

Learning Object Models for Whole Body Manipulation

Mike Stilman,¹ Koichi Nishiwaki,² Satoshi Kagami²

¹ The Robotics Institute
Carnegie Mellon University
5000 Forbes Ave., Pittsburgh, PA 15213, USA
robot@cmu.edu

² Digital Human Research Center
National Institute of Advanced Industrial Science and Technology
2-41-6 Aomi, Koto-ku, Tokyo, Japan 135-0064
{k.nishiwaki, s.kagami}@dh.aist.go.jp

Abstract—We present a successful implementation of rigid grasp manipulation for large objects moved along specified trajectories by a humanoid robot. HRP-2 manipulates tables on casters with a range of loads up to its own mass. The robot maintains dynamic balance by controlling its center of gravity to compensate for reflected forces. To achieve high performance for large objects with unspecified dynamics the robot learns a friction model for each object and applies it to torso trajectory generation. We empirically compare this method to a purely reactive strategy and show a significant increase in predictive power and stability.

I. INTRODUCTION

A number of successful approaches to humanoid locomotion simplify the problem to controlling the robot center of mass. Consider a similar approach to the task of humanoid manipulation. Manipulated objects may be flexible or articulated due to joints or casters. Unlike robots, these objects rarely have pre-existing models. Yet, we show that simple low order differential equations may be sufficient to express object dynamics. In the context of humanoid manipulation, we evaluate a data driven algorithm that learns and uses simple object models to increase performance.

We are primarily interested in manipulation of large objects such as carts, tables, doors and construction materials. Small objects can be lifted by the robot and modeled as additional robot links. Heavy objects are typically supported against gravity by external sources such as carts, door hinges or construction cranes. Yet, neither wheeled objects nor suspended objects are reliable sources of support for the robot. Large, heavy objects are interesting because they require the robot to handle significant forces while maintaining balance. Using learned models we show that even 55kg objects, equal to the robot's mass, can be moved along specified trajectories.

II. RELATED WORK

Early results in humanoid manipulation considered balance due to robot dynamics. Inoue [1] changed posture and stance for increased manipulability and Kuffner [2] found collision-free motions that also satisfied balance constraints. These methods did not take into account object dynamics. In contrast, Harada [3] extended the ZMP balance criterion for pushing on an object with known dynamics. Harada [4] also proposed an impedance control strategy for pushing objects during the double support phase of walking. We focus on continuous manipulation during all walking phases.

With the introduction of preview control by Kajita[5], Takubo [6] applied this method to adapting step positioning

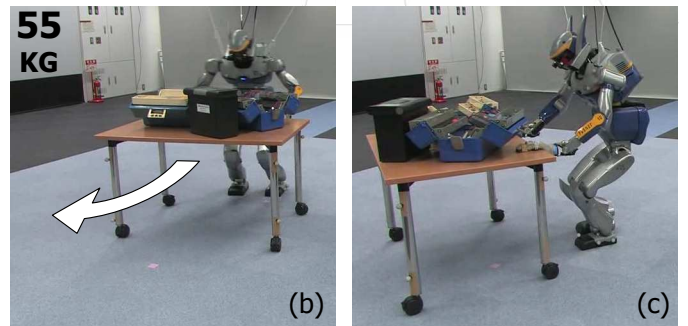
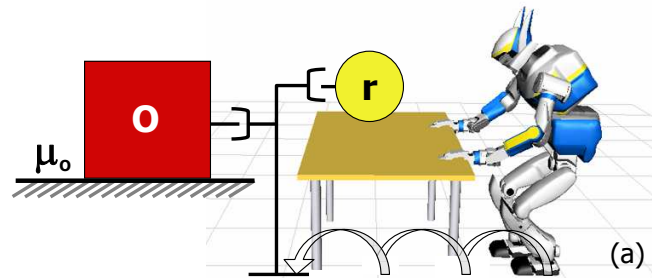


Fig. 1. (a) Abstract model represents a grasped object and the robot center of gravity. (b-c) Successful manipulation of a 55kg object on casters without prior knowledge of mass or friction.

while pushing on an object. Nishiwaki[7][8] proposed that the external forces from pushing could be handled by rapid trajectory regeneration. Yoshida [9][10] locally modified the planned path for a light carried object to avoid collisions introduced by applying preview control. Our work extends beyond pushing and modification to realizing a desired trajectory for a heavy object. Furthermore, in contrast to assuming that objects are known or external sources of error, we learn about their response to our force inputs.

Recently, most studies of interaction with unknown objects have been kinematic. Krotkov[11] and Fitzpatrick[12] studied impulsive manipulation to detect the affordances of object through various sensors. Stoychev[13] considered learning to use objects for specific behaviors and Christiansen[14] learned to manipulate an object between a set of discrete states. However, when manipulating large objects the controller must take into account the continuous dynamic effect these objects have on the balance and stability of the robot.

While our focus is on learning the dynamic model of an unknown *object*, this paper is closely related to modeling *robot* dynamics. Atkeson [15] summarizes approaches to

learning or adapting parameters to achieve precise trajectory following. Friction modeling, in particular has been studied extensively as summarized by Canudas [16] and Olsson[17]. More sophisticated methods for learning the dynamics of tasks in high dimensional spaces are studied by Atkeson, Moore and Schaal [18][19].

III. MOTIVATION FOR LEARNING MODELS

Our goal is to reliably manipulate massive objects along desired trajectories. The robot is not informed of the object dynamics, but it must handle large time-varying forces reflected from the object in order to maintain balance. This section introduces two challenges that modeling addresses: noise in force sensor readings and the dependence of balance control on future information.

The first challenge is common to many robot systems. Figure 2(a) shows the noise in force sensor readings from the robot hands during whole-body manipulation. While high frequency forces have no significant impact on balance, the low frequency force response must be compensated. The complication is that online filtering introduces a time delay of up to 500 ms for noise free data as shown in Figure 2(b). Modeling gives us the force estimate in Figure 3(b), a low frequency response without time delay.

For balancing robots such as humanoids, we not only require accurate estimates of current state but also of future forces. Typically, a balance criterion such as center of pressure location (ZMP) is achieved by commanding a smooth trajectory for the robot COM. [5] demonstrates that accurate positioning of ZMP requires up to two seconds of future information about its placement. Since external forces at the hands create torques that affect the ZMP, they should be taken into account two seconds earlier, during trajectory generation.

In summary, the purpose of modeling is to use known information such as the target object trajectory to accurately predict its low frequency force response in advance. The predicted response is used to generate a smooth trajectory for the robot COM that satisfies the desired ZMP.

IV. MODELING METHOD

Environment objects vary in kinematics, dynamics, compliance and friction. The tables and chairs used in our experiments are on casters. Each caster has joints for wheel orientation and motion. Among other properties, object dynamics depend on wheel orientation. Currently, we do not have a perception system that can detect and interpret this level of detail. We approach modeling from the perspective of finding a simple and effective strategy that maps an object trajectory to a predicted force response.

Despite the complex kinematic structure of a humanoid robot, the robot is often modeled as a point mass attached to the stance foot with prismatic joints. Likewise, an object can be modeled as a point mass in Eq. 1. Given experimental data we can compute the mass and friction for an object and use them to predict force. However, due to uncertainty in caster orientation and the low velocities of manipulation our experiments did not result in a consistent relationship between

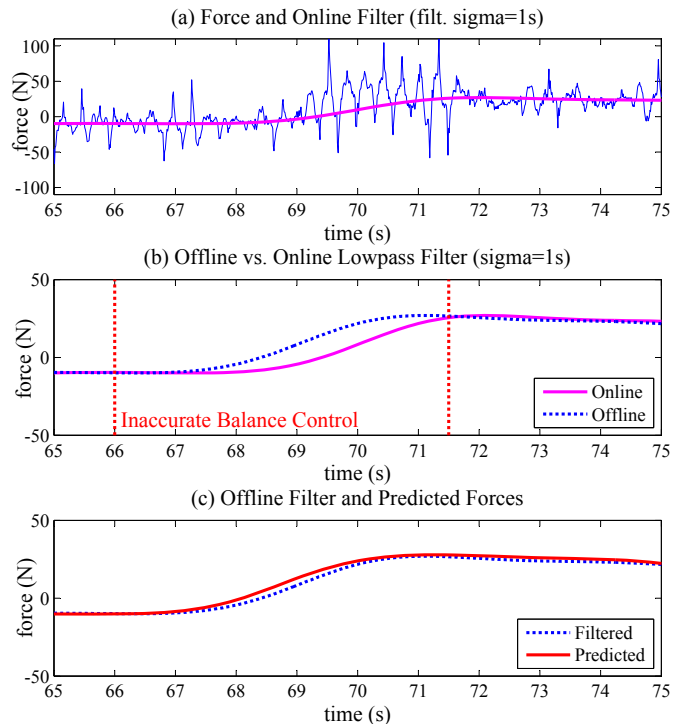


Fig. 2. (a) shows two degrees of online filtering. (b) Notice the time delay of low-pass filtering and the error window for preview control. (c) Forces predicted by our model.

acceleration and force. Consequently we chose to base our model solely on viscous friction as given in Eq. 2.

$$f^t = m_o \ddot{x}_o^t + c \dot{x}_o^t \quad (1)$$

$$f^t = c \dot{x}_o^t \quad (2)$$

To find c that satisfies this relationship we applied least squares regression on collected data. We executed a trajectory that displaced the object at distinct velocities, \dot{x}_o^t , and measured the force at HRP-2s hands, f^t , at millisecond intervals. The collected data was represented in Eq. 3. The term b was used to remove bias which appeared as a constant force offset allowed by impedance control after grasp.

$$\begin{bmatrix} \dot{x}^1 & \dot{x}^2 & \dots & \dot{x}^n \\ 1 & 1 & \dots & 1 \end{bmatrix}^T \begin{bmatrix} c \\ b \end{bmatrix} = [f^1 \quad f^2 \quad \dots \quad f^n]^T \quad (3)$$

The solution to this set of over-constrained equations is found simply by applying the right pseudo-inverse. During data collection we used a reactive balancing strategy which assumed a constant force response during a .15s trajectory cycle. This approach was sufficiently stable for brief interactions.

V. IMPLEMENTATION DETAILS

Section IV gave an overview of the method applied in this paper. We now expand on the control strategy for whole body manipulation that takes into account the object model. The controller consists of three significant elements:

- Decoupling the object and robot centers of mass.
- Trajectory generation satisfying ZMP and object motion.
- Online feedback for balance and compliance.

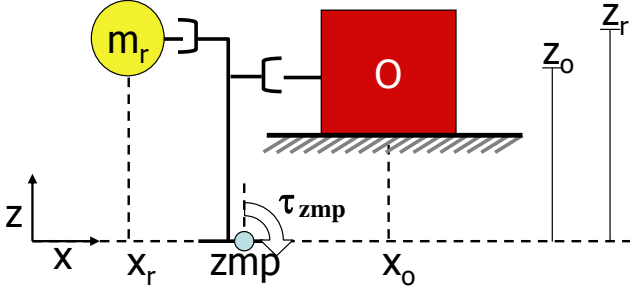


Fig. 3. Model of the robot and object used in our work.

Instantiating these three components lifts control from the 30 dimensional space of robot joints to a higher level of abstraction: a system that realizes a single object trajectory.

A. Decoupled Positioning

At the highest level, we represent the manipulation task as a system of two bodies. The object, o , and robot, r , are attached by horizontal prismatic joints to a grounded stance foot. The stance foot position changes in discrete steps at a constant rate $k = 900ms$. Section V-B computes independent workspace trajectories for \mathbf{x}_r and \mathbf{x}_o . To implement this abstraction we describe how workspace trajectories map to joint space.

We start the mapping by defining the trajectories for hands and feet relative to the object. Due to rigid grasp manipulation, the hand positions, \mathbf{p}_{lh} and \mathbf{p}_{lr} remain at their initial displacements from \mathbf{x}_o . For simpler analysis, the stance foot \mathbf{p}_{st} is fixed relative to \mathbf{x}_o at each impact. The robot swing foot, \mathbf{p}_{sw} follows a cubic spline connecting its prior and future stance positions.¹

We also fix the trajectory for the robot torso, \mathbf{p}_{torso} relative to \mathbf{x}_r . Although the center of mass position, \mathbf{x}_r , is a function of all the robot links we assume that \mathbf{x}_r remains fixed to \mathbf{p}_{torso} after grasp. This assumption is relaxed in Section V-B through iterative controller optimization. Notice that *although many of the link positions are highly coupled, the two positions of interest \mathbf{x}_r and \mathbf{x}_o are not.*

Suppose we have workspace trajectories for both \mathbf{x}_o and \mathbf{x}_r . The former specifies trajectories for hands and feet and the latter defines \mathbf{x}_{torso} . Joint values that position the four ungrounded links are found with resolved rate control [20]. We solve inverse kinematics for angles in four kinematic chains:

$$\begin{array}{l|l} \mathbf{p}_{st} \rightarrow \mathbf{x}_r & \text{6 Stance leg} \\ \mathbf{x}_r \rightarrow \mathbf{p}_{sw} & \text{6 Swing leg} \end{array} \quad \begin{array}{l|l} \mathbf{x}_r \rightarrow \mathbf{p}_{lh} & \text{7 L arm} \\ \mathbf{x}_r \rightarrow \mathbf{p}_{rh} & \text{7 R arm} \end{array}$$

These solutions complete the mapping from any valid workspace placement of \mathbf{x}_r and \mathbf{x}_o to robot joints.

In all the applications of inverse kinematics we use analytical solutions to expedite calculations and avoid drift. The two chest joint values are constants that maximize the workspace. Redundancy in the arms is resolved by fixing elbow rotation about the line connecting the wrist and shoulder.

¹To achieve a fixed displacement from the object on each step, the object velocity is bounded by the maximum stride length and step rate. We restrict the values of \dot{x}_o in advance.

B. Trajectory Generation

Section V-A gave a mapping from commanded workspace positions of \mathbf{x}_r and \mathbf{x}_o to joint positions. We now focus on workspace control. Given a commanded trajectory for \mathbf{x}_o we compute a trajectory for \mathbf{x}_r that satisfies balance constraints.

We define balance by relating the zero moment point to stance foot position. Let x be the direction of object motion. z_o is the height of the hands and f is the reflected force. Eq. 4 introduces zmp as the ground point around which the torques due to gravity acting on \mathbf{x}_r , reflected force from accelerating \mathbf{x}_r and reflected force from the object sum to zero.

$$\tau_{zmp} = m_r g(x_r - zmp) - m_r \ddot{x}_r z_r - z_o f = 0 \quad (4)$$

Solving for zmp yields:

$$zmp = x_r - \ddot{x}_r \frac{z_r}{g} - \frac{z_o f}{m_r g}. \quad (5)$$

Dynamic balance requires zmp to remain in the robot support polygon. To maximize error tolerance we seek a trajectory that minimizes the distance between zmp and the stance foot center $zmp_d = x_{st}$. Recall that x_{st} , and thus zmp_d are known given a trajectory for \mathbf{x}_o . (S.V-A)

Let $J_0 = \sum_t (zmp_d^t - zmp^t)^2$ be the performance index for balance. Eq. 6 further defines β and β_d as functions of zmp_d and x_r respectively.

$$\beta_d = zmp_d + \frac{z_o f}{m_r g} \quad \beta = x_r - \ddot{x}_r \frac{z_r}{g} \quad (6)$$

Substitution yields $J_0 = \sum_t (\beta_d^t - \beta^t)^2$. Notice that zmp_d is the trajectory of foot centers and $\{z_o, m_r, g\}$ are constants. Hence assuming that f is known, the trajectory of future values for β_d is fully determined.

Suppose we interpret β as the observation of a simple linear system in x_r with the input \ddot{x}_r . For smoothness, we add squared input change to the performance index.

$$J = \sum_{t=1}^{\infty} Q_e (\beta^t - \beta_d^t)^2 + R (\ddot{x}_r^t - \ddot{x}_r^{t-1})^2 \quad (7)$$

We can now determine the optimal \ddot{x}_r with preview control [5]. At any time t we know the error $e(t) = \beta^t - \beta_d^t$, state $\mathbf{x}(t) = [x_r^t \dot{x}_r^t \ddot{x}_r^t]^T$ and N future β_d^i . Preview control finds the gains G_1 , G_2 and G_3 such that the incremental control in Eq. 8 minimizes J .

$$\Delta \ddot{x}_r^t = -G_1 e(t) - G_2 \Delta x_r^t - \sum_{i=1}^N G_3^i (\beta_d^{t+i} - \beta_d^{t+i-1}) \quad (8)$$

More is available in Appendix B of [21] summarizing [22]. The control $\Delta \ddot{x}_r$ is discretely integrated to generate the trajectory $\{\ddot{x}_r, \dot{x}_r$ and $x_r\}$ for \mathbf{x}_r . The trajectory for y_r is found by direct application of preview control since the object reflects no forces tangent to x .

Since \mathbf{x}_r is assumed to be fixed to the robot torso, the generated joint space trajectory still results in zmp tracking error. We incorporate this error into the reference trajectory and iterate optimization.

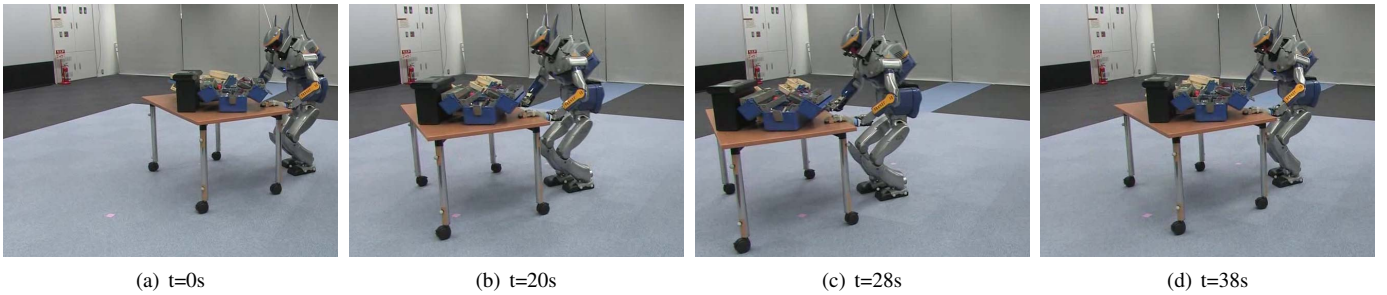


Fig. 4. Executed trajectory using the model-based approach with a 55kg loaded table. The robot pushes then pulls the table.

C. Online Feedback

Section V-B described the generation of a balanced trajectory for \mathbf{x}_r given \mathbf{x}_o . To handle online errors we modify these trajectories online prior to realization with robot joints. Online feedback operates at a $1ms$ cycle rate.

Accumulated ZMP tracking error can lead to instability over the course of execution. Therefore, a proportional controller modifies the acceleration of \mathbf{x}_r to compensate for ZMP errors perceived through the force sensors at the feet. These corrections are discretely integrated to achieve \mathbf{x}_r position.

The trajectory for \mathbf{x}_o , or the robot hands, is modified by impedance. We use a discrete implementation of the virtual dynamic system in Eq. 9 to compute the offset for \mathbf{x}_o that results from integrating the measured force error F .

$$F = m_i \ddot{\mathbf{x}}_o + d_i \dot{\mathbf{x}}_o + k_i (\mathbf{x}_o - \mathbf{x}_o^d) \quad (9)$$

Impedance serves two goals. First of all, we ensure that hand positioning errors do not lead to large forces pushing down on the object. Since the robot does not use the object for support, d_i and k_i are set low for the z direction.

Second, we prevent the robot from exceeding torque limits when the trajectory cannot be executed due to large reflected forces or un-modeled dynamics. The position gain for the x direction trades a displacement of $10cm$ for a $100N$ steady state force. This allows for precise trajectory following and soft termination when the trajectory offset exceeds force limits.

VI. EXPERIMENTS AND RESULTS

We conducted experiments on model-based whole body manipulation using a loaded table on casters, as shown in Figure 4 and 8. The robot grasped the table and followed a smooth trajectory for \mathbf{x}_o as generated from a joystick input.

Our results were compared to a rigid grasp implementation of a reactive approach to handling external forces presented in [8], which assumed that sensed forces would remain constant. Both methods recomputed the trajectory for \mathbf{x}_r every $150ms$, at which time the reactive strategy updated its estimated force.

The reactive method was applied first to gather data and learn an object model from Section IV. Brief experiments of less than $10s$ were necessary to collect the data. We applied both methods on a series of experiments that included a change of load such that the total mass ranged from $30kg$ to $55kg$.

A. Prediction Accuracy

First, we look at how well our model predicts force. The comparisons in this subsection use data from experiments that

are not used to build the model. The comparison in Table I shows that the mean squared error between modeled and measured force is lower than the error of assuming that force remains constant during the control cycle.

Since preview control takes into account future β_d , including predicted force, next we propose a more accurate prediction measure. Let β_d reflect the difference in predicted and actual force. Preview control is applied to find a trajectory that compensates for the simulated error. It generates an erroneous x_r displacement, x_{err}^{PC} , during the $150ms$ that the trajectory is active. x_{err}^{PC} is the expected trajectory error given the error in force prediction.

The comparison between the expected trajectory error, shown in Figure 6 and Table I, also favors the model based method. x_{err}^{PC} decreases if we assume a faster control cycle. However, even for a $20ms$ cycle, we found that error decreases proportionally for both controllers and the ratio of their MSE remains in favor of modeling.

B. System Stability

The accuracy of prediction has significant effect on the overall stability of the controlled system. Incorrect predictions affect the trajectories for \mathbf{x}_r and \mathbf{x}_o . First consider the resulting ZMP of the robot. While both controllers exhibit a slight offset in ZMP from grasping the object, the constant error can be removed with integral or adaptive control. A greater concern is the variance in this error. Figure 7 shows the increased noise in the control signal for ZMP when using the reactive control. Table II summarizes this effect.

An even clearer distinction between the two methods is directly reflected in the noise of the perceived force data. Table II also shows the variance in the noise given the off-line filtered signal. The difference in noise is clearly seen in Figure 5.

C. Summary

Our experiments show a significant improvement both in prediction accuracy and system stability when using the learned object model for control. One of the most convincing results is the accurate force prediction for a $55kg$ object in Figure 5(d). Also notice the low noise variance in sensed forces when using the model based controller.

We also evaluated the performance of the system with an inaccurate model. Due to online balance compensation and impedance it was possible to manipulate the $55kg$ table using the $30kg$ feed-forward model. However, as seen in Figure 9, the impedance controller saturates at the object force and the zmp error error grows significantly.

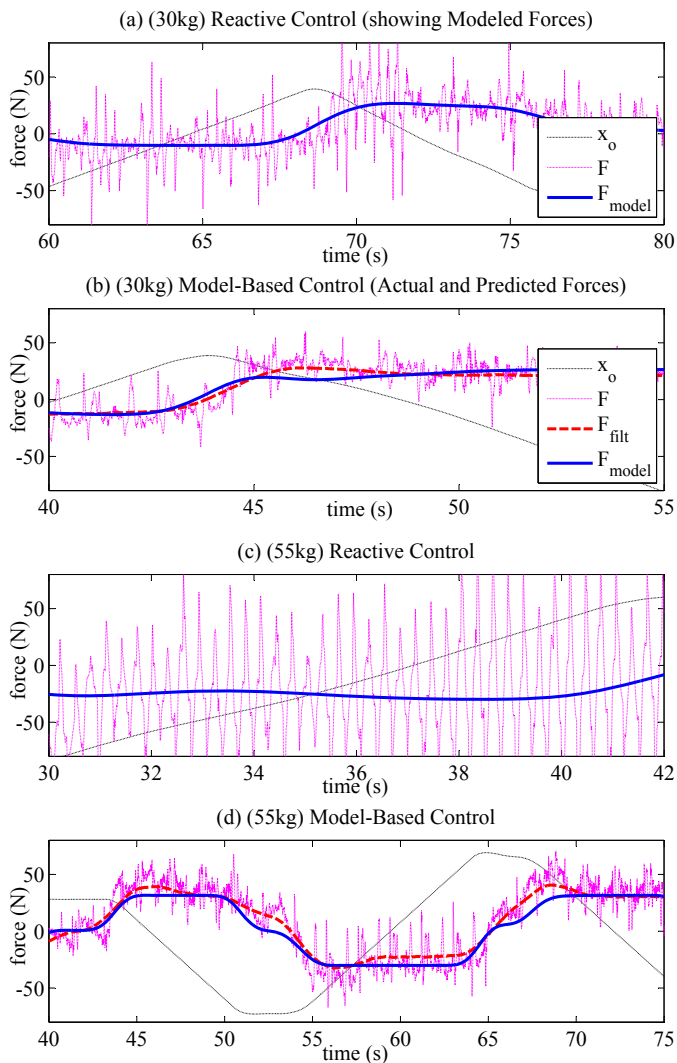


Fig. 5. (a,c) show forces during reactive manipulation. (b,c) use model-based method and reduce variance in experienced force.

VII. CONCLUSION

In this paper we have shown that it is possible to reliably manipulate unknown, large, heavy objects such as tables along specified trajectories with existing humanoid robots. We found that simple statistical methods such as least squares regression can be used to learn a dynamic model for the unknown object and use it to improve balance during manipulation.

Our estimated model of viscous friction was compared to a reactive method which assumed that forces would remain constant. Learning models proved to yield better prediction of forces and increase balance stability when used during whole body manipulation.

In future work, we are interested in merging the feed-forward model with feedback information. State estimators such as Kalman filters can be used to maximize the performance of the robot by combining information sources. Furthermore, adaptive control techniques could be applied in handling online changes to friction and mass.

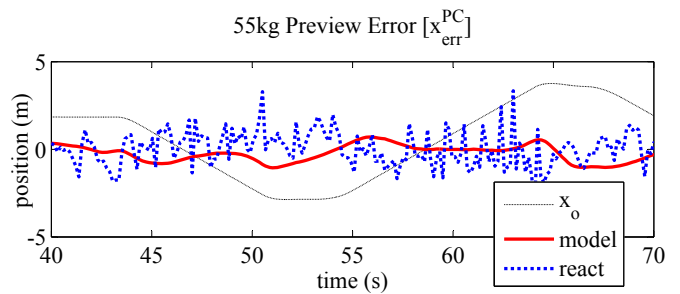


Fig. 6. Trajectory error introduced by preview control with erroneous prediction. Computed for each active trajectory period

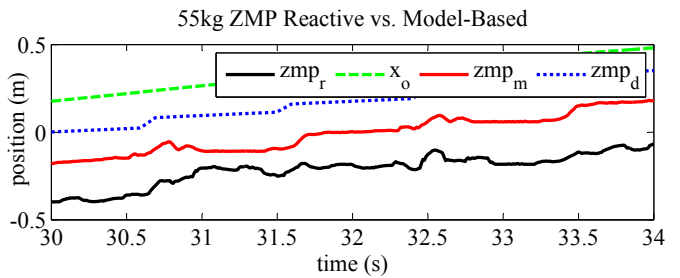


Fig. 7. Realized ZMP for identical reference trajectories. Plots are position shifted to show greater variance in reactive approach.

TABLE I

AVERAGE PREDICTION ACCURACY

| | MSE F_{err} (N) | | MSE x_{err}^{PC} (m) | |
|------|-------------------|-------|------------------------|-------|
| | model | react | model | react |
| 30kg | 4.44 | 9.19 | .427 | .674 |
| 55kg | 5.21 | 12.5 | .523 | .971 |

TABLE II

SYSTEM STABILITY

| | ZMP SD (m) | | Force SD (N) | |
|------|------------|-------|--------------|-------|
| | model | react | model | react |
| 30kg | .0214 | .0312 | 11.06 | 15.79 |
| 55kg | .0231 | .0312 | 12.15 | 46.25 |

VIII. ACKNOWLEDGEMENTS

We are grateful to James Kuffner and Chris Atkeson for their support and insight during the development of this work.

REFERENCES

- [1] K. Inoue, H. Yoshida, T. Arai, and Y. Mae. Mobile manipulation of humanoids: Real-time control based on manipulability and stability. In *IEEE Int. Conf. Robotics and Automation (ICRA)*, pages 2217–2222, 2000.
- [2] J.J. Kuffner, S. Kagami, K. Nishiwaki, M. Inaba, and H. Inoue. Dynamically-stable motion planning for humanoid robots. *Autonomous Robots*, 12(1), 2002.
- [3] K. Harada, S. Kajita, K. Kaneko, and H. Hirukawa. Pushing manipulation by humanoid considering two-kinds of zmps. In *IEEE Int. Conf. on Robotics and Automation*, pages 1627–1632, 2003.
- [4] K. Harada, S. Kajita, F. Kanehiro, K. Fujiwara, K. Kaneko, K. Yokoi, and H. Hirukawa. Real-time planning of humanoid robot's gait for force controlled manipulation. In *IEEE Int. Conf. on Robotics and Automation*, pages 616–622, 2004.
- [5] Shuji Kajita et. al. Biped walking pattern generation by using preview control of zero-moment point. In *IEEE Int. Conf. on Robotics and Automation*, pages 1620–1626, 2003.

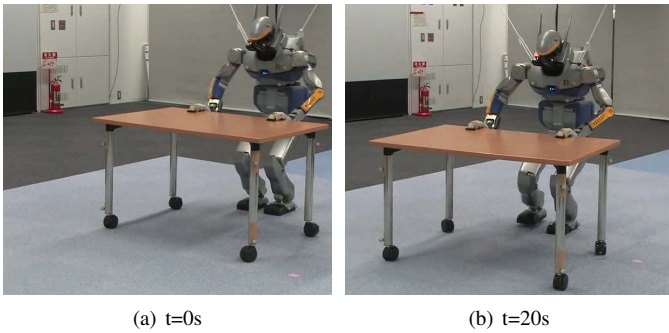


Fig. 8. Model-based trajectory execution with a 30kg table.

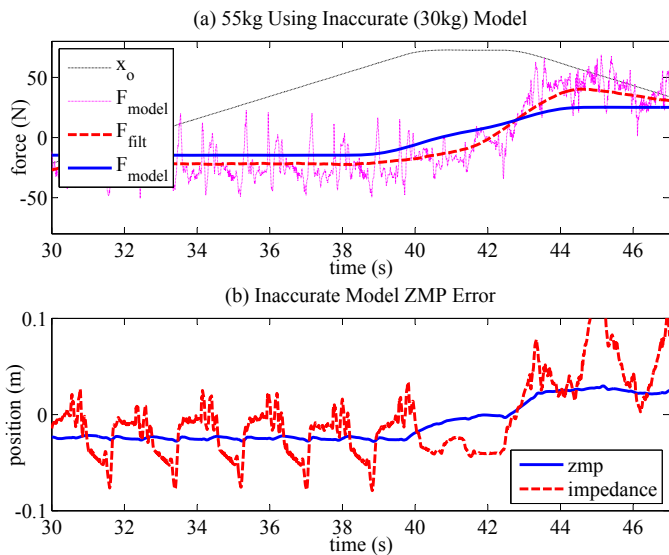


Fig. 9. (a) shows the forces predicted by the 30kg model on a 55kg experiment. (b) gives the impedance offset and zmp error.

[6] T. Takubo, K. Inoue, and T. Arai. Pushing an object considering the hand reflect forces by humanoid robot in dynamic walking. In *IEEE Int. Conf. on Robotics and Automation*, pages 1718–1723, 2005.

[7] K. Nishiwaki, W-K. Yoon, and S. Kagami. Motion control system that realizes physical interaction between robot's hands and environment during walk. In *IEEE Int. Conf. on Humanoid Robotics*, 2006.

[8] K. Nishiwaki and S. Kagami. High frequency walking pattern generation based on preview control of zmp. In *IEEE Int'l Conf. on Robotics and Automation (ICRA'06)*, 2006.

[9] E. Yoshida, I. Belousov, Claudia Esteves, and J-P. Laumond. Humanoid motion planning for dynamic tasks. In *IEEE Int. Conf. on Humanoid Robotics (Humanoids'05)*, 2005.

[10] E. Yoshida, C. Esteves, T. Sakaguchi, J-P. Laumond, and K. Yokoi. Smooth collision avoidance: Practical issues in dynamic humanoid motion. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2006.

[11] E. Krotkov. Robotic perception of material. In *IJCAI*, pages 88–95, 1995.

[12] P. Fitzpatrick, G. Metta, L. Natale, S. Rao, and G. Sandini. Learning about objects through interaction - initial steps towards artificial cognition. In *IEEE Int. Conf. on Robotics and Automation*, pages 3140–3145, 2005.

[13] A. Stoytchev. Behavior-grounded representation of tool affordances. In *IEEE Int. Conf. on Robotics and Automation*, pages 3060–3065, 2005.

[14] A. Christiansen, T. M. Mitchell, and M. T. Mason. Learning reliable manipulation strategies without initial physical models. In *Proc. IEEE Int. Conf. on Robotics and Automation*, 1990.

[15] C.H. An, C.G. Atkeson, and J.M. Hollerbach. *Model-Based Control of a Robot Manipulator*. MIT Press, 1988.

[16] C. Canudas de Wit, P. Nol, A. Aubin, and B. Brogliato. Adaptive friction compensation in robot manipulators: low velocities.

[17] H. Olsson, KJ Astrom, CC. de Wit, M. Gafvert, and P. Lischinsky. Friction models and friction compensation.

[18] Stefan Schaal Chris Atkeson, Andrew Moore. Locally weighted learning. *AI Review*, 11:11–73, April 1997.

[19] Andrew Moore, C. G. Atkeson, and S. A. Schaal. Locally weighted learning for control. *AI Review*, 11:75–113, 1997.

[20] D.E. Whitney. Resolved motion rate control of manipulators and human prostheses. *IEEE Transactions on Man Machine Systems*, 10:47–53, 1969.

[21] M. Stilman. Navigation among movable obstacles. Technical Report CMU-RI-TR-07-37, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, October 2007.

[22] T. Katayama, T. Ohki, T. Inoue, and T. Kato. Design of an optimal con-troller for a discrete time system subject to previewable demand. *Int. Journal of Control*, 41(3), 1985.