

Whole-body Motion Planning with Simple 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 present an approach to generate whole-body motions using a simple dynamics model, which enforces that the linear and angular momentum of the robot be consistent with the external wrenches on the robot, and a full-body kinematics model that enforces rich geometric constraints, such as end-effector positioning or collision avoidance. We obtain a trajectory for the robot and profiles of contact wrenches by solving a nonlinear optimization problem (NLP). We further demonstrate that we can plan without pre-specifying the contact sequence by exploiting the complementarity conditions between contact forces and contact distance. 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.

I. INTRODUCTION

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 [9], 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, making it 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 [6]. 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.

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

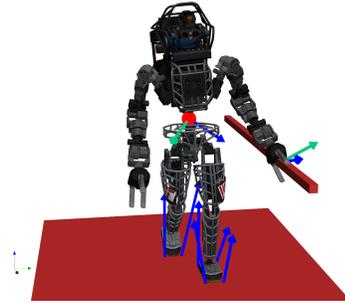


Fig. 1: Atlas robot subject to contact forces (blue arrows) on feet and hand, and contact torque (green arrows) on left hand when it grasps the hand rail. These contact wrenches and the gravitational force generate the aggregated force (blue arrow), and the aggregated torque (green arrow) at the Center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

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 [4]. 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 dynamic state into a simple quantity, in this case its momenta. There has been a great success in controlling robots based on momenta [7], including the resolved momentum control framework proposed by Kajita et. al. [5]. 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 very fast and rich inverse kinematics engine based on nonlinear optimization. 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 very powerful middle ground, with simple dynamics but rich kinematics. The hope is that we can very rapidly find feasible trajectories for very complex tasks. The fear is that we might sacrifice some richness in the dynamic motions, however

we demonstrate a variety of very rich dynamic behaviors including a humanoid dynamically negotiating an obstacle course and dynamic gait optimization for a quadruped.

II. APPROACH

A. Simple dynamics model

Robots with n joints have a total of $n + 6$ degrees of freedom (DOF), including joints and the floating base. Even with full actuation of the joints, the six DOFs of the floating base are un-actuated. Those six DOFs can be represented using the robot’s linear and angular momentum at the COM. A necessary condition for a physical 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:

$$m\ddot{\mathbf{r}} = \sum_j \mathbf{F}_j + m\mathbf{g} \quad (1a)$$

$$\dot{\mathbf{H}}(\mathbf{q}, \mathbf{v}) = \sum_j (\mathbf{c}_j - \mathbf{r}) \times \mathbf{F}_j + \boldsymbol{\tau}_j \quad (1b)$$

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. $\boldsymbol{\tau}_j \in \mathbb{R}^3$ 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 [8]

$$\mathbf{H}(\mathbf{q}, \mathbf{v}) = \mathbf{A}(\mathbf{q})\mathbf{v} \quad (2)$$

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. Thus we can use the six equations (1a-1b) to describe the robot dynamics. 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.

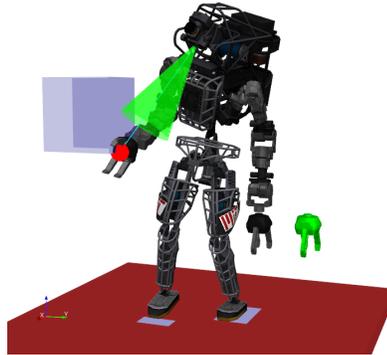


Fig. 2: Solving inverse kinematics problem with different types of kinematic constraints. The left foot and the right foot toes are constrained to lie within the shaded regions. A point (red sphere) on the right hand is constrained to be within the shaded bounding box. The head camera gazes at the point (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.

C. Collision model

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. We wish to ensure that the distance between each pair of potentially colliding geometries remains larger than some minimum allowable distance. While the distance between each pair of collision geometries can be computed efficiently [2], [1], 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, we can combine those constraints using a hinge-loss-like function. Schulman et. al. use a true hinge-loss function for a similar purpose [10]. 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. Applying this function to the difference of the distance between a pair of geometries and the minimum allowable distance yields a penalty term for that pair of geometries. This penalty is zero if and only if those geometries are separated by more than the minimum distance. By taking the sum of the penalties for all potential collision pair, we obtain a quantity which is zero if and only if a given configuration satisfies the minimum distance constraint for the entire robot.

D. Trajectory Optimization

To compute a feasible motion plan, we sample the trajectory with certain number of knot points, with the time interval between the adjacent knot points being flexible, and transcribe the differential equations of the simple dynamics (1a-1b) to algebraic equations using numerical integration. We also include the kinematics constraints described in sections II-B and II-C. We solve this trajectory optimization problem as a nonlinear program. This problem has very sparse gradients, since most constraints only depend on



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

Fig. 3: Snapshots of Atlas during stance

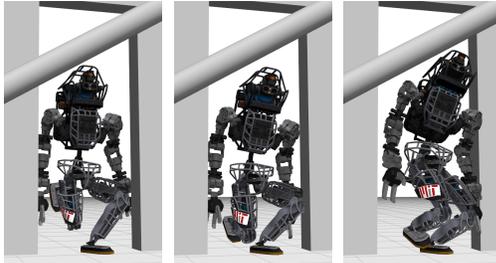


Fig. 4: Atlas goes through an obstructed door

variables at a single knot point or two adjacent knot points. It can be solved efficiently with modern solvers like SNOPT [3], which we use for the examples in Section III.

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 [9]. In this paper we apply the same idea to our motion planning algorithm.

III. RESULT

For our numerical experiments, we primarily use a dynamic model of the Atlas humanoid robot, built by Boston Dynamics, Inc for use in the DARPA Robotics Challenge. We show some snapshots of running, playing monkey bar, climbing salmon ladder. To show the generality of our algorithm, we also computed periodic gaits of a quadruped, the Little Dog.

REFERENCES

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

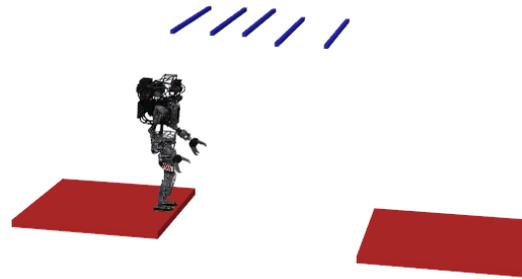


Fig. 5: A snapshot of the monkey bars

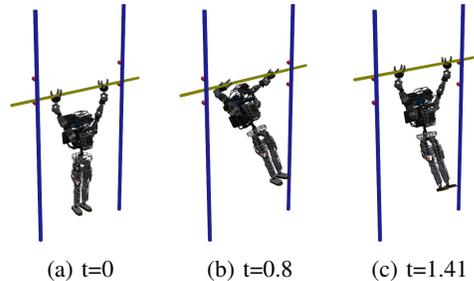


Fig. 6: Snapshots of climbing salmon ladder

- [2] 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.
- [3] 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.
- [4] 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.
- [5] 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.
- [6] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, 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.
- [7] 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.
- [8] David E. Orin, Ambarish Goswami, and Sung-Hee Lee. Centroidal dynamics of a humanoid robot. *Autonomous Robots*, (September 2012):1–16, jun 2013.
- [9] 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.
- [10] 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.



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

Fig. 7: Snapshots of the flight phase during LittleDog running