2014 14th IEEE-RAS International Conference on
Humanoid Robots (Humanoids)
November 18-20, 2014. Madrid, Spain

Whole-body Motion Planning with Centroidal Dynamics and Full Kinematics

Hongkai Dai, Andrés Valenzuela and Russ Tedrake

***Abstract*— To plan dynamic, whole-body motions for robots, one conventionally faces the choice between a complex, full-body dynamic model containing every link and actuator of the robot, or a highly simplified model of the robot as a point mass. In this paper we explore a powerful middle ground between these extremes. We exploit the fact that while the full dynamics of humanoid robots are complicated, their centroidal dynamics (the evolution of the angular momentum and the center of mass (COM) position) are much simpler. By treating the dynamics of the robot in centroidal form and directly optimizing the joint trajectories for the actuated degrees of freedom, we arrive at a method that enjoys simpler dynamics, while still having the expressiveness required to handle kinematic constraints such as collision avoidance or reaching to a target. We further require that the robot's COM and angular momentum as computed from the joint trajectories match those given by the centroidal dynamics. This ensures that the dynamics considered by our optimization are equivalent to the full dynamics of the robot, provided that the robot's actuators can supply sufficient torque. We demonstrate that this algorithm is capable of generating highly-dynamic motion plans with examples of a humanoid robot negotiating obstacle course elements and gait optimization for a quadrupedal robot. Additionally, we show that we can plan without pre-specifying the contact sequence by exploiting the complementarity conditions between contact forces and contact distance.**

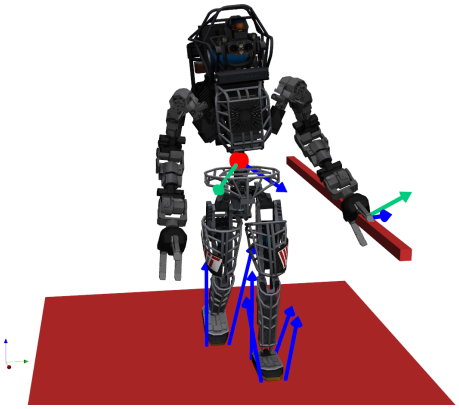


Fig. 1: Atlas robot subject to contact forces (blue arrows) on feet and hand, and contact torque (green arrows) on left hand when it grasps the hand rail. These contact wrenches and the gravitational force generate the aggregated force (blue arrow), and the aggregated torque (green arrow) at the Center of Mass (red sphere). The aggregated force and the torque equals to the rate of the centroidal linear and angular momentum.

I. INTRODUCTION

Humanoids are created with the dream of performing complex and dynamical motions like humans. Recent demonstrations, like the December 2013 Trials of the DARPA Robotics Challenge, have shown that while today's humanoids are capable of performing kinematically complex motions in uncontrolled environments, they are often restricted to the quasi-static regime. One reason for this is the difficulty in planning complex whole-body dynamic motions at interactive rates when the environment is not known a priori.

There are, broadly speaking, two approaches to dynamic motion planning for a humanoid robot. Some researchers use trajectory optimization with full-body dynamics. This approach can produce beautiful trajectories [16] [20] [4], but due to the complexity of the full-body dynamics, these optimizations can take an excessively long time to run, and may also suffer from local minima. Thus, this approach can become intractable for complex robots. On the other hand, there exists a large arsenal of planning algorithms that use a simple dynamics model like the linear inverted pendulum [12], [5]. With Zero Moment Point (ZMP) as the stability criteria, motion plans can be computed at interactive rates. However, there are some limitations to this approach. The over-simplified model regards the robot as a point-mass, and thus ignores all the kinematics constraints. Moreover, these models typically rely on the assumption that the center of mass (COM) height is constant (or on a constant slope), that the ground is flat, and the robot is only subject to unilateral ground contact forces on the feet. Thus, the formulation requires variations to apply to walking on uneven ground, and it is not applicable to more complicated motions like jumping and climbing. Additionally, the point-mass model suggests that the centroidal angular momentum is zero, which is not valid for motions requiring fast arm swinging. As a result, we need to resort to other stability criteria and models to design complex whole-body motion.

Maintaining the contact wrench sum (CWS) within the contact wrench cone (CWC) is proposed as a universal stability criteria for robot dynamics to replace the conventional ZMP [10]. It states that the aggregated wrench generated by the contact and the gravitational force, should be equal to the rate of linear and angular momentum of the robot. Unlike maintaining the ZMP within a support polygon, this criterion holds for arbitrary motions and contact profiles. However, like ZMP-based criteria, it eschews the complex, joint-level dynamics of a full-body model, and summarizes the robot's

H.Dai, A.Valenzuela and R.Tedrake are with Computer Science and Artificial Intelligence Lab, MIT daih,avalenzu,russt@csail.mit.edu

978-1-4799-7174-9/14/$31.00 ©2014 IEEE

dynamic state into a simple quantity, in this case its momenta. There has been a great success in controlling robots based on momenta [14] [13], including the resolved momentum control framework proposed by Kajita et. al. [11] [18]. In the graphics community, Ye constructs an abstract model with momenta being the state, and develops an optimal controller to simulate the character in the physics based animation [26]. The success of using momenta in motion control encourages us to apply the similar idea to whole-body motion planning.

A key observation is that the simple dynamics formulations, including ZMP and CWS formulations, can all be formulated as a nonlinear trajectory optimization problem. While this is not the standard formulation, it can still provide extremely efficient solutions. Similarly, while the inverse kinematics problem can be solved in closed-form for simple kinematic chains and relatively simple constraints, we have recently developed a fast and rich inverse kinematics engine based on nonlinear optimization[6]. Together, these observations highlight a continuum of algorithms which range from using simple dynamics to full dynamics, and/or simple kinematics to full kinematics. In this paper we explore a powerful middle ground, with simple dynamics and full kinematics. The hope is that we can rapidly find feasible trajectories for complex tasks. We demonstrate a variety of dynamic behaviors including a humanoid dynamically negotiating an obstacle course and dynamic gait optimization for a quadruped.

This paper is organized as follows. In Section II we describe the simple dynamics model and full kinematics model, as well as our formulation of the nonlinear programing problem. We also describe the variation of our formulation to incorporate planning unscheduled contact sequence. In Section III, we show our results on Atlas performing a variety of complex motions, and on LittleDog. We conclude our discussion in Section IV.

II. Approach

When a humanoid robot is in interacting with complex environments, kinematic reachability and collision avoidance can be just as important and constraining as the dynamic constraints like maintaining contact forces inside of a friction cone. To this end, we use a full kinematics model to enforce geometric contact conditions, and a dynamics model that encodes the relationship between the contact wrench (force/torque) on the robot and the robot's momenta.

A. Simple dynamics model

Robots with n joints have a total of $n + 6$ degrees of freedom (DOF), including joints and the 6 generalized coordinates for floating base. Even with full actuation of the joints, the six DOFs of the floating base are un-actuated. Those six degrees of freedom cannot be controlled directly; instead, the rates of those six DOFs are determined by the motion of the robot's links and the external wrenches on the robot, namely, the contact wrenches and the gravitational force. Those six DOFs can be represented using the robot's linear and angular momentum at the COM. A necessary condition for a physically tractable motion is that the rate of centroidal linear and angular momentum, computed from the robot's joint angles and velocities, equals the total wrench generated by the external contacts and the gravitational force:

$$m\ddot{\mathbf{r}} = \sum_j \mathbf{F_j} + m\mathbf{g} \tag{1a}$$

$$\dot{\mathbf{h}}(\mathbf{q}, \mathbf{v}) = \sum_j (\mathbf{c_j} - \mathbf{r}) \times \mathbf{F_j} + \tau_\mathbf{j} \tag{1b}$$

where m is the total mass of the robot, $\mathbf{r} \in \mathbb{R}^3$ is the COM position, $\mathbf{F_j} \in \mathbb{R}^3$ is the contact force at j^{th} contact point, and $\mathbf{g} \in \mathbb{R}^3$ is the gravitational acceleration. Eq.(1a) is Newton's second law enforcing that the rate of linear momentum of the robot equals the total external forces. $\mathbf{h}(\mathbf{q}, \mathbf{v}) \in \mathbb{R}^3$ is the centroidal angular momentum computed from the robot posture $\mathbf{q} \in \mathbb{R}^{n+6}$ and posture velocity $\mathbf{v} \in \mathbb{R}^{n+6}$. $\mathbf{c_j} \in \mathbb{R}^3$ is the position of the j^{th} contact point. $\tau_\mathbf{j} \in \mathbb{R}^3$ is the contact torque at the j^{th} contact point. Eq.(1b) enforces that the rate of centroidal angular momentum equals the torque generated by the contact wrenches at the COM. The centroidal angular momentum can be computed using the method described in [19]

$$\mathbf{h}(\mathbf{q}, \mathbf{v}) = \mathbf{A}(\mathbf{q})\mathbf{v} \tag{2}$$

where $\mathbf{A}(\mathbf{q}) \in \mathbb{R}^{3\times(n+6)}$ is the centroidal angular momentum matrix.

Assuming sufficient control authority (sufficient DOFs away from singularity and strong actuators), the six equations (1a-1b) are also sufficient conditions for planning dynamically feasible motions. Many robots, including most humanoids, have actuators for every internal joint; in that case, for any desired joint acceleration there is always a joint torque to achieve such motion. As a result, if we ignore force/torque limits of the actuators, then we can ignore the internal, joint-level dynamics of the robot. Thus the six equations (1a-1b), which relate the external wrenches to the overall momentum of the robot, are necessary and sufficient to describe the dynamics of the robot. This dynamics model is much simpler than the full-body model, with fewer constraints ($n+6$ to 6), and fewer variables, as the joint torques can be computed subsequently using inverse dynamics.

B. Full kinematics model

In order to accommodate the geometric constraints imposed by interaction between the robot and its environment, we plan using a full model of the robot's kinematics. This allows us to specify a rich variety of constraints on the robot's motion. These range from simple constraints on the position/orientation of the robot's end-effectors, to gaze constraints between links of the robot ("the head must look at the right hand"), to constraints across multiple points in time ("the right foot must remain stationary between times t_1 and t_2"), to collision avoidance constraints. Several of these constraint types are demonstrated in Figure 2.

Since we wish to resolve kinematic constraints at multiple instants in time, where the joint configuration at each

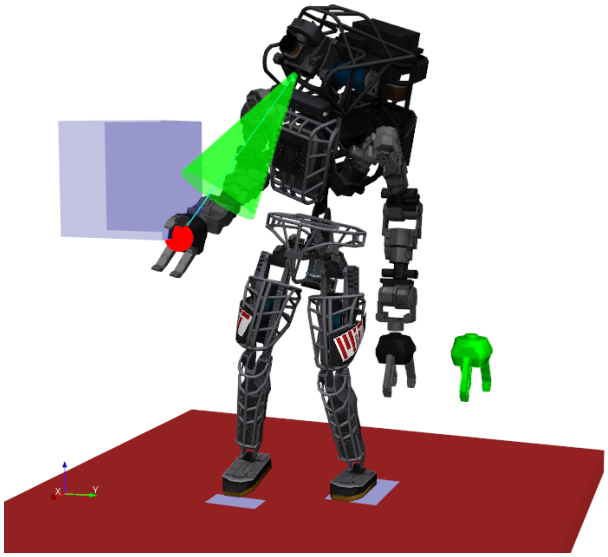


Fig. 2: Solving inverse kinematics problem with different types of kinematic constraints. The left foot and the right foot toes are constrained to lie within the shaded regions. A point (red sphere) on the right hand is constrained to be within the shaded bounding box. The head camera gazes at the point (red sphere) on the right hand, such that the point is within a cone originated from the camera, with 15° being the half angle of the cone. The left hand orientation is constrained to be the same as the green hand drawn by side.

instant is in some way related to that at adjacent instants, analytical or Jacobian transpose based approaches to inverse kinematics that consider the solution at a single instant in time [2] are not sufficient. However, the formulation of inverse kinematics as a nonlinear optimization extends quite naturally to this situation. The kinematic model requires $n+6$ decision variables for each instant considered which leads to large optimization problems. Fortunately, satisfaction of each kinematic constraints depends only on the state of the robot during a particular interval in time. Therefore a motion planning problem with kinematic constraints will tend to be sparse, in that each constraint will depend on only a small fraction of the decision variables.

C. Collision model

One of the more complex kinematic constraints on the motion of a humanoid robot is that the motion be collision free. Our collision model consists of n_{elem} convex collision geometries each of which is attached to the world or one of the robot's links at a known transform. Let $d_{ij}(\mathbf{q})$ denote the minimum distance between the i-th and j-th collision elements for the configuration vector $\mathbf{q}$. The distance between two collision elements can be efficiently computed for many classes of convex geometries with the Gilbert-Johnson-Keerthi algorithm (GJK) [7]. In this work we use the implementation of GJK in the Bullet Physics SDK [3]. Let $d_{\min}$ denote the minimum allowable distance between any pair of collision geometries, and let $\bar{d}_{ij}(\mathbf{q})$ denote their difference. Thus we wish to enforce that

$$\bar{d}_{ij} = d_{ij}(\mathbf{q}) - d_{\min} \geq 0,\ \forall (i,j) \in P \tag{3}$$

where $P \subset \{1,\ldots,n_{elem}\} \times \{1,\ldots,n_{elem}\}$ is the set of index pairs that correspond to pairs of potentially colliding geometries. The number of potential collision pairs grows with the square of the number of collision geometries. In order to decrease the number of collision avoidance constraints that must be added to the trajectory optimization, we can combine all of the collision pairs using a hinge-loss-like function. Schulman et. al. use a true hinge-loss function for a similar purpose [22]. Here we use a smooth function $\gamma(x)$ that is identically zero for all positive x, greater than zero for all negative x, and approaches $-x$ asymptotically as x goes to negative infinity:

$$\gamma(x) = \begin{cases} 0 & x \geq 0 \\ -xe^{\frac{1}{x}} & x < 0 \end{cases} \tag{4}$$

This function has the advantage of being infinitely differentiable for all x. The overall collision constraint is given by

$$\mathbf{\Gamma}(\mathbf{q}) = \sum_{(i,j)\in P_{ij}} \gamma\left(c\bar{d}_{ij}(\mathbf{q})\right) = 0, \tag{5}$$

where c is a positive scaling factor. In the examples shown here c was taken to be $\frac{1}{d_{\min}}$. Since each term of the sum in (5) is non-negative, (5) holds if and only if all terms of that sum are zero, which in turn implies that (3) holds.

D. Trajectory Optimization

To compute a feasible motion plan, we transcribe the differential equations of the simple dynamics (1a-1b) to algebraic equations and solve them through nonlinear optimization. This technique is widely used in trajectory optimization [1], [9], [25]. Here we sample all time-varying quantities at N knot points, with the time durations, h, between knot points being flexible. The optimization problem we formulate contains as decision variables the robot state $\mathbf{q}, \mathbf{v}$, COM position $\mathbf{r}$, velocity $\dot{\mathbf{r}}$, acceleration $\ddot{\mathbf{r}}$, contact positions $\mathbf{c}$, contact forces $\mathbf{F}$, contact torques τ, centroidal angular momentum $\mathbf{h}$ and its rate $\dot{\mathbf{h}}$ at each knot as well as the time duration between each pair of adjacent knot points, dt. We use the following objective for our optimization:

$$\min_{\substack{\mathbf{q}[k],\mathbf{v}[k],dt[k] \\ \mathbf{r}[k],\dot{\mathbf{r}}[k],\ddot{\mathbf{r}}[k] \\ \mathbf{c}_j[k],\mathbf{F}_j[k],\tau_j[k] \\ \mathbf{h}[k],\dot{\mathbf{h}}[k]}} \sum_{k=1}^{N} \Big(|\mathbf{q}[k] - \mathbf{q_{nom}}[k]|^2_{Q_q} + |\mathbf{v}[k]|^2_{Q_v} + |\ddot{\mathbf{r}}[k]|^2 + \sum_j (c_1 |\mathbf{F}_j[k]|^2 + c_2 |\tau_j[k]|^2) \Big) dt[k], \tag{6}$$

where $|\mathbf{x}|^2_Q$ is the abbreviation for the quadratic cost $\mathbf{x}'Q\mathbf{x}, Q \succeq 0$. The square bracket $[k]$ means the sampled value at the k^{th} knot point. $\mathbf{q_{nom}}$ represents a nominal posture. The first three quantities in the cost are penalization of the weighted sum on the posture error, joint velocities and COM acceleration. We also penalize the weighted L_2

norm of the contact wrench with the cost $\sum_j (c_1 \left|\mathbf{F}_j[k]\right|^2 + c_2|\tau_j[k]|^2)$,where c_1, c_2 are positive scaling factors. This L_2 norm cost has two effects: it can prevent a large contact wrenches which could damage the robot, and it also encourages the contact wrenches to be more evenly distributed.

The constraints for the optimization problem include the dynamical constraints (1a-1b, 2), at each knot point:

$$m\ddot{\mathbf{r}}[k] = \sum_j \mathbf{F_j}[k] + m\mathbf{g} \tag{7a}$$

$$\dot{\mathbf{h}}[k] = \sum_j (\mathbf{c_j}[k] - \mathbf{r}[k]) \times \mathbf{F_j}[k] + \tau_\mathbf{j}[k] \tag{7b}$$

$$\mathbf{h}[k] = \mathbf{A}(\mathbf{q}[k])\mathbf{v}[k]. \tag{7c}$$

For simplicity, we use Euler integration to approximate the time derivative function for posture $\mathbf{q}$ and centroidal angular momentum $\mathbf{h}$. For numerical stability, backward-Euler is adopted in our formulation. The time integration constraints are

$$\mathbf{q}[k] - \mathbf{q}[k-1] = \mathbf{v}[k]dt[k] \tag{7d}$$

$$\mathbf{h}[k] - \mathbf{h}[k-1] = \dot{\mathbf{h}}[k]dt[k]. \tag{7e}$$

The COM position is approximated using a piecewise quadratic polynomial, and its time integration constraints are

$$\mathbf{r}[k] - \mathbf{r}[k-1] = \frac{\dot{\mathbf{r}}[k] + \dot{\mathbf{r}}[k-1]}{2} dt[k] \tag{7f}$$

and

$$\dot{\mathbf{r}}[k] - \dot{\mathbf{r}}[k-1] = \ddot{\mathbf{r}}[k]dt[k]. \tag{7g}$$

We also have the kinematic constraints that compute the COM and contact positions from robot posture

$$\mathbf{r}[k] = \mathrm{com}(\mathbf{q}[k]) \tag{7h}$$

$$\mathbf{c}_j[k] = p_j(\mathbf{q}[k]) \tag{7i}$$

$$\mathbf{c}_j[k] \in \mathcal{S}_j[k] \tag{7j}$$

where $\mathrm{com}\,(\mathbf{q})$ is a function that computes the COM location given the robot posture $\mathbf{q}$, $p_j(\mathbf{q})$ is the forward kinematics function to compute the position of the j^{th} contact point for configuration $\mathbf{q}$, and the set $\mathcal{S}_j$ is the desired contact region (ground, stepping stones, hand rail, etc).

We further include joint-limit constraints on posture $\mathbf{q}$, constraints on joint velocities, and constraints on the contact wrench.

$$\mathbf{q} \in \mathcal{Q}, \mathbf{v} \in \mathcal{V}, \begin{bmatrix} \mathbf{F}_j \\ \tau_j \end{bmatrix} \in \mathbf{w}_j \tag{7k}$$

where $\mathcal{Q}$ is the admissible set of the posture, $\mathcal{V}$ is the admissible set of velocities. The set $\mathbf{w}_j$ represent the constraints on the contact wrench, for example, friction cone constraints, or bounded magnitude on the contact wrench. We can also incorporate additional kinematics constraints, such as those described in subsections II-B and II-C.

This nonlinear optimization problem has very sparse gradients, since most constraints only depend on variables at a single knot point or two adjacent knot points. Such nonlinear programs with sparse gradients can be solved efficiently using powerful nonlinear solvers like SNOPT [8], which we use for the examples in Section III.

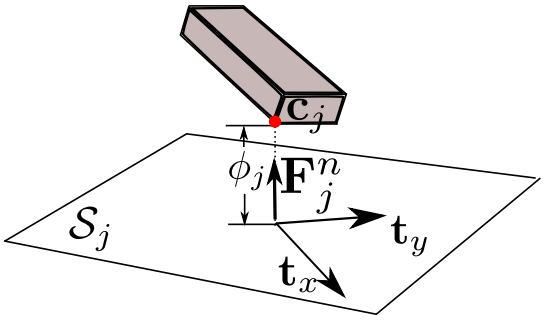


Fig. 3: Illustration of the contact point $\mathbf{c}_j$, its distance ϕ_j to the contact surface $\mathcal{S}_j$, and the local coordinate frame on the tangential surface, with unit vector $\mathbf{t}_x, \mathbf{t}_y$. The complementarity condition holds between contact distance ϕ_j and the normal contact force $\mathbf{F}_j^n$.

E. Unscheduled Contact Sequence

When designing robot motion with contact, the traditional approach is to pre-specify a contact mode sequence, for example, heel touch $\rightarrow$ toe touch $\rightarrow$ heel off $\rightarrow$ toe off, and then use optimization to find a trajectory for this fixed mode sequence. However, the number of possible contact modes grows exponentially with the number of contact points, and the number of possible mode sequences for a given set of contact modes grows exponentially with the number of knot points. This makes it hard to choose a mode sequence prior to optimization in many cases. Optimization methods that can search over all possible mode sequences at once are therefore very useful. By exploiting the complementarity condition between the contact force and the distance to contact, Posa formulates a direct trajectory optimization problem that does not require a pre-specified contact sequence [21]. In this paper we apply the same idea to our motion planning algorithm.

The complementarity constraints on the contact wrench and contact distance are

$$\mathbf{F}_j^n[k]\phi_j(\mathbf{q}[k]) = 0 \tag{8a}$$

$$|\tau_j[k]|^2\phi_j(\mathbf{q}[k]) = 0 \tag{8b}$$

$$\mathbf{F}_j^n[k] \geq 0, \phi_j(\mathbf{q}[k]) \geq 0 \tag{8c}$$

where $\mathbf{F}_j^n \in \mathbb{R}$ is the magnitude of the normal contact force at the j^{th} contact point, $\phi_j(\mathbf{q})$ is the distance of the j^{th} contact point $\mathbf{c}_j$ to the contact surface $\mathcal{S}_j$ (ground, hand rail, etc.). The scalar product being zero in equations (8a-8b) means that either the body is not in contact with the environment, and thus the contact wrench is zero, or the distance between the body and the contact surface is zero and therefore the contact wrench may be non-zero. The relationship between the contact force and distance is illustrated in Fig. 3.

For simplicity of analysis, we also impose the constraint that bodies in contact with the environment do not slide. We

can use the following complementarity constraint when only contact forces exist at a certain contact point, and the contact torque is always zero:

$$\mathbf{F}_j^n[k]\left((\mathbf{c}_j[k]-\mathbf{c}_j[k-1])'\mathbf{t}_x\right)=0 \tag{9a}$$

$$\mathbf{F}_j^n[k]\left((\mathbf{c}_j[k]-\mathbf{c}_j[k-1])'\mathbf{t}_y\right)=0, \tag{9b}$$

where $\mathbf{t}_x, \mathbf{t}_y \in \mathbb{R}^3$ are mutually orthogonal unit vectors on the tangent surface of the contact (Fig. 3). The complementarity constraints (9a-9b) state that if the normal contact force exists, i.e. , the body is in contact, then the tangential displacement of the body between the two adjacent knots must to be zero; if the body moves in the tangential directions, then the normal force has to be zero, i.e. , the body is not in contact.

When we want the solver to search over all possible contact sequences, we add the complementarity constraints (8a-8b, 9a-9b) into the optimization problem in the previous subsection. Additionally, the objective function (6) must be changed: it can no longer include the L_2 norm on the contact wrench. This is due to the fact that penalizing the L_2 norm would encourage the forces to be distributed more evenly among the contact points, biasing the optimization towards solutions in which all potential contact points are active at the same knot point. Therefore, the objective function is reduced to

$$\min \sum_{k=1}^{N}\left(|\mathbf{q}[k]-\mathbf{q_{nom}}[k]|^2_{Q_q}+|\mathbf{v}[k]|^2_{Q_v}+|\ddot{\mathbf{r}}[k]|^2\right)dt[k]. \tag{10}$$

Alternatively, we could also add the L_1 norm of the wrench in the cost, so as to encourage sparse solutions, meaning the contact points tend to be active at different moments.

With the extra complementarity constraints and the revised objective function, the solver can search over all possible contact sequences.

III. Results

For our numerical experiments, we primarily use a dynamic model of the Atlas humanoid robot [17], built by Boston Dynamics, Inc for use in the DARPA Robotics Challenge.

A. Jumping

Our first example is to command the robot to jump off the ground, as illustrated in Fig. 4. We assign contact points to the four corners of each foot. The contact sequence is fixed as 1) heels take off 2) toes take off 3) toes touch ground 4) heels touch ground. After computing the planned trajectory, we simulate the jumping motion in Drake [24], stabilized by the feedback controller developed by Kuindersma, Permenter, and Tedrake [15]. The simulated motion is shown in the supplemental video.

We also compute the motion of Atlas jumping off of a box, with box height 29 centimeters. The contact sequence is the same as that used in the flat ground case. In this example we add the kinematic constraint that the feet must avoid collision with the box during the flight phase. Some snapshots of the resulting motion are shown in Figure 5.

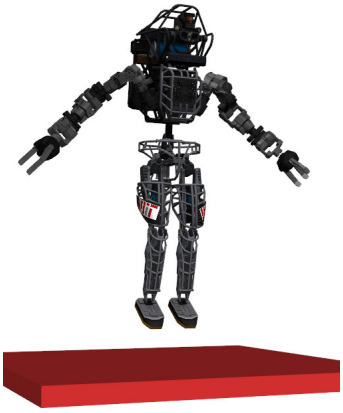


Fig. 4: Atlas jumps off the ground

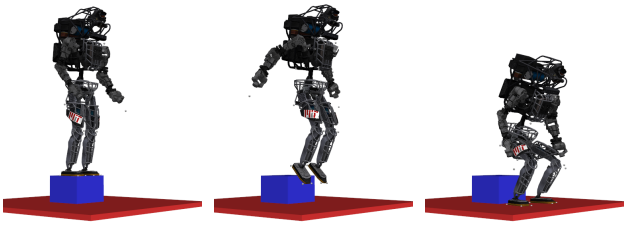


Fig. 5: Snapshots of Atlas jumping off from a box

B. Running

Our next example consists of generating a periodic running gait for Atlas. To do this, we plan a trajectory consisting of a single half-stride starting at the apex of a flight phase. The mode sequence is specified to be flight, left-stance, left-toe-stance, flight. We constrain the initial and final states relative to each other such that all quantities are mirrored about the robot's sagittal plane. This yields a half-stride that can be mirrored and concatenated to yield a full stride. In addition to these periodicity constraints and the kinematic constraints enforcing the mode sequence, we specify a stride length and speed ($2\,\mathrm{m}$ and $2\,\mathrm{m/s}$ respectively), require a minimum distance between collision geometries of $3\,\mathrm{cm}$, and constrain the robot's head cameras be point within 15° of the robot's direction of travel. The remainder of the robot's trajectory is unconstrained. Snapshots from the stance phase of the resulting motion are shown in Figure 6. Figure 7 shows the resulting COM trajectory and vertical ground reaction force profile. We also simulated this running motion with the controller from [15]; the simulated motion is shown in the supplemental video.

Starting from this periodic gait we can employ our collision avoidance constraints to generate running motions in obstructed environments. Figure 8 shows snapshots from a trajectory which takes Atlas through the obstructed door presented by Schulman et. al. [22] at $2\,\mathrm{m/s}$. The end-points

(a) Touch-down (b) Mid-stance (c) Toe-off

Fig. 6: Snapshots of Atlas during stance

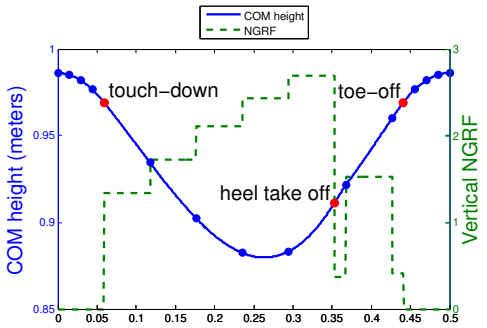


Fig. 7: COM height and normalized vertical ground reaction force $\left(\frac{F}{mg}\right)$ during running. The markers indicate the height of the COM at the knot points

of this trajectory are the same as those of the original, un-obstructed half-stride, allowing an immediate return to periodic motion after traversing the door.

C. Monkey bars

In this test case, the robot jumps up from a platform to grasp a bar, then makes several swings, and finally to lands on another platform, as illustrated in Fig. 9. The spacing or the height of the bars changes between each swing.

As grasp-planning is outside of the scope of this work, we do not use a full hand model for this example. Rather, we assume that the kinematic constraint for grasping consists of the center of the hand coinciding with the axis of the bar, and a predefined axis on the hand being coaxial with the bar

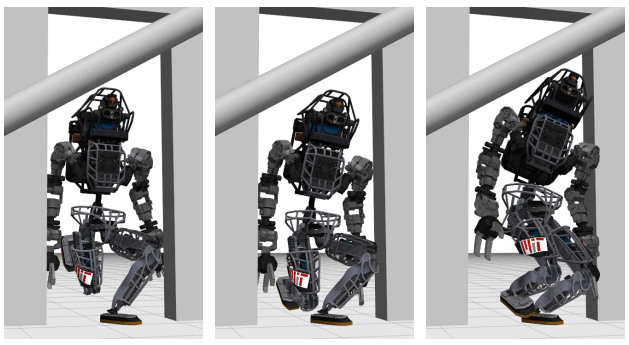


Fig. 8: Atlas goes through an obstructed door

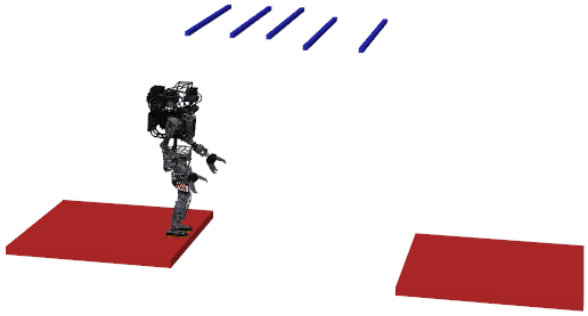
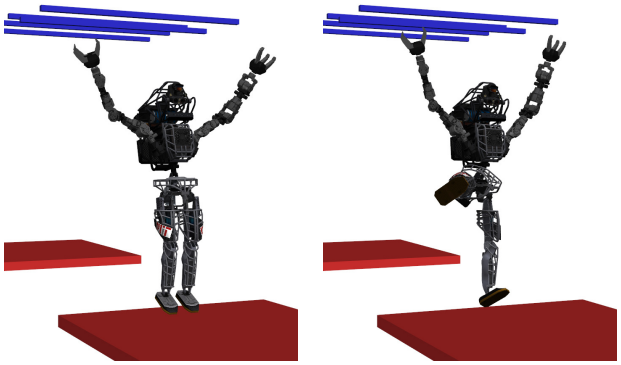


Fig. 9: A snapshot of the monkey bars

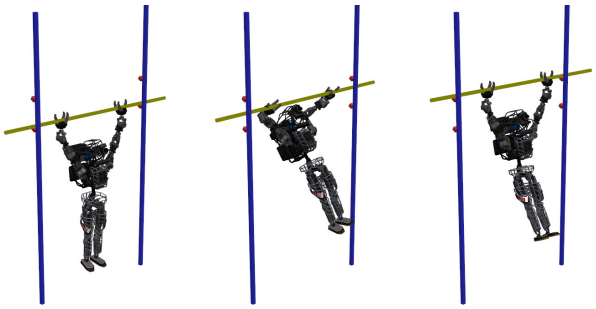


(a) Landing with a fixed mode sequence. The left and right toes touch the ground first (b) Landing without a pre-specified mode sequence. The left heel touches the ground first

Fig. 10: Comparison of landing posture without pre-specified contact sequence

to within $15°$. In order not to avoid damaging the hand with excessive contact wrenches, we suppose the magnitude of the contact force on each hand is bounded, and the contact torque is within a bounding box centered at the origin. With these assumptions, we compute the motion in the monkey bar tasks, which is shown in the attached video.

Unspecified contact sequence for landing: While it is simple to determine the contact sequence for swinging across the bars (recall that the *timing* of the modes is still determined by the optimization), it is non-intuitive to determine the contact sequence while the robot swings itself forward from the last bar and lands on the platform. We compare the fixed mode version to the mode-free version with complementarity constraints for the landing phase. We first fix the mode to be 1) left/right foot toes touch ground 2) left/right foot heels touch ground. With the solution to the fixed mode as the seed, we re-compute the landing motion using the complementarity formulation and free mode sequence. The results are compared in Fig. 10. The contact sequence and the robot motion change significantly after optimization with the complementarity constraints. This indicates that our algorithm is indeed searching over different mode sequences. To our eyes the motion obtained from complementarity formulation looks more natural.



(a) t=0 (b) t=0.8 (c) t=1.41

Fig. 11: Snapshots of climbing salmon ladder

(a) Take off (b) Flight (c) Landing

Fig. 12: Snapshots of the flight phase during LittleDog running

D. Salmon ladder

Our final humanoid test-case is the obstacle course element called the "salmon ladder" (popularized by the 'Sasuke' and now 'American Ninja Warrior' competitions), shown in Fig. 11 and in the attached video. Despite the restrictive kinematics of Atlas' arms, the robot is able to launch itself from one level to the next by bending its elbows and swinging its legs in much the same way as human athletes.

E. LittleDog

Our algorithm is not restricted to humanoids, it can be applied to a large class of robots. To show the generality of our approach, we compute a running trot trajectory for LittleDog, a quadruped robot [23]. Snapshots of its flight phase are shown in Fig. 12. We also designed several galloping, bounding, and walking gaits for LittleDog, with very little variation on the code. Those gaits are also included in the attached video.

IV. Discussion and Conclusion

A major concern in solving any NLP is running time. The current un-optimized MATLAB code employed in the examples above usually generates those motions within several minutes, but can take as long as several hours in some poorly initialized cases. We will implement the approaches described here in C++ next; our experience is that the C++ code is about two orders of magnitude faster than the MATLAB code for NLPs of a comparable size to the ones presented here. For the DARPA Robotics Challenge Trials, we developed a C++ implementation of an NLP-based kinematics-only planner that treated the same kinematic constraints as those employed here. It generates motion trajectories interactively with the user, in less than 0.2 seconds.

We also wrote a planner that uses the full-body dynamics model for comparison. For the rich constraints used in these examples, we struggled to have that solver return even a feasible solution, even after hours or days of computation. We believe this failure is due to the large size of the problem, and the presence of many local minima; however more work is required to show that that is the case.

In this paper we combine a simple dynamics model and a full kinematics model to generate robots' whole-body motions. We believe this combination encodes the physical laws necessary to efficiently generate dynamically and kinematically feasible motions, giving it an advantage over both the full-body model and over-simplified models. To handle the case when the contact mode sequence cannot be obtained prior to the trajectory optimization, we present a formulation with complementarity constraints, so that the solver can search over all possible combinations of contact sequences simultaneously. In addition to demonstrating the effectiveness of our algorithm on several examples with a humanoid robot, we further show that our algorithm is not restricted to humanoids, but is applicable to a much larger class of robots.

V. Acknowledgement

We gratefully acknowledge the support of the Defense Advanced Research Projects Agency via Air Force Research Laboratory award FA8750-12-1-0321, and the support of David S.Y Wong and Harold Wong Fellowship. We are also grateful for the help we received from members of the Robot Locomotion Group and the MIT DARPA Robotics Challenge Team. Special thanks go to Pat Marion for his great help in visualizing the robot motions, Phil Cherner for making the Robotiq hand model, and Amara Mesnik for editing the video. We also want to express our deep gratitude to Prof. Seth Teller for his guidance of our team and his passion and vision for robotics.

References

[1] John T. Betts. *Practical Methods for Optimal Control Using Nonlinear Programming*. SIAM Advances in Design and Control. Society for Industrial and Applied Mathematics, 2001.

[2] Samuel R. Buss. Introduction to inverse kinematics with jacobian transpose, pseudoinverse and damped least squares methods. April 2004.

[3] Erwin Coumans et al. Bullet physics library. *bulletphysics. org*, 4(6), 2014.

[4] Moritz Diehl, Hans Georg Bock, Holger Diedam, and P-B Wieber. Fast direct multiple shooting algorithms for optimal robot control. In *Fast motions in biomechanics and robotics*, pages 65–93. Springer, 2006.

[5] Dimitar Dimitrov, Alexander Sherikov, and Pierre-Brice Wieber. A sparse model predictive control formulation for walking motion generation. In *Proceedings of the 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2011.

[6] Maurice Fallon, Scott Kuindersma, Sisir Karumanchi, Matthew Antone, Toby Schneider, Hongkai Dai, Claudia Pérez D'Arpino, Robin Deits, Matt DiCicco, Dehann Fourie, Twan Koolen, Pat Marion, Michael Posa, Andrés Valenzuela, Kuan-Ting Yu, Julie Shah, Karl Iagnemma, Russ Tedrake, and Seth Teller. An architecture for online affordance-based perception and whole-body planning. *Journal of Field Robotics*, September 2014.

[7] E.G. Gilbert, D.W. Johnson, and S.S. Keerthi. A fast procedure for computing the distance between complex objects in three-dimensional space. *Robotics and Automation, IEEE Journal of*, 4(2):193–203, Apr 1988.

[8] Philip E. Gill, Walter Murray, and Michael A. Saunders. SNOPT: An SQP algorithm for large-scale constrained optimization. *SIAM Review*, 47(1):99–131, 2005.

[9] C. R. Hargraves and S. W. Paris. Direct trajectory optimization using nonlinear programming and collocation. *J Guidance*, 10(4):338–342, July-August 1987.

[10] H Hirukawa, S Hattori, K Harada, S Kajita, K Kaneko, F Kanehiro, K Fujiwara, and M Morisawa. A universal stability criterion of the foot contact of legged robots - Adios ZMP. *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pages 1976–1983, May 2006.

[11] Kajita, S. Kanehiro, F. Kaneko, K. Fujiwara, K. Harada, K. Yokoi, K. Hirukawa, and H. Resolved momentum control: humanoid motion planning based on the linear and angular momentum. *Intelligent Robots and Systems (IROS), Proceedings*, 2003.

[12] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiware, K. Harada, K. Yokoi, and H. Hirukawa. Biped walking pattern generation by using preview control of zero-moment point. In *ICRA IEEE International Conference on Robotics and Automation*, pages 1620–1626. IEEE, Sep 2003.

[13] Twan Koolen, Jesper Smith, Gray Thomas, Sylvain Bertrand, John Carff, Nathan Mertins, Douglas Stephen, Peter Abeles, Johannes Englsberger, Stephen Mccrory, and Jeff Egmond. Summary of team ihmc s virtual robotics challenge entry. In *Proceedings of the IEEE-RAS International Conference on Humanoid Robots.*, Atlanta, GA, oct 2013. IEEE, IEEE.

[14] K. Koyanagi, H. Hirukawa, S. Hattori, M. Morisawa, S. Nakaoka, K. Harada, and S. Kajita. A pattern generator of humanoid robots walking on a rough terrain using a handrail. In *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, pages 2617–2622, Sept 2008.

[15] Scott Kuindersma, Frank Permenter, and Russ Tedrake. An efficiently solvable quadratic program for stabilizing dynamic locomotion. In *Proceedings of the International Conference on Robotics and Automation*, Hong Kong, China, May 2014. IEEE.

[16] Katja D. Mombaur. Using optimization to create self-stable human-like running. *Robotica*, 27(3):321–330, 2009.

[17] Gabe Nelson, Aaron Saunders, Neil Neville, Ben Swilling, Joe Bondaryk, Devin Billings, Chris Lee, Robert Playter, and Marc Raibert. Petman: A humanoid robot for testing chemical protective clothing. *Journal of the Robotics Society of Japan*, 30(4):372–377, 2012.

[18] E.S. Neo, K. Yokoi, S. Kajita, and K. Tanie. Whole-body motion generation integrating operator's intention and robot's autonomy in controlling humanoid robots. *Robotics, IEEE Transactions on*, 23(4):763–775, Aug 2007.

[19] David E. Orin, Ambarish Goswami, and Sung-Hee Lee. Centroidal dynamics of a humanoid robot. *Autonomous Robots*, (September 2012):1–16, jun 2013.

[20] Michael Posa, Cecilia Cantu, and Russ Tedrake. A direct method for trajectory optimization of rigid bodies through contact. *International Journal of Robotics Research*, 33(1):69–81, January 2014.

[21] Michael Posa and Russ Tedrake. Direct trajectory optimization of rigid body dynamical systems through contact. In *Proceedings of the Workshop on the Algorithmic Foundations of Robotics*, page 16, Cambridge, MA, June 2012.

[22] John Schulman, Jonathan Ho, Alex Lee, Ibrahim Awwal, Henry Bradlow, and Pieter Abbeel. Finding locally optimal, collision-free trajectories with sequential convex optimization. In *Robotics: Science and Systems*, volume 9, pages 1–10. Citeseer, 2013.

[23] Alexander Shkolnik, Michael Levashov, Ian R. Manchester, and Russ Tedrake. Bounding on rough terrain with the littledog robot. *The International Journal of Robotics Research (IJRR)*, 30(2):192–215, Feb 2011.

[24] Russ Tedrake. Drake: A planning, control, and analysis toolbox for nonlinear dynamical systems. http://drake.mit.edu, 2014.

[25] O. von Stryk and R. Bulirsch. Direct and indirect methods for trajectory optimization. *Annals of Operations Research*, 37:357–373, 1992.

[26] Yuting Ye and C. Karen Liu. Optimal feedback control for character animation using an abstract model. *ACM Trans. Graph.*, 29(4):74:1–74:9, July 2010.