed here a multi-contact motion planning algorithm which can directly work with the demonstration data.

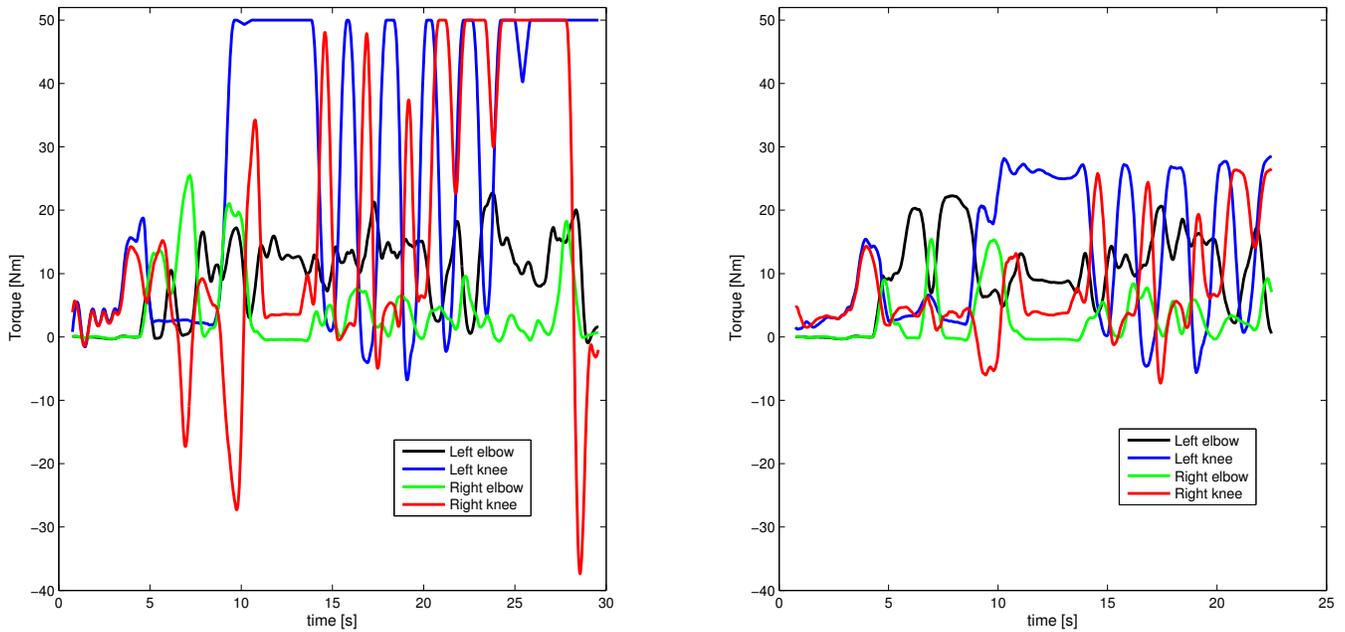


Fig. 7: Torques measured at the following joints: left knee (Lknee), right knee (Rknee), left elbow (Lelb) and right elbow (Relb). The left plot shows the torques measured during playback of movements in simulation. For comparison, right plot shows the torque measurements using the optimized solutions. We see significant reduction in the torque needs (below 30 Nm) with optimization, making it possible to execute the climb task in simulation.

Our algorithm builds on the whole-body control approach which formulated the task based on the observed contact states to generated keyframes for contact positions. Then, we obtain keyframes consistent with contact sequence through a reference optimization subject to stability costs, contact constraints and collision constraints.

Finally, through a search optimization of keyframe references with relaxed stability constraint, statically stable intermediate postures are generated for a humanoid robot. We applied our approach for the humanoid robot COMAN to climb on top of a table obstacle and successfully executed the simulation of climbing task. With this we have extended the state-of-the-art locomotion to climb over large obstacles via multi-contact motion planning algorithm. In the future, we plan to evaluate how our algorithm adapts to changes in the obstacle size and execute it on the real robot COMAN.

#### REFERENCES

- [1] P. Kryczka, P. Kormushev, N. G. Tsagarakis and D. G. Caldwell, "On-line regeneration of bipedal walking gait pattern optimizing footstep placement and timing," 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Hamburg, 2015.
- [2] Yajia Zhang et al., "Motion planning of ladder climbing for humanoid robots," 2013 IEEE Conference on Technologies for Practical Robot Applications (TePRA), Woburn, MA, 2013, pp. 1-6.
- [3] P. Kryczka, K. Hashimoto, A. Takahashi, P. Kormushev, N. Tsagarakis, and D. G. Caldwell, "Walking despite the passive compliance: Techniques for using conventional pattern generators to control intrinsically compliant humanoid robots," in Proc. Intl Conf. on Climbing and Walking Robots CLAWAR 2013, Sydney, Australia, July 2013.
- [4] A. Rocchi, E. M. Hoffman, D. G. Caldwell and N. G. Tsagarakis, "OpenSoT: A whole-body control library for the compliant humanoid robot COMAN," 2015 IEEE International Conference on Robotics and Automation (ICRA) 2015.
- [5] Shu-Yun Chung and O. Khatib, "Contact-consistent elastic strips for multi-contact locomotion planning of humanoid robots," 2015 IEEE International Conference on Robotics and Automation (ICRA), 2015.
- [6] P. Kormushev, D. N. Nenchev, S. Calinon and D. G. Caldwell, "Upper-body kinesthetic teaching of a free-standing humanoid robot," 2011 IEEE International Conference on Robotics and Automation, Shanghai, 2011, pp. 3970-3975.
- [7] Igor Mordatch, Emanuel Todorov, and Zoran Popovic. 2012. Discovery of complex behaviors through contact-invariant optimization. *ACM Trans. Graph.* 31, 4, Article 43 (July 2012).
- [8] A. J. Ijspeert, J. Nakanishi and S. Schaal, "Movement imitation with nonlinear dynamical systems in humanoid robots," Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No.02CH37292), Washington, DC, 2002, pp. 1398-1403.
- [9] J.J. Kuffner, K. Nishiwaki, S. Kagami, M. Inaba, and H. Inoue. "Motion Planning for Humanoid Robots," Proc. 11th Int. Symp. of Robotics Research (ISRR 2003), November, 2003.
- [10] M. Zucker, Y. Jun, B. Killen, T. G. Kim and P. Oh, "Continuous trajectory optimization for autonomous humanoid door opening," 2013 IEEE Conference on Technologies for Practical Robot Applications (TePRA), Woburn, MA, 2013, pp. 1-5.
- [11] S. Faraji, S. Pouya, C. G. Atkeson and A. J. Ijspeert, "Versatile and robust 3D walking with a simulated humanoid robot (Atlas): A model predictive control approach," 2014 IEEE International Conference on Robotics and Automation (ICRA), 2014.
- [12] J. Luo et al., "Robust ladder-climbing with a humanoid robot with application to the DARPA Robotics Challenge," 2014 IEEE International Conference on Robotics and Automation (ICRA), Hong Kong, 2014, pp. 2792-2798.
- [13] A. Gams, J. van den Kieboom, F. Dzeladini, A. Ude, and A. J. Ijspeert, "Real-time full body motion imitation on the COMAN humanoid robot," *Robotica*, vol. 33, no. 5, pp. 1049-1061, Jun. 2015.
- [14] J. Pan, S. Chitta and D. Manocha, "FCL: A general purpose library for collision and proximity queries," 2012 IEEE International Conference on Robotics and Automation, Saint Paul, MN, 2012, pp. 3859-3866.
- [15] Pavan Kanajar, Isura Ranatunga, Jartuwat Rajruangrabin, Dan O. Popa, and Fillia Makedon. 2011. Neptune: assistive robotic system for children with motor impairments. In Proceedings of the 4th International Conference on Pervasive Technologies Related to Assistive Environments (PETRA '11). ACM, New York, NY, USA, Article 59.