Proceedings of the IEEE Int. Conf. on Robotics and Automation (ICRA'08), May, 2008.

Humanoid Teleoperation for Whole Body Manipulation

Mike Stilman, Koichi Nishiwaki and Satoshi Kagami

***Abstract*— We present results of successful telemanipulation of large, heavy objects by a humanoid robot. Using a single joystick the operator controls walking and whole body manipulation along arbitrary paths for up to ten minutes of continuous execution. The robot grasps, walks, pushes, pulls, turns and re-grasps a 55kg range of loads on casters. Our telemanipulation framework changes reference frames online to let the operator steer the robot in free walking, its hands in grasping and the object during mobile manipulation. In the case of manipulation, our system computes a robot motion that satisfies the commanded object path as well as the kinematic and dynamic constraints of the robot. Furthermore, we achieve increased robot stability by learning dynamic friction models of manipulated objects.**

I. Introduction

This paper presents a humanoid robot manipulating large, heavy objects via teleoperation. Our experimentally validated system takes advantage of both human and robot strengths through shared control. The operator understands the environment and gives directional commands. The robot computes trajectories that satisfy kinematic and dynamic constraints to safely execute the motion. Our work makes progress towards applications in hazardous rescue or construction. Robots that move debris and materials would reduce the physical load on operators and keep them away from danger.

We use humanoid robots because their anthropomorphic design makes them suitable for interacting in human environments and intuitive for humans to operate. We also focus on 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, hinges or construction cranes. Large, heavy objects that do not support the robot require management of significant forces while performing manipulation.

The operator is not affected by the challenging domain. Our flexible task frame interface reduces the 38 degree of freedom humanoid control problem to the command space of a three axis joystick. We make it possible, even simple, for non-experts to perform complex manipulation as shown in Figure 1. Low demands on users imply significant challenges for robots. We give strategies that interpret operator commands and control the robot while maintaining balance, avoiding joint limits and handling external forces.

M. Stilman is with the Center for Robotics and Intelligent Machines, School of Interactive Computing, Georgia Tech, 85 5th Street NW,Atlanta, GA 30332, USA. mstilman@cc.gatech.edu

K. Nishiwaki and S. Kagami are with the 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}@aist.go.jp

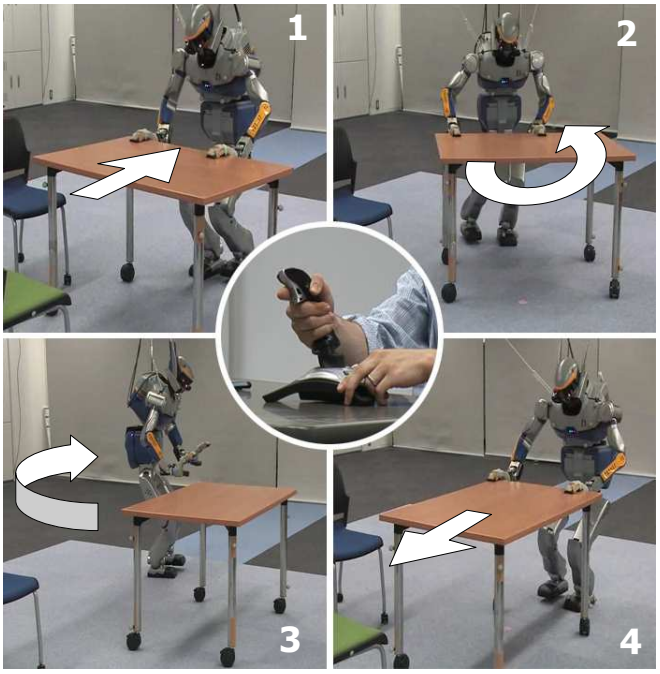
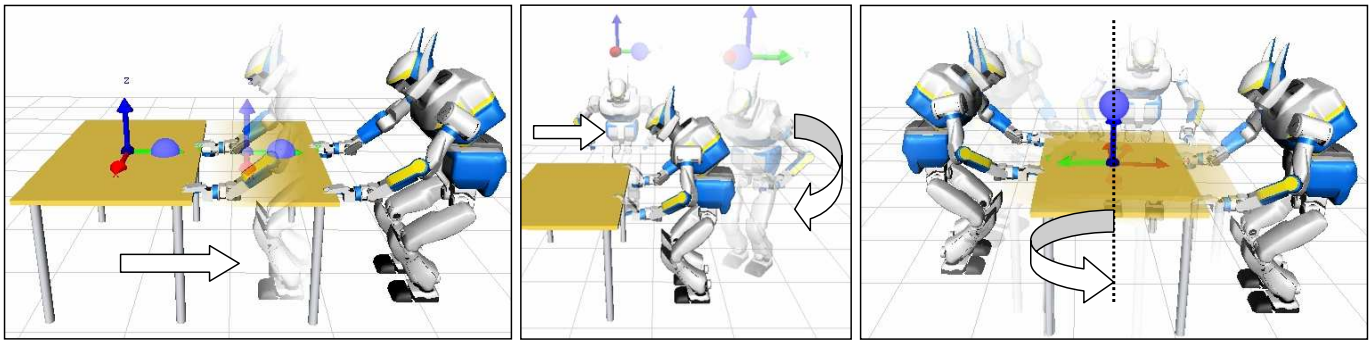


Fig. 1. HRP-2 successfully rotates the table by executing the operator's commands to pull it backwards, turn it, walk around, re-grasp and push.

II. Related Work

Significant telemanipulation research focuses on telepresence, or making the robot and communication transparent to the user [1], [2], [3], [4]. While transparency is particularly important for surgical applications [5], [6], Ambrose has also taken this strategy with Robonaut [7] and Hasunuma and Tachi applied it to HRP [8], [9]. These studies are complementary to our goals. We aim to isolate the operator from the complexity of dynamic trajectory generation, foot placement and force balance.

Existing work that maps low dimensional commands to complex humanoid trajectories typically controls the robot's hands in stance or its locomotion. In stance, Inoue [10] autonomously changes robot posture and steps to increase manipulability while Sian positions specific body points with automated balance compensation [11]. During locomotion a number of approaches modify heading online with varying levels of autonomy. For instance Nishiwaki presents fast trajectory generation for balanced walking gaits that can be used to teleoperate locomotion [12]. Yokoi applies a linear inverted pendulum for pattern generation [13]. Yasutaka[14] controls the hand frame in walking and Chestnutt modifies foot placements online to avoid obstacles [15]. To our knowledge no existing humanoid telemanipulation system



(a) Pulling the object (b) Walking (c) Turning the object

Fig. 2. Simulation and visualization of the task frame used in teleoperation. Control is performed in the robot frame (b) or the object frame (a,c).

handles large, heavy objects, allows walking and re-grasping or uses a single three-axis joystick to command all tasks.

Our implementation resolves dynamic constraints by extending preview control described by Kajita [16]. Our approach is distinct from earlier work by Harada [17], Takubo[18] and Nishiwaki[19] in that it not only handles pushing but arbitrary commanded displacements. To increase stability for distinct loads we also learn dynamic friction models of manipulated objects and apply them in control through force prediction. Learning about object dynamics is related to learning robot parameters. Atkeson, Moore and Schaal [20], [21], [22] summarize approaches to learning or adapting parameters to achieve precise trajectory following. Friction modeling, in particular has been studied extensively as summarized by Canudas [23] and Olsson[24].

III. Overview

Our teleoperation system consists of three modes of interaction: hand, object and robot centered. The operator chooses a mode by holding a button and uses a three-axis joystick to specify the target velocity for a reference pose or frame: $\mathbf{p}_H$, $\mathbf{p}_O$ or $\mathbf{p}_R$ respectively. $\mathbf{p}_O$ is fixed to the object and $\mathbf{p}_H$ to the robot hands. $\mathbf{p}_R$ is an abstract pose representing the robot. The object frame is an offset from $\mathbf{p}_R$ given by the operator or identified with external optical tracking in lieu of an on-board tracking system. The operator commands:

HANDS three dimensional translation of $\mathbf{p}_H$ with respect to $\mathbf{p}_R$ while the robot is standing.

OBJECT planar translation and rotation of $\mathbf{p}_O$ in absolute coordinates. Applied with the hands rigidly attached to the object.

ROBOT planar translation and rotation of $\mathbf{p}_R$ in absolute coordinates.

Figure 2 gives the operator display. The controlled frames, $\mathbf{p}_R$ in (b) and $\mathbf{p}_O$ in (a,c) are displayed with axes. The ball shows commanded velocity with rotation given by the z-axis.

In this paper, Section IV completes a simplified problem specification, allowing us to focus on the most significant challenge: Given that $\mathbf{p}_O$ is a rigid transformation of $\mathbf{p}_H$ after grasp: *Generate valid trajectories for all the robot joints such that $\mathbf{p}_O$ follows a commanded path*. Section VI gives hand and foot trajectories that satisfy kinematic constraints. Section VII address dynamic balance for a known force and Section VIII-A predicts a force from the manipulated object.

IV. Detailed Specification

This section completes our specification of the problem and introduces our assumptions. In the following, $\Delta t(150ms)$ is the time interval between joystick commands. $\mathbf{p}_A = \mathbf{T}\mathbf{p}_B$ indicate that a pose A is fixed with respect to B according to transform $\mathbf{T}$. The assumptions are not required to use our methods, however they aid in the interpretability of workspace descriptions and simplify the operator's task.

First, consider mode switching. All three modes are initiated with the robot at rest. HANDS can be terminated at any time since the robot legs remain static. Both walking modes impose a stopping delay since dynamic balance does not allow safe stops in mid-stride. To maximize preview control performance, ROBOT and OBJECT commands specify linear and angular velocities for two seconds, close to two steps. The robot comes to rest two seconds after the last command.

Despite the two second delay for stops, the overall walking control system has a latency of only $150ms$. At time t we are given two seconds of valid joint trajectories and a new joystick command. We use the existing trajectories to compute the robot state at $t + \Delta t$ and pose it as the start state for commanded velocities. The new command is interpreted by Section VI and Section VII during the Δt interval. New trajectories are implemented immediately after they are generated and validated as proposed by [12].

Second, for grasping arbitrary objects, teleoperation should permit the operator to move each hand independently. In this work we seek simple interaction and an interpretable workspace. Without loss of generality, we assume that both hands are fixed with respect to each other and $\mathbf{p}_H$ and that the orientation of the hands remains fixed with respect to $\mathbf{p}_R$. The hands are always opened at the start of HANDS mode and closed in a power grasp at the end. Grasp completion is determined by a threshold on finger strain-gauges. During HANDS, joystick position and rotation specify horizontal and vertical velocity respectively. To achieve this velocity we apply resolved rate control [25], mapping the workspace velocity of $\mathbf{p}_H$ to the robot joints. Target displacements are modified for compliance via impedance control (VIII-B).

Third, during walking we must relate $\mathbf{p}_R$ to a physical body. We do not use the robot torso since its motion maints dynamic balance in Section VII and should not be controlled by the operator. Instead, we assume the stance foot changes position at a constant rate t ($t = 900ms$). $\mathbf{p}_R$ is a point such that any time a foot ($F = lf$ or rf) contacts the ground:

$$\mathbf{p}_F^{kt} = \mathbf{T}_R^F \mathbf{p}_R^{kt} \tag{1}$$

Paths for $\mathbf{p}_R$ are continuous while footsteps are discrete. During the ROBOT mode, the operator specifies a path for $\mathbf{p}_R$. During OBJECT, we assume that $\mathbf{p}_R = \mathbf{T}_O^R \mathbf{p}_O$. Notice that ROBOT is a special case of OBJECT where $\mathbf{p}_R = \mathbf{p}_O$ and $\mathbf{p}_H$ is unconstrained. While we focus on the OBJECT mode, free walking is solved identically with fewer constraints. The remainder of this paper will focus on executing walking manipulation during OBJECT.

V. LAYERED TRAJECTORY GENERATION

Our system generates 38 valid joint trajectories that execute object motion and satisfy dynamic balance. To achieve this, we decouple the tasks and solve them sequentially. First we compute a workspace trajectory for the object, $\mathbf{p}_O$. Second, we find a trajectory for the torso, $\mathbf{p}_T$ that satisfies dynamic balance. Last, we map these trajectories to robot joints. For this approach to be feasible, we must first show that it is possible to independently compute trajectories for $\mathbf{p}_O$ and $\mathbf{p}_T$.

To independently position the object and the torso we specify the kinematic chains that place $\mathbf{p}_O$ and $\mathbf{p}_T$ from the stance foot. First, we define the positions of the hands and feet relative to the object. Due to rigid grasp manipulation, the hand positions, $\mathbf{p}_{lh}$ and $\mathbf{p}_{rh}$ are fixed to $\mathbf{p}_O$. Given $\mathbf{p}_R = \mathbf{T}_O^R \mathbf{p}_O$ and Eq. 1, the stance foot position is assigned at impact:

$$\mathbf{p}_{st} = \mathbf{T}_R^{st} \mathbf{T}_O^R \mathbf{p}_O \tag{2}$$

The robot swing foot, $\mathbf{p}_{sw}$ follows a cubic spline connecting its prior and future stance positions.

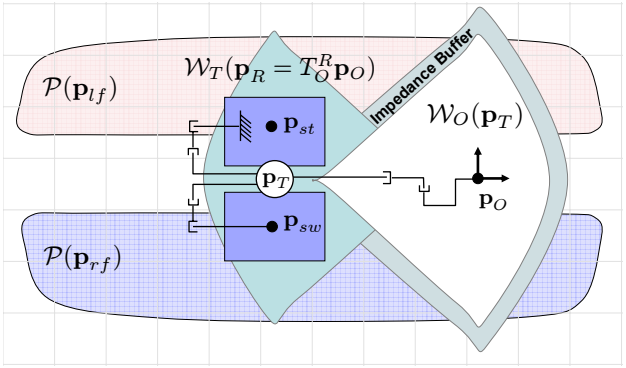
Given $\mathbf{p}_O$ and $\mathbf{p}_T$ we apply inverse kinematics to four kinematic chains to find all the robot joints.

$\mathbf{p}_{st}$	$\rightarrow$	$\mathbf{p}_R$	6 Stance leg	$\mathbf{p}_R$	$\rightarrow$	$\mathbf{p}_{lh}$	7 L arm
$\mathbf{p}_R$	$\rightarrow$	$\mathbf{p}_{sw}$	6 Swing leg	$\mathbf{p}_R$	$\rightarrow$	$\mathbf{p}_{rh}$	7 R arm

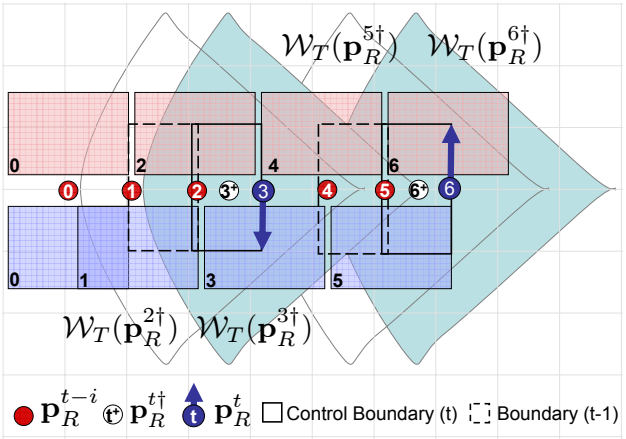
We use analytical IK solutions to expedite calculations and avoid drift. Chest joint values are constants that maximize the workspace. Redundancy in the arms is resolved by fixing elbow rotation about the line connecting the wrist and shoulder. [26] *Although many of the link positions are highly coupled, the two positions of interest, $\mathbf{p}_T$ and $\mathbf{p}_O$, are not.*

VI. OBJECT TRAJECTORY GENERATION

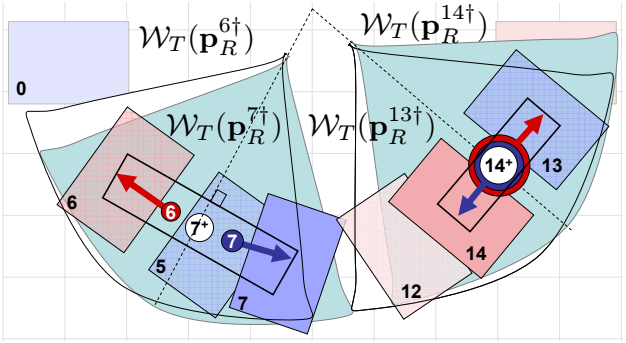
Since there are six independent joints separating $\mathbf{p}_{st}$ and $\mathbf{p}_T$ as well as $\mathbf{p}_T$ and $\mathbf{p}_O$ the robot has the necessary redundancy to independently control the object and its torso. Due to joint limits only a subset of ($\mathbf{p}_O$,$\mathbf{p}_T$) pairs are valid. Since we compute the $\mathbf{p}_O$ trajectory prior to $\mathbf{p}_T$ we must ensure that future torso trajectory generation will be valid with regard to kinematic constraints on robot hands and feet.



(a) Workspace for feet, object and torso.



(b) Valid trajectories for object translation.



(c) Valid trajectories for object rotation.

Fig. 3. Workspace constraints and valid trajectories during OBJECT control. In (b,c) two pairs of steps demonstrate the relationship between the torso workspace and the controller bounds.

The directional joystick command can be interpreted as a curve along which the object will travel. To find a trajectory we need to determine how far along this curve the object will be displaced. We summarize the motion constraints and give a method for setting the displacement.

A. Trajectory Constraints

Figure 3(a) illustrates two constraints that the object trajectory must satisfy: foot placement and torso workspace. We compute the trajectory for $\mathbf{p}_O$ in foot step increments. Suppose we have valid $\mathbf{p}_O^{t-1}$ and hence $\mathbf{p}_{st}^{t-1}$. We describe the constraints on a valid $\mathbf{p}_O^t$

Since the object trajectory uniquely determines foot placement we must ensure that $\mathbf{p}_{st}^t$ can be executed by the robot. Valid foot placements $\mathcal{P}(\mathbf{p}_{sw}^t)$ are determined with respect to $\mathbf{p}_{st}^{t-1}$. The region shown in Figure 3(a) was experimentally determined by previous work on walking control [12]. In addition to satisfying the bounds of stable walking it precludes foot collisions and satisfies joint limits.

When moving the stance foot from $\mathbf{p}_{st}^{t-1}$ to $\mathbf{p}_{st}^t$, the robot center of pressure must also switch to the new position. Section VII generates a torso trajectory that realizes this change. Although we have no analytical bounds for $\mathbf{p}_T$ generated by Preview Control[16], experimentally we have found that the torso does not leave the $\mathbf{p}_T$-oriented rectangle surrounding $\mathbf{p}_{st}^{t-1}$ and $\mathbf{p}_{st}^t$. Consequently, we ensure that this rectangle is included in the torso workspace.

$\mathcal{W}_O(\mathbf{p}_T)$ in Figure 3(a) is the workspace of the robot hands holding the object given a fixed torso. Conversely, we compute $\mathcal{W}_T(\mathbf{p}_R = T_O^R \mathbf{p}_O)$ as the workspace of the torso for a fixed object. Generally every point on the $\mathbf{p}_O$ trajectory should yield a workspace that includes the control rectangle. For translations and small rotations it is sufficient to ensure the condition at $\mathcal{W}_T(\mathbf{p}_R^{t-1})$ and $\mathcal{W}_T(\mathbf{p}_R^t)$. Since the object may deviate from the trajectory due to impedance control we reduce $\mathcal{W}_T(\mathbf{p}_R)$ by a safety margin.

B. Command Interpretation

We have now identified the criteria for a valid object displacement. The simplest strategy for choosing a distance along the commanded curve is to verify the displacement that results from the commanded velocity. If the displacement satisfies the constraints it is applied, otherwise the future object placement is unchanged. This strategy has merit for legged robots. Even if the robot cannot achieve the displacement with the current swing foot it might after changing support.

The disadvantage of this approach is that bringing the object to a stop makes manipulation less intuitive and introduces error due to static friction. We propose two improvements. First, since our representation of $\mathcal{P}(\mathbf{p}_{sw})^t$ is analytic, we solve for the maximum swing foot displacement. We find the maximum object displacement due to Eq. 2 and bound the target velocity. Second, we modify the object trajectory by incorporating one time step of command history.

Arrows in Figure 3(b) stem from $\mathbf{p}_R^t$ that correspond to operator specified $\mathbf{p}_O^t$. The foot $\mathbf{p}_{sw}^t$ is placed according to Eq. 1. However, when creating an object trajectory a history-adjusted $\mathbf{p}_O^{t\dagger}$ is placed at the midpoint of $\mathbf{p}_O^{t-1}$ and $\mathbf{p}_O^t$. This is reflected by $\mathbf{p}_R^{t\dagger}$ as shown in Figure 3(b,c). The workspace trajectories for the feet and the torso are generated from the original $\mathbf{p}_{sw}^t$ and modified $\mathbf{p}_O^{t\dagger}$ respectively.

Trajectory modification proved very effective in creating uninterrupted motion for the object even when the robot makes no progress with a single step. Figure 3(c) shows two cases that occur during rotation. Odd steps, 7, move the right foot sideways and even steps, 14, place the left foot adjacent to the right. The object pose, $\mathbf{p}_O^{t\dagger}$, moves in equal increments on each step leading to a constant velocity object trajectory. Furthermore, as seen in Figure 3(a,b), our approach centers $\mathcal{W}_T(\mathbf{p}_R^{t\dagger})$ around the rectangular controller bounds. This allows the robot to take longer steps safely.

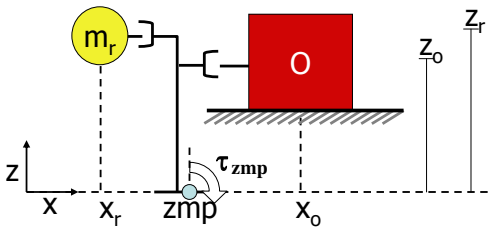


Fig. 4. Model of the robot used for torso trajectory generation.

VII. Torso Trajectory Generation

The object trajectory in Section VI specifies foot placements and allows for bounded torso motion around $\mathbf{p}_R$. We now compute a trajectory for the torso, $\mathbf{p}_T$, that balances the robot during manipulation. Reaction forces from the object are included in our controller. Section VIII-A provides two methods identifying these forces.

We define balance by relating the zero moment point to stance foot position. The robot is modeled as a point mass, m_r, at a fixed displacement from the torso, $\mathbf{p}_r = \mathbf{T}_T^r \mathbf{p}_T$. $\mathbf{p}_r = \{x_r, y_r, z_r\}$, z_o is the height of the hands and f is the reflected force. This section computes a trajectory for x_r and therefore x_T. y_r is found analogously and z_r is constant.

Eq. 3 introduces zmp as the ground point around which the torques due to gravity acting on m_r, reflected force from accelerating m_r and the force from the object sum to zero.

$$\tau_{zmp} = m_r g(x_r - zmp) - m_r \ddot{x}_r z_r - z_o f = 0 \qquad (3)$$

Solving for zmp yields:

$$zmp = x_r - \ddot{x}_r \frac{z_r}{g} - \frac{z_o f}{m_r g}. \qquad (4)$$

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 $zmp_d = x_{st}$ as given by the object trajectory.

Let $J_0 = \sum_t (zmp_d^t - zmp^t)^2$ be the performance index for balance. Eq. 5 further defines β and β_d as functions of zmp_d and x_r respectively.

$$\beta_d = zmp_d + \frac{z_o f}{m_r g} \quad \beta = x_r - \ddot{x}_r \frac{z_r}{g} \qquad (5)$$

Substitution yields $J_0 = \sum_t (\beta_d^t - \beta^t)^2$. β_d^t is fully determined since zmp_d is the trajectory of foot centers, $\{z_o, m_r, g\}$ are constant and f is assumed to be known.

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

$$J = \sum_{t=1}^{\infty} Q_e(\beta^t - \beta_d^t)^2 + R(\dddot{x}^t - \dddot{x}^{t-1})^2 \qquad (6)$$

We determine the optimal $\dddot{x}_r$ with preview control [16]. 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 $\Delta\dddot{x}_r^t$ in Eq. 7 minimizes J.

$$\Delta\dddot{x}_r^t = -G_1 e(t) - G_2 \Delta x_r^t - \sum_{i=1}^{N} G_3^i(\beta_d^{t+i} - \beta_d^{t+i-1}) \quad (7)$$

More is available in Appendix B of [27] summarizing [28]. The control $\Delta\dddot{x}_r$ is discretely integrated to generate the trajectory $\{\ddot{x}_r, \dot{x}_r$ and $x_r\}$. Since $\mathbf{p}_r$ 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.

VIII. Further Considerations

Having described the generation of teleoperation trajectories we now introduce force modeling and compliance modules that led to the experimental success of our system.

A. Force Modeling

Section VII described torso trajectory generation given a known external force. When the humanoid interacts with an unspecified object, the reflected forces may not be known in advance. We compared a reactive strategy for handling external forces to one that modeled the object dynamics with viscous friction in [29]. The reactive strategy assumed that an experienced force would remain constant for .15 seconds during trajectory execution. [19]

For modeling, we accumulated a small set of data using the reactive strategy and applied least squares estimation to compute a friction coefficient, c, that satisfies Eq. 8.

$$\begin{bmatrix} \dot{x}^1 & \dot{x}^2 & \cdots & \dot{x}^n \\ 1 & 1 & \cdots & 1 \end{bmatrix}^T \begin{bmatrix} c \\ b \end{bmatrix} = \begin{bmatrix} f^1 & f^2 & \cdots & f^n \end{bmatrix}^T \quad (8)$$

The term b was used to remove bias. [29] showed that this method reduces variance in sensed forces, improves prediction accuracy and system stability. We applied modeling to predict forces tangent to the direction of motion. The robot was prevented from pulling the object laterally except during rotation. Un-modeled lateral forces were handled reactively.

B. Online Feedback

We have described the generation of a balanced trajectory for $\mathbf{p}_R$ given $\mathbf{p}_O$. To handle online errors we modify these trajectories online prior to realization with robot joints. Online feedback operates at a $1ms$ cycle.

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

The trajectory for $\mathbf{p}_O$, or the robot hands, is modified by impedance. We use compute an offset for $\mathbf{p}_O$ that results from integrating the measured force error F in Eq. 9.

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

Impedance ensures that hand positioning errors do not lead to large forces pushing down on the object. d_i and k_i are set low for vertical displacements. Impedance also prevents the robot from exceeding torque limits when the trajectory cannot be executed due to un-modeled dynamics. The horizontal position gain trades a displacement of $10cm$ for $100N$ steady state force. This allows for precise trajectory following and soft termination when the offset exceeds force limits.

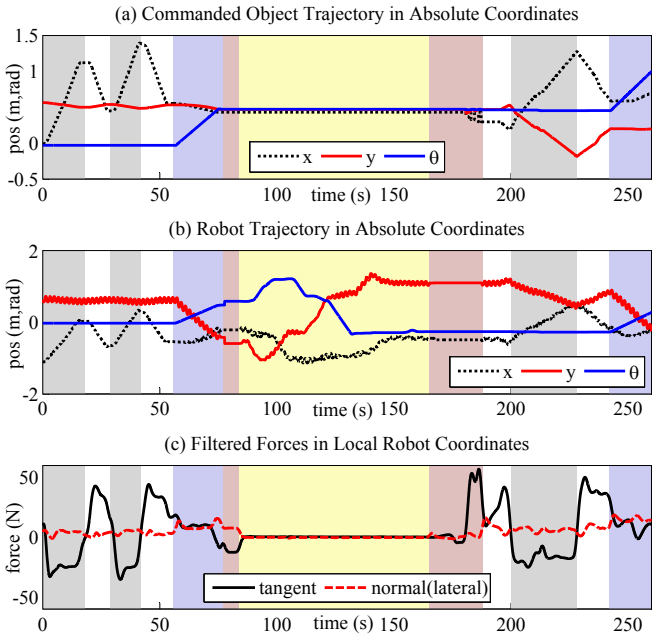


Fig. 5. Trajectory traces of telemanipulation. The robot was driven to manipulate, walk, re-grasp and repeat for 4.5 minutes as shown in Figure 6.

IX. Results

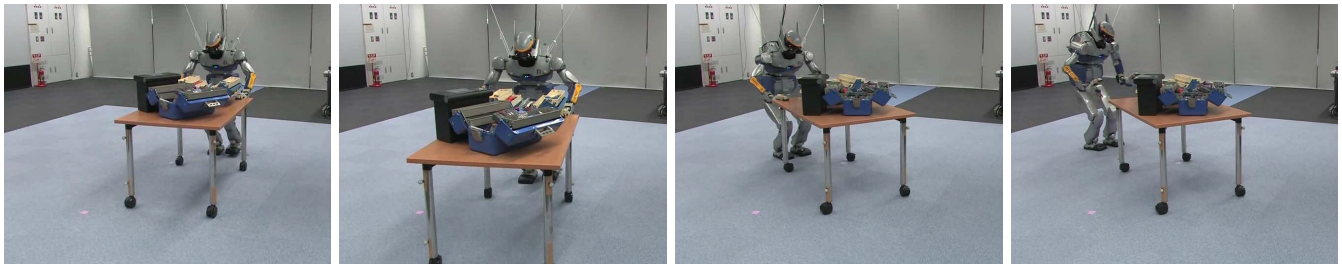
We applied our approach to a number of teleoperation tasks that required the robot to transport a table on casters with varying loads. Object models were constructed during the initial interaction with a new load. The system performed repeatable telemanipulation for ten minute intervals.

Figures 5 and 6 show traces and images of a representative four minute execution with a load of 55kg. The load is equal to the mass of the robot. In this experiment the robot pushes and pulls the table twice, then rotates it. It is then commanded to release the object, walk around, grasp it from the side and repeat pulling, pushing and rotation.

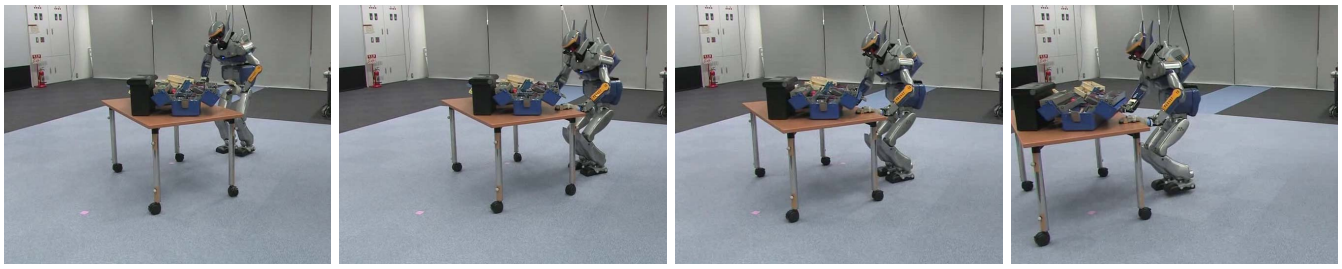
In Figure 5 the object is static during walking and the robot is standing during grasping. Observe the correspondence between experienced force and commanded object velocity. Forces during rotation about the object center are significantly smaller. The large force after the second grasp is due to commanded pulling while standing.

X. Discussion

In this paper we presented a novel approach to telemanipulation for humanoid robots using a single three axis joystick. Our flexible task frame strategy and simple interface allow a novice user to easily interact with the system. Using a layered architecture, we guarantee at each layer that future layers will find successful trajectories. In conjunction with friction-based object modeling our method successfully completed telemanipulation tasks with objects on casters and loads of up to 55kg. Multiple experiments lasted up to 10 minutes and included grasping, pulling, pushing and re-grasping.



(a) 29s Stand (b) 42s Push (c) 77s Pull, Rotate (d) 95s Release



(e) 130s Walk (f) 165s Grasp (g) 210s Pull, Push (h) 255s Rotate, Push

Fig. 6. Images from a single continuous execution of teleoperation of a 55kg table with pushing, pulling and re-grasping

Creating a low dimensional control space has further applications for continued work in robot autonomy. Advances in environment sensing will make it possible for the robot to judge the utility of actions and plan object manipulation. A low dimensional space of safe controls increases the efficiency of planning autonomous behavior for the robot.

REFERENCES

[1] G. Niemeyer and J.J.E. Slotine. Telemanipulation with Time Delays. *The International Journal of Robotics Research*, 23(9):873, 2004.

[2] T. Itoh, K. Kosuge, and T. Fukuda. Human-machine cooperative telemanipulation with motion and forcescaling using task-oriented virtual tool dynamics. *Trans. Robotics and Automation*, 16(5):505–516, 2000.

[3] JE Colgate. Robust impedance shaping telemanipulation. *Robotics and Automation, IEEE Transactions on*, 9(4):374–384, 1993.

[4] WB Griffin, WR Provancher, and MR Cutkosky. Feedback strategies for shared control in dexterous telemanipulation. *Intelligent Robots and Systems, 2003.(IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on*, 3, 2003.

[5] SE Butner and M. Ghodoussi. Transforming a surgical robot for human telesurgery. *Robotics and Automation, IEEE Transactions on*, 19(5):818–824, 2003.

[6] G.S. Guthart and J.K. Salisbury. The Intuitive TM Telesurgery System: Overview and Application. *IEEE ICRA, San Francisco, CA, April*, 2000.

[7] RO Ambrose, H. Aldridge, RS Askew, RR Burridge, W. Bluethmann, M. Diftler, C. Lovchik, D. Magruder, F. Rehnmark, N.J.S. Center, et al. Robonaut: NASA's space humanoid. *Intelligent Systems and Their Applications, IEEE*, 15(4):57–63, 2000.

[8] H. Hasunuma, K. Nakashima, M. Kobayashi, F. Mifune, Y. Yanagihara, T. Ueno, K. Ohya, K. Yokoi, S.T.D. Center, K.H.I. Ltd, et al. A tele-operated humanoid robot drives a backhoe. *IEEE Int. Conf. on Robotics and Automation, 2003. Proceedings. ICRA'03.*, 3, 2003.

[9] S. Tachi, K. Komoriya, K. Sawada, T. Nishiyama, T. Itoko, M. Kobayashi, and K. Inoue. Telexistence cockpit for humanoid robot control. *Advanced Robotics*, 17(3):199–217, 2003.

[10] K. Inoue, H. Yoshida, T. Arai, and Y. Mae. Mobile manipulation of humanoids: Real-time control based on manipulability and stabilty. In *IEEE Int. Conf. Robotics and Automation*, pages 2217–2222, 2000.

[11] N. Sian, K. Yokoi, S. Kajita, and K. Tanie. Whole body teleoperation of a humanoid robot integrating operators intention and robots autonomy an experimental verification. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS'03)*, pages 1651–1656, 2003.

[12] K. Nishiwaki, S. Kagami, Y. Kuniyoshi, M. Inaba, and H. Inoue. Online generation of humanoid walking motion based on a fast generation method of motion pattern that follows desired ZMP. *Intelligent Robots and System, 2002. IEEE/RSJ International Conference on*, 3, 2002.

[13] K. Yokoi, F. Kanehiro, K. Kaneko, K. Fujiwara, S. Kajita, and H. Hirukawa. A Honda Humanoid Robot Controlled by AIST Software. *Proc. of the IEEE-RAS International Conference on Humanoid Robots*, pages 259–264, 2001.

[14] Y. Fukumoto, K. Nishiwaki, M. Inaba, and H. Inoue. Hand-centered whole-body motion control for a humanoid robot. In *IEEE Int. Conf. on Intelligent Robots and Systems*, pages 1186–1191, 2004.

[15] J. Chestnutt, P. Michel, K. Nishiwaki, J. Kuffner, and S. Kagami. An intelligent joystick for biped control. In *Proceedings of the IEEE International Conference on Robotics and Automation*, May 2006.

[16] Shuuji Kajita and 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.

[17] 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.

[18] 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.

[19] 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.

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

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

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

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

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

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

[26] D. Tolani, A. Goswami, and N.I. Badler. Real-time inverse kinematics techniques for anthropomorphic limbs. *Graphical Models*, 62(5):353–388, 2000.

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

[28] 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.

[29] M. Stilman, K. Nishiwaki, and S. Kagami. Learning object models for whole body manipulation. In *Int. Conf. on Humanoid Robotics (Humanoids'07)*, 2007.