Dynamic whole-body robotic manipulation

Yeuhi Abe, Benjamin Stephens, Michael P. Murphy, Alfred A. Rizzi
Boston Dynamics, 78 4th Ave., Waltham, MA, USA 02451

ABSTRACT

The creation of dynamic manipulation behaviors for high degree of freedom, mobile robots will allow them to accomplish increasingly difficult tasks in the field. We are investigating how the coordinated use of the body, legs, and integrated manipulator, on a mobile robot, can improve the strength, velocity, and workspace when handling heavy objects. We envision that such a capability would aid in a search and rescue scenario when clearing obstacles from a path or searching a rubble pile quickly. Manipulating heavy objects is especially challenging because the dynamic forces are high and a legged system must coordinate all its degrees of freedom to accomplish tasks while maintaining balance. To accomplish these types of manipulation tasks, we use trajectory optimization techniques to generate feasible open-loop behaviors for our 28 dof quadruped robot (BigDog) by planning trajectories in a 13 dimensional space. We apply the Covariance Matrix Adaptation (CMA) algorithm to solve for trajectories that optimize task performance while also obeying important constraints such as torque and velocity limits, kinematic limits, and center of pressure location. These open-loop behaviors are then used to generate desired feed-forward body forces and foot step locations, which enable tracking on the robot. Some hardware results for cinderblock throwing are demonstrated on the BigDog quadruped platform augmented with a human-arm-like manipulator. The results are analogous to how a human athlete maximizes distance in the discus event by performing a precise sequence of choreographed steps.

Keywords: Mobile Manipulation, Dynamic Manipulation, Legged Robotics, BigDog, Dynamic Lifting, Dynamic Throwing, Whole Body Manipulation

1. INTRODUCTION

As long as mobile robots have been utilized for mobile manipulation tasks in the field, operators have been pushing them to their extreme limits. The success of robotic manipulation solutions in dangerous arenas, such as the defeat of improvised explosive devices (IEDs), has prompted their widespread use, and led to their use for increasingly challenging tasks. Unfortunately, the limited payload capacity for mobile manipulator platforms limits their applicability for many scenarios. One solution is to build stronger manipulators; however, this will result in a heavier platform, higher system cost and limited mission duration due to power depletion.

An approach to overcome these limitations that we examine in this paper focuses on utilizing all of the degrees of freedom of a robot to augment the manipulator capabilities, potentially resulting in greater payload capacity, greater reach, and faster end effector speed, all without requiring hardware enhancements. Although this process has some applicability to wheeled and tracked vehicles, we choose to focus on legged systems, as these robots typically have many high-force, large workspace degrees-of-freedom in the robot's legs, which are normally utilized solely for mobility. Therefore, there is potential to increase manipulation performance by using whole-body behaviors and control strategies that repurpose the existing hardware to amplify manipulation capabilities.

A major goal of this work is to transition robotic manipulation behaviors from the traditional deliberate and precise movements found in industrial robot arms toward a more biological, or athletic regime. It is clear that whole-body

Unmanned Systems Technology XV, edited by Robert E. Karlsen, Douglas W. Gage,
Charles M. Shoemaker, Grant R. Gerhart, Proc. of SPIE Vol. 8741, 87410V

strategies are used by humans and animals to increase, efficiency, speed, etc. In short, we are working toward more natural manipulation behaviors.

Figure 1. A mission concept rendering: the Big Dog quadruped robot, equipped with a manipulator, searches through rubble at the site of a disaster.

In the past [17] we have demonstrated two types of heavy manipulation behaviors: lifting and throwing. In the heavy lift behavior, the robot moves a mass from the ground to the highest point in the workspace. The heavy lift behavior is similar to the Olympic Weightlifting Snatch or Clean and Jerk events where athletes attempt to lift a weighted bar from the floor to a position extended overhead. In these events, humans are able to lift significantly heavier loads than their arms are capable of manipulating. This is accomplished by utilizing whole-body motions that inject energy from the legs into inertia of the payload, pulling the mass through areas of kinematic weakness using momentum, and exploiting the increased load carrying capacity of the arms at singular configurations both in tension and compression.

Previously we have also explored dynamic throwing behaviors. The goal of this task is to throw a mass, in this case a standard cinder block, as far as possible. Much like the lifting task, there are many athletic analogues, including Olympic Hammer Throw, Discus, Javelin Throw, and Shot putting. All of these sports involve injecting kinetic energy into a heavy object in order to maximize throw distance, and to do so, the athletes twist, spin, run, and extend their bodies to inject energy, manage momentum, and maintain balance. These are the characteristics we hope to imbue in our robotic control.

In this paper we focus primarily on improvements to our throwing behavior. In contrast to our previously demonstrated behaviors, where the robot's feet remained stationary, our new throwing behavior takings coordinated steps to enhance balance, thus enabling more aggressive throwing. By taking steps, similar to a human athlete, we increase the maximum throwing distance by over a meter while also improving dynamic stability.

The incorporation of stepping into our behavior also complicates controller design. While the robot throws the cinderblock, the large force exerted on the robot "throw" its body, which must be subsequently "caught" by its legs. This "throw/catch" metaphor is central to our behavior design. Due to the underactuated nature of legged dynamics, the task of stepping itself has been compared to juggling one's own body upon the ground or self-juggling [18]. Our problem is then complicate by the fact that this self-juggling must be coordinated with the motion of the manipulator. To accomplish this feat, we pre-plan trajectories for the robot, offline by solving an optimal control problem. The resulting feed-forward terms are used to inform the controller online during behavior execution.

1.1 Related Work

Typically, robotic mobile manipulation has meant a manipulator system mounted to a mobile base, with the two systems operated relatively independently. Recently, researchers have demonstrated a technique for performing fine manipulation from a platform which is not statically stable (segway RMP) [1]. In this example, the techniques developed demonstrate an ability to perform manipulation despite the presence of a mobile platforms rather than enhancing their manipulation performance by utilizing their mobile base. There have been a few exceptions to this approach, most notably [2,3], which take advantage of the additional degrees of freedom provided by the mobility system to improve kinematic conditioning (avoid singularities) and maintain balance while resisting/producing external forces through the manipulation system.

A core difficulty in whole-body manipulation of heavy objects is motion planning. Since the actuator base is no longer attached to the ground and the payload mass has significant effect on the base motion, the control architecture must consider the fully coupled dynamics of the object-manipulator-base system. To maximize capability, the system must operate in a regime far from statically stable and near actuator limits, which introduces all the complexity associated with underactuated systems [11] and dynamic stability of legged robots [14]. The problem of finding motions that maximize task performance without exceeding physical limitations of the robot falls squarely in the domain of trajectory optimization[12,13], which is a notoriously difficult problem to solve for large, non-linear systems, such as legged robots. For our throwing behavior, we formulate our trajectory optimization problem using a direct shooting approach [12]. The problem is solved using the Covarience Matrix Adaptation (CMA) algorithm [10], which has recently been demonstrated [15,16] to be a robust and efficient way to solve such problems.

Even after plausible trajectories are found, there are still challenges in formulating a controller capable of performing the motions on the robot platform. In particular, there is the difficulty of resolving force and motion redundancy. In other words, choosing how to position and allocate forces among the entire collection of joints in order to satisfy constraints and ensure robustness to disturbances. In robotics and computer animation, researchers have explored prioritized task space control strategies to turn low-dimensional (i.e. abstract) task space commands into high-dimensional joint level controls [4,5]. Later, these approaches were extended to employ unilateral Coulomb friction models [6] with mixed, rather than prioritized, task space control. Another approach to whole-body manipulation is to decouple body and limb control. Rather than explicitly form a Jacobian that includes all of the robot's degrees of freedom [7], we could treat the base of the robot as if it was floating. Wrenches on the base are then converted into desired foot forces in a least-squares fashion [8]. The result is that force-control can be realized using a prioritized version of virtual model control [9].

2. MECHANICAL DESIGN

2.1 The BigDog platform

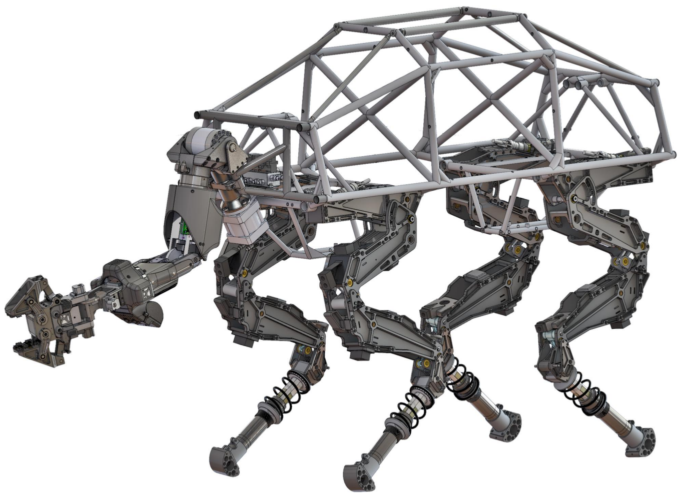

Figure 2. CAD rendering of the BigDog robot (internal components not shown) with an integrated hydraulic manipulator used as the platform for this work.

The base platform for our manipulation efforts in this work is the BigDog robot, a quadrupedal mobile robot capable of carrying up to 250 lb of payload, and highly mobile in outdoor unstructured environments. This robot (shown with manipulator in Figure), is an ideal platform for investigating dynamic manipulation due to the high force actuation of its legs (in excess of what is required for simple mobility), relatively large range of motion, and proprioceptive sensors.

2.2 Manipulator design

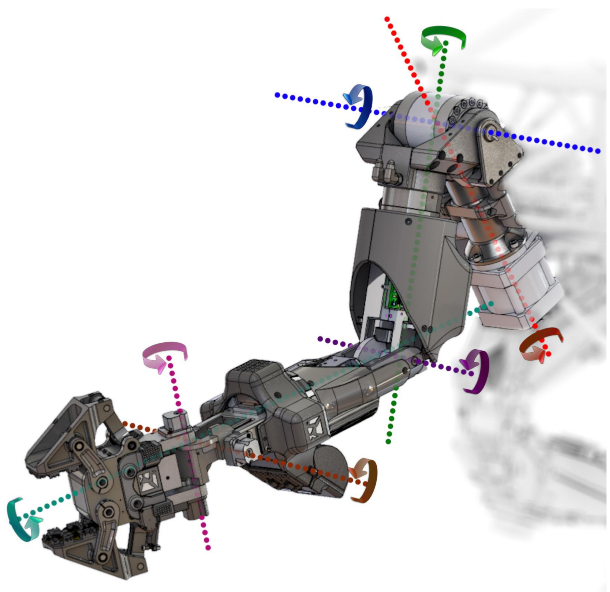

Figure 3. The hydraulic manipulator has seven rotational degrees of freedom (three shoulder, one elbow, three wrist), and a parallel jaw gripper mechanism.

After a comprehensive survey of commercially available manipulators, we opted to design and build a custom manipulator for the BigDog platform. Hydraulic actuation was chosen for its high force density and the availability of hydraulic power from the BigDog's existing power plant. The manipulator was designed with eight degrees of freedom including gripping. This configuration is similar in morphology and scale to a human arm, having a 3-dof shoulder, an elbow, and a 3-dof wrist. The morphology of the manipulator is illustrated in Figure . The goal was to enable the robot to appropriately interact with objects in manmade environments, while not providing sufficient strength to manipulate the target object (a 16.5kg cinder block) by static methods, thus requiring the use of whole-body manipulation techniques. The large number of joints also allows for more potential kinematic solutions, which enables the generation of more complex and natural behaviors than a simpler manipulator configuration. The parallel jaw gripping mechanism was designed with enough gripping force (approximately 250 lbf) to hold a cinder block under heavy acceleration to allow for dynamic throwing behaviors.

The manipulator is mounted on the BigDog platform in an orientation which allows for a large workspace below, in front, laterally, and above the robot, but avoids large overlap of the workspace with the body and legs. The manipulator can be assembled in either an 'elbow up' or 'elbow down' configuration, each with its own workspace.

3. BEHAVIOR GENERATION TECHNIQUES

3.1 Overview

In order to generate a robust dynamic throwing behavior that takes steps we start by layering our controller upon an existing trotting behavior. The trotting controller we start with is extremely robust, but lacks any knowledge of the throwing task. It naively generates feedback terms (on foot placements and body forces), which attempt to bring the robot state back to a stable operating point (i.e., trotting in place). As such, it serves as both a predictable entry point and a robust exit point for more aggressive body motion during throwing. The trotting controller has no notion of the manipulation task, however, by inserting deliberate feedforward terms into the trotting controller, it is possible to excite the dynamics of the system and induce rapid body accelerations away from the stable operating point. In order for these accelerations to aid in throwing, they must be carefully synchronized with the motion of the manipulator. To this end, optimal control techniques are used offline to compute the feedforward terms that accomplish the goal of throwing a cinderblock maximum distance, without the robot falling over.

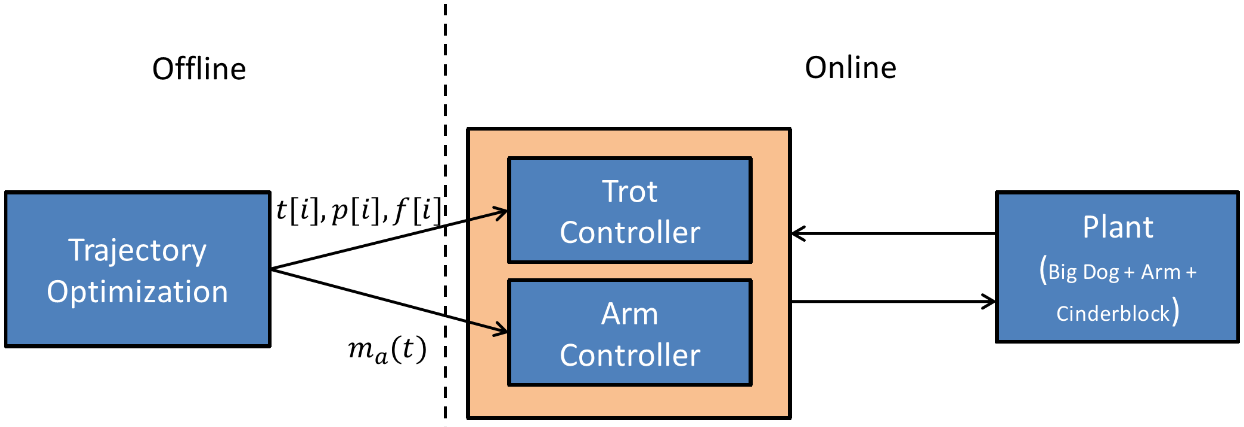

Figure 4: Robust throwing behavior is accomplished by leveraging offline optimization to determine feedforward, per step timing, foot locations, and body forces (resp. t[i], p[i], f[i] for step i) to produce desired full-body throwing behaviors. A synchronized motion, $m_a(t)$, of the arm is also generated and fed as input to the online control system.

3.2 Simplified Dynamics Model

Our behaviors use a simplified rigid-body dynamics model of the robot which abstracts the legs to be generalized force sources which act on a finite base of support formed by the convex hull of foot locations. The 6 degree-of-freedom (DOF) body and the 7 DOF arm result in a 13 DOF system, shown in Figure . This simplifies both offline optimization and online control by reducing the dimensionality of the plant by 20 DOF (i.e., 4 legs with 5 DOF each).

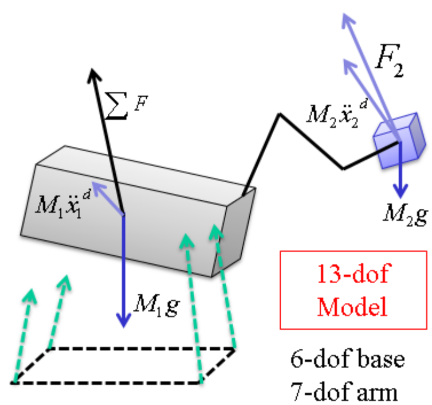

Figure 5: The dynamic behaviors assume a simplified model of the robot.

Given the generalized coordinates, $\vec{q}$, and derivatives, $\dot{\vec{q}}$ and $\ddot{\vec{q}}$, the generalized forces, $\vec{\tau}$, of this fully actuated system can be solved for analytically and take the form,

$$\vec{\tau} = \left(M_1(\vec{q}) + J(\vec{q})^T M_2 J(\vec{q})\right)\ddot{\vec{q}} - C(\vec{q},\dot{\vec{q}}) - G(\vec{q}) - J(\vec{q})^T M_2\left(\vec{g} + \dot{J}(\vec{q},\dot{\vec{q}})\dot{\vec{q}}\right) \quad (\#)$$

where $M_1(\vec{q})$ and M_2 are the inertia matrices of the robot and the manipulated mass, respectively, $J(\vec{q})$ is the Jacobian relating the velocity of the manipulated mass to the generalized velocities of the robot, $C(\vec{q},\dot{\vec{q}})$ is the centrifugal and Coriolis forces, and $G(\vec{q})$ is the gravitational force.

3.3 Full Body Control

Solving the inverse dynamics above only gives the arm torques and 6-dof generalized wrench on the body. Depending upon the number and configuration of legs on the ground, it may or may not be possible to achieve the desired body wrench. A force allocation scheme determines the nearest achievable wrench by formulating a quadratic program that solves for valid foot contact forces, $\vec{F}_i^d$. The quadratic program includes constraints on leg torques and friction cone constraints on the foot forces (as in [6].) The achievable wrench is computed by mapping the Cartesian point forces back to generalized coordinates, $\vec{\tau} = J_i^T \vec{F}_i^d$, where J_i is the Jacobian associated with each foot end effector.

3.4 Trajectory Optimization

We use an offline trajectory optimization to find a sequence of feedforward footsteps and body wrenches that are used as input to our online controller. Since the types of motion that we optimize maximize performance, the robot is regularly up against constraints on actuator strength and speed, kinematic workspace, and frictional contact dynamics. These constraints are highly non-linear, time dependent, unilateral, and difficult to differentiate, all attributes that cause problems for optimization. The trajectory optimizer must find throwing motions that avoid violating these constraints or the result may be impossible for the robot to perform. Finding a trajectory optimization scheme that can handle such a difficult optimization landscape is a major consideration.

Trajectory optimization schemes are numerical solutions to optimal control problems [12]. These schemes broadly fall into 2 categories; transcription methods, which directly solve for free variables that parameterize the state of the plant in time, and shooting methods, which represent only the control signal as a function of the free variables. When plausible trajectories of the robot are compactly representable, transcription methods may result in problems that are easier to solve. Conversely, when the motion is highly constrained, as is our case, transcription methods require increasingly complex representations of the state (with a greater number of free variables), which soon become impractical to solve.

For this reason, we employ a shooting strategy. In shooting, the dynamics of the plant are forward simulated using a set of control inputs that are free variables given to the optimizer. The optimizer examines the results of the simulation and assigns a score. The score heavily penalizes constraint violations and encourages desired behavior. For example, in the case of optimizing throwing behaviors, the desired behavior includes the robot staying up and throwing the object maximum distance. Due to the fact that the optimizer directly scores the result of a forward simulation, the approach is very flexible. Anything that can be simulated can be scored. For example, we can simply encode the throwing behavior's reliance on the underlying trotting controller by embedding the controller directly in the simulation. Instead of solving for actuator forces, we solve for a set of feedforward inputs given to the underlying controller. Using the shooting method, some constraints need not be included directly in the score. For example, actuator forces are simply clamped when they reach their limit in the forward simulation, ensuring that they are enforced, without the solver having any specific knowledge of them.

While the shooting method has many advantages and is quite flexible, one disadvantage is that the solution may be extremely sensitive to the initial point given to the optimizer, due to the divergent nature of forward simulation.

Furthermore, due to contact dynamics in the forward simulation, the score function may not be smooth or differentiable. This is true in our case and has motivated the use of the highly robust, sample-based Covariance Matrix Adaptation (CMA) algorithm. The CMA algorithm does not require gradients of the score function and has been demonstrated to be remarkably robust and efficient when solving tough optimization problems [10]. Recent work has demonstrated it to be a highly effective tool for optimizing the gait patterns of humans and animals [15,16]. We too have found it to be a remarkably effective given the correct problem formulation.

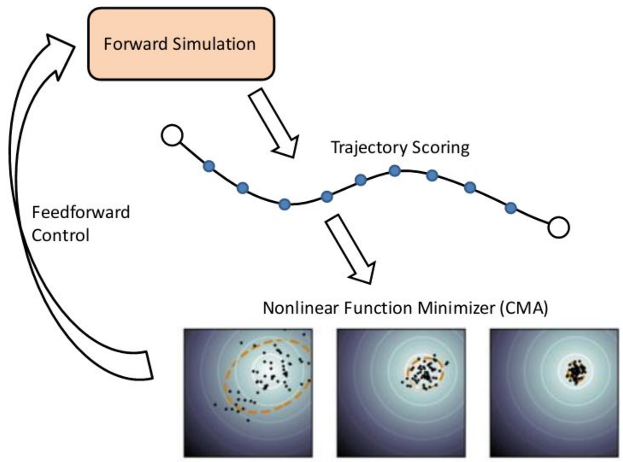

Figure 6: Our direct shooting approach to trajectory optimization uses a forward simulation of the Big Dog + arm + cinderblock system in conjunction with the Covarience Matrix Adaptation search algorithm to find feasible motions that maximize throwing distance.

3.5 Dynamic Throwing

The goal of the dynamic throwing behavior is to toss a 16.5kg cinderblock as far as possible using the manipulator-equipped quadruped. When throwing, it is not sufficient to simply command the manipulator arm to follow a predefined joint trajectory. This is because the heavy weight of the cinder block causes the relatively weak actuators of the arm to saturate. This is especially true of the weaker distal degrees of freedom, but even the stronger shoulder actuators quickly reach their limit when attempting to throw a heavy mass. To maximize throwing distance it becomes necessary to utilize a dynamic, full body motion that imparts force on the cinder block by pushing through the legs and pulling along the axis of the arm. The ability to coordinate the motion of the body in this manner is a key advantage of our legged platform. However this coordination is non-trivial since the legs must simultaneously keep the robot balanced during the behavior.

The throwing strategy we implemented was inspired by observation of human attempts to throw a heavy mass. To avoid the use of the weaker joints and to take advantage of body motion, humans keep their arms straight when throwing a large mass. Only the shoulder degrees of freedom are in motion. We use a similar strategy on the robot. Momentum is imparted to the cinder block by swinging it across the body from right to left. Over the course of the motion, maximum torque is applied to this degree of freedom. The motion of the y-axis shoulder joint is slaved to the motion of the x-axis joint in order to avoid collisions with the legs and the ground. The release of the gripper is triggered when the x-axis joint crosses a specified threshold. After release, the arm motion continues for a brief period of time and the wrist joint is oriented to avoid collision with the free-flying cinder block. The release threshold and swing start time are determined by the offline trajectory optimization.

Swinging the heavy mass of the cinder block generates considerable bias forces on the robot (both gravitational and centrifugal). Though the underlying trotting controller is amongst the most robust quadraped gaits known, it is unable to independantly compensate for the large body disturbances during throwing, without some ability to "look-ahead" and plan foot steps that anticipate the disturbance. As previously discussed, this is the job of the trajectory optimizer. The optimizer provides the trotting controller with feedforward body velocities, forces, and foot step locations. The trotting controller naively applies its own stabilizing feedback on top of the feedforward terms. We find that this helps compensate for unmodeled dynamics and improves tracking performance.

4. RESULTS AND DISCUSSION

4.1 Throwing result

Our dynamic throwing behavior is able to launch a full-weight cinderblock over 5 meters. This is nearly a meter further than previously demonstrated [17] without taking steps. The behavior coordinates 27 internal degrees of the freedom of the robot to produce a full-body motion that looks both athletic and aggressive. The actual throw occurs in just under 2 seconds.

The resulting behavior occurs in three stages. In the first stage the robot picks up the cinderblock and trots in place to ensure a consistent initial state from which to start the throw. In the second stage the robot begins a sequence of 3 diagonal-pair foot placements and swings the arm, as determined by the trajectory optimizer. In the last stage, the robot releases the block, finishes the step sequence and returns to the nominal trotting behavior.

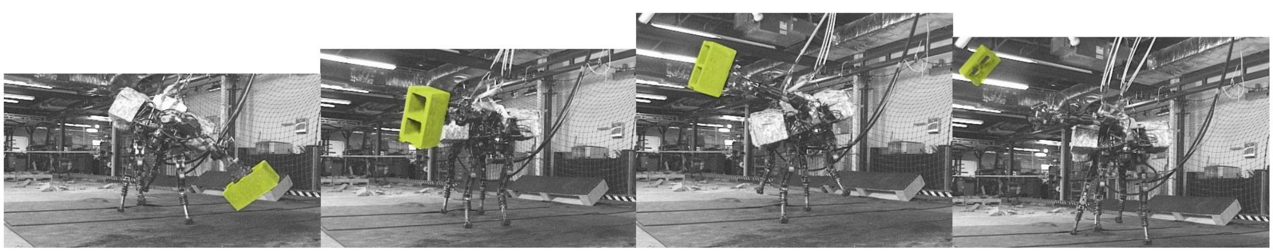

Figure 7: The Big Dog + arm robot throwing a 16.5 kg cinderblock. Video is available at: https://www.youtube.com/watch?v=2jvLalY6ubc.

Maximum throwing is achieved by careful coordination between the body and the manipulator. At the start of the throwing sequence, before the arm even starts to swing, the robot generates foot forces to produce a large yaw moment about the body. This starts accelerating the block in the direction of travel. Once the arm starts swinging, the motion of the arm counteracts the yaw and produces a large roll about the body. Simultaneously, the robot shifts its center of mass backwards to counteract the centrifugal force produced by the swinging block. Once the block is released, the robot pushes the body upward and steps out quickly to "catch" itself, preventing a fall.

4.2 "Throwing" and "catching" the body

A useful metaphor which has guided our controller design is the idea of a "throw" followed by a "catch". Here, the throw referred to is not of the cinderblock, but of the body of the robot itself. During the stepping sequence, the body is thrown off balance in a way that can be somewhat anticipated, but not fully controlled due to the underactuated nature of stepping, especially when the degrees of freedom in the legs are already against kinematic and actuation limits. Not unlike a thrown baseball, the motion of the robot's body during the stepping sequence is hard to change once started. As in baseball, the ball can be thrown harder and more aggressively if a large glove is used to catch it. With regard to this metaphor, the underlying trotting controller is the equivalent of a big glove into which we can aggressively throw the body. We attribute a large part of the success of our behavior to the robustness of our trotting controller and its ability to serve as both a consistent initial state and a large "target" into which the trajectory optimizer "throws" the body.

4.3 Comparison to throwing without stepping

We have found that the strength of the legs is a fundamental limitation on how far the robot can "throw" the body and thus the cinderblock. In previous work we have demonstrated a similar throwing behavior without using the legs to take steps. An advantage of this approach is that there are more legs are on the ground for a longer period of time, enabling the robot to push harder on the body. However, having more legs on the ground for longer also limits the kinematic workspace of the body. If the body is pushed too hard, the legs my not be able to reach far enough to support the body during the "catch" phase of the motion. Indeed, during our previous experiments without taking steps, the forces applied to the body were limited by exactly these constraints. When taking steps, the robot cannot push as hard instantaneously, but is able to achieve a higher body speed by pushing less hard for longer and traveling farther, ultimately generating more velocity on the cinderblock prior to release. The trade-off between kinematic workspace and ground force relates to the pattern and duration of feet placements on the ground. This relationship is non-trivial and difficult to include as a variable in trajectory optimization [15]. In our case, stepping always occurs in diagonal pairs. There are always two legs on the ground at any given time. However, in the future, further experimentation with alternative stepping patterns would be an interesting avenue to pursue.

5. CONCLUSIONS

We have described our recent advancement in developing high degree of freedom, dynamic manipulation behaviors for heavy objects. We have demonstrated superior dynamic throwing by coordinating the lower body motion with the manipulator movement. By using an offline trajectory optimization to plan step placements, we have increased the range of throwing behaviors and demonstrated a promising framework for whole-body, manipulation behavior development.

We have also framed our behavior in the context of a "throw" and "catch" metaphor which we believe elucidates many of the key concepts involved.

In the future, using whole body motion for dynamic manipulation should allow greater payload capacity and enable new capabilities for mobile robots, or allow for manipulation hardware to be miniaturized and made more cost effective while still matching the performance of existing systems that use traditional manipulation techniques. We are already taking steps to bring these behaviors outside the lab setting into the real world. We have plans to integrate vision-based sensing into the Big Dog + arm platform to allow the robot to autonomously identify objects, pick them up and manipulate them to accomplish a mission. Although we have successfully demonstrated one-off, athletic throwing behaviors, a significant challenge remains in extending the techniques such that a robot system can perform generalized manipulation behaviors online in the field.

REFERENCES

[1] Stilman, M., Olson, J., Gloss, W., "Golem Krang: Dynamically Stable Humanoid Robot for Mobile Manipulation," Proceedings of the 2007 IEEE International Conference on Robotics and Automation (2010).

[2] Tooru et al., "Whole Body Manipulation with a Quadruped Robot. Avoidance of Increased Mobility Singularity," Journal of the Robotics Society of Japan, (2002).

[3] Deegan, P., Thibodeau, B. J., and Grupen, R., "Designing a Self-Stabilizing Robot For Dynamic Mobile Manipulation," (2006). http://handle.dtic.mil/100.2/ADA459932

[4] Abe, Y. and Popovic', J., "Interactive Animation of Dynamic Manipulation," ACM SIGGRAPH / Eurographics Symposium on Computer Animation (2006).

[5] Sentis, L. and Khatib, O., "A Whole-Body Control Framework for Humanoid Operating in Human Environments," In Proc. of the IEEE International Conference on Robotics and Automation, pp. 2641-2648 (2006).

[6] Abe, Y., da Silva, M., and Popovic, J., "Multiobjective Control with Frictional Contacts," J. ACM Siggraph/Eurographics Symposium on Computer Animation (2007).

[7] Shkolnik, A. and Tedrake, R., "Inverse kinematics for a point-foot quadruped robot with dynamic redundancy resolution," In Proceedings of the 2007 IEEE International Conference on Robotics and Automation (2007).

[8] Nelson, G.M. and Quinn, R.D., "Posture control of a cockroach like robot," In IEEE Int. Conf. on Robotics and Automation. 157-162 vol 1. (1998).

[9] Pratt, J. "Virtual Model Control of a Biped Walking Robot," M.Eng. Thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge MA, (1995).

[10] Hansen, N., "The CMA Evolution Strategy : A Comparing Review," in Towards a New Evolutionary Computation: Advances on Estimation of Distribution Algorithms, 102(2006), 75-102 (2006).

[11] Spong M. W., "Underactuated mechanical systems," in Control Problems in Robotics and Automation, ser. Lecture notes in control and information sciences 230, B. Siciliano and K. P. Valavanis, Eds. Springer-Verlag (1997)

[12] Betts, J. T., "Practical Methods for Optimal Control Using Nonlinear Programming", SIAM Press, (2001)

[13] Wang, J. M., Fleet, D. J., Hertzmann, A. "Optimizing Walking Controllers". ACM Transactions on Graphics 28, 5, (2009)

[14] Raibert, M. H. "Legged robots," Commun. ACM, vol. 29, no. 6, (1986)

[15] Wampler, K., Popovic, Z. "Optimal gait and form for animal locomotion", ACM Trans. on Graphics, (2009)

[16] Wang, J. M., Fleet, D. J., Hertzmann, A. "Optimized Walking Controllers" ACM Trans. on Graphics, (2009)

[17] Murphy, M. et al. "High degree-of-freedom dynamic manipulation" SPIE Defense, Security, and Sensing, (2012)

[18] Mason, M., "Mechanics of Robot Manipulation" A Bradford Book, 2001. Print.