A Sensor-Based Dual-Arm Tele-Robotic System

Daniel Kruse, *Student Member, IEEE,* John T. Wen, *Fellow, IEEE,* and Richard J. Radke, *Senior Member, IEEE*

***Abstract*—We present a novel system to achieve coordinated task-based control on a dual-arm industrial robot for the general tasks of visual servoing and bimanual hybrid motion/force control. The industrial robot, consisting of a rotating torso and two seven degree-of-freedom arms, performs autonomous vision-based target alignment of both arms with the aid of fiducial markers, two-handed grasping and force control, and robust object manipulation in a tele-robotic framework. The operator uses hand motions to command the desired position for the object via Microsoft Kinect while the autonomous force controller maintains a stable grasp. Gestures detected by the Kinect are also used to dictate different operation modes. We demonstrate the effectiveness of our approach using a variety of common objects with different sizes, shapes, weights, and surface compliances.**

***Note to Practitioners*—Industrial robots traditionally are pre-programmed with teach pendants to perform simple repetitive tasks without any sensor feedback. This work was motivated by demonstrating that industrial robots can also perform advanced, sensor-based tasks such as visual servoing, force-feedback control, and tele-operation. Industrial robots are typically limited by the long delay between command and action, but with careful tuning, we show that these sensor-based methods are still feasible even with off-the-shelf sensors.**

The specific experimental testbed involves a 15 degree-of-freedom dual-arm industrial robot with each wrist outfitted with a camera, a rubber contact pad, and a force/torque sensor. A Microsoft Kinect is used to communicate operator commands through gesture. The integrated system involves seven processes running on three computers (2 running Windows 7, 1 running Windows XP) connected through a local hub using the TCP/IP protocol. The communication between the components is based on an object-oriented distributed control and communication software architecture called Robot Raconteur.

Though the implementation is for our specific testbed, the approach is sufficiently general to be extended to other robots, end effectors, sensors, and operating systems.

Keywords—Primary: Dual-arm manipulation, tele-robotics, human interface, industrial robot
Secondary: Visual servoing, binary tag, force control, distributed architecture

I. Introduction

The most prevalent use of robots today involves industrial robots in manufacturing lines. These robots are programmed through teach pendants to traverse through a pre-taught set of points to execute repetitive tasks [1]. Industrial robots allow limited feedback from sensors, such as vision or force/torque sensors, through command trajectory modification (e.g., *external alter* in the VAL robot programming language [2]), but they are not designed for human interaction. Even when external sensors are used, they are tailored for specific tasks, e.g., welding [3], filament winding [4], grinding [5], or drilling [6], and tied to specific platforms, e.g., VAL [7] or RAPID [8]. These systems typically involve a single robot arm equipped with an end effector dedicated to a specific class of tasks.

Multi-arm industrial robots and tele-robots are not common due to their mechanical and system level complexity. When multiple arms jointly hold a load, in addition to the motion of the load, the internal force within the load needs to be regulated for stable grasping while avoiding damaging the part. In the case of force-reflecting teleoperation, synchronization and stability issues are even more severe, since the human operator needs to regulate both the force of interaction between the load and the environment and the internal squeeze force in the load.

In this paper, we present a novel tele-robotic framework for human-directed dual-arm manipulation. The human operator provides gestural commands and motion directives while the control system autonomously locates the object of interest and maintains grasp force closure. Our approach is sensor-based, allowing flexibility in task specification and execution. We consider robots with multiple kinematically redundant arms. Such robots can tackle a much broader range of tasks than a single arm, but at the same time incur increased complexity in terms of potential collision as well as force of interaction in collaborative tasks. These dual-arm and humanoid-style robots have become especially interesting in recent years due to the ongoing DARPA Robotics Challenge [9], intended to develop highly sophisticated robotic response systems for extreme emergencies.

For the specific implementation and demonstration of our approach, we use a 15-degree-of-freedom (dof) dual-arm industrial robot (Motoman SDA10) and a suite of peripheral sensors (cameras, Kinect, and force/torque sensors) distributed over multiple computers. We integrate these subsystems together in a distributed system using *Robot Raconteur* (RR) [10], the object-oriented distributed control and communication software system developed in our laboratory, and now available for download at `robotraconteur.com`. We choose RR over other distributed robotic middleware systems such as ROS Industrial [11] due to its multi-platform compatibility (including MATLAB for data analysis, visualization, and user interface), true distributed implementation (no master node), object-oriented philosophy (flexible data structures), and ease of use (automatically generated client/service object interface code).

We use an example scenario to guide the development. An unknown object is placed arbitrarily in the robot workspace. The robot must detect the location of the object, align both end effectors for a grasp using visual servoing, actually grasp the

All three authors are with the Department of Electrical, Computer, and Systems Engineering, Rensselaer Polytechnic Institute, Troy, NY 12180, USA. J. Wen is also with the Department of Industrial and Systems Engineering. e-mail: `krused2@rpi.edu, rjradke@ecse.rpi.edu, wenj@rpi.edu`

object, and then allow a human operator to command the robot using a Kinect. The objects included in this study, all handled with the same controller, include a light cardboard box, a plastic tub, a flexible rubber ball, a large plastic bucket, and a highly rigid plastic cooler. These five objects differ in size, weight, shape, surface, and compliance. Some experiments also involve placing a sliding weight inside the plastic tub or plastic bucket. We place the objects in arbitrary poses inside the robot workspace, and the robot is to locate the object, determine the grasp location, stably grasp the object, and then manipulate the object either through a pre-programmed commanded signal or the operator's gestures.

The successful execution of this task scenario requires the solution of multiple technical issues. A planning-based approach would be general and guarantee finding a solution if one exists, but the solution would be computationally intensive and require detailed geometric and other model information of the system and its environment. Instead, we choose a sensor-driven reactive approach to allow simple but robust solutions. Our goal is not to generate a complete autonomous solution but to assist the human operator to find a feasible solution. Our system consists of the following key components (described in detail in Section V):

- *Robot motion control.* Typical industrial controllers have closed proprietary torque-level control and only allow modification of the joint trajectory at a slower rate. Since we are interested in the task space motion, we use the damped least-squares algorithm to modify the joint-level motion based on task space motion requirements. We exploit kinematic redundancy in our system to avoid collision and enhance manipulability and grasp stability through the use of potential fields [12].
- *Object identification and location.* The robot needs to identify and locate the object, and determine good places for grasping. We draw on planar tagging schemes developed in the artificial reality community, in particular the VTT ALVAR tags [13], to locate the object and determine suitable grasp locations.
- *Force control.* To securely hold the object with the robot arms, the robot must apply a squeeze to result in enough friction to prevent the object from slipping out of grasp. We use the well-known integral force control, with a force-dependent gain to enhance robustness [14] and show closed loop stability for a compliant object.
- *Load compensation.* To allow large excursion in load orientation, the weight and center of gravity of the load are estimated and incorporated into the force control. We show that the system remains stable even under estimation errors and time-varying loads.
- *Human Interface.* We use the Microsoft Kinect sensor to interpret the user's gestures and convert them into desired poses for the task at hand. This is versatile, robust, and more natural for the user since it is not limited by the motion range limitation or mechanical impedance of a mechanical interface. The user only directs the task sequencing and object motion; the control system ensures a stable grasp autonomously.

We envision that the human-directed tele-robotic system described in this paper, which facilitates integration, prototyping, and human interaction, will serve as the foundation for more complex human-robot collaborative sensing and manipulation in the future. This paper is an extension of a conference paper [15] that focused on the Kinect gestural interface.

II. Problem Statement

The goal of this work is to develop a robust and versatile tele-robotic system for dual-arm manipulation of a held object. The placement, dimensions, weight and mass distribution of the object (which may be time-varying) are unknown prior to manipulation, and all necessary parameters are estimated on-line based on sensor measurements.

Consider the dual-arm grasping problem represented by Fig. 1. The system is to locate a relatively large object within the robot workspace, grasp it at two designated, near-parallel contact points, and then manipulate the object according to a reference signal provided either through tele-operation from a human operator or a pre-defined sequence of poses.

We define the Euclidean coordinate frame of a rigid body as a reference point, $\mathcal{O}$, and a right-handed orthonormal frame, $\mathcal{E}$. The relative configuration between two frames, say **b** relative to **a**, is given by the homogeneous transform $H_{ab} = \begin{bmatrix} R_{ab} & p_{ab} \\ 0 & 1 \end{bmatrix}$ where R_{ab} is $\mathcal{E}_b$ represented in $\mathcal{E}_a$, and p_{ab} is the vector from $\mathcal{O}_a$ to $\mathcal{O}_b$, represented in $\mathcal{E}_a$. The world and task (object) frames are denoted by the subscripts O and T, respectively. The camera and end effector frames are denoted by the subscripts $C_{L/R}$, $E_{L/R}$, respectively, with L and R specifying the left or right arm.

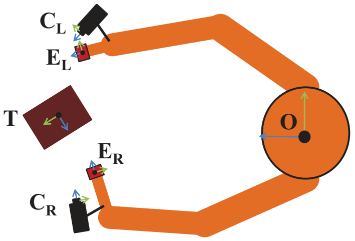


Fig. 1: The different frames of reference for the inertial frame O and task frame T along with the end effectors E and cameras C on each arm.

Location of the target object combines the camera-based measurement of the task frame, H_{CT}, and camera location in the world frame, H_{OC}, to determine the object location, H_{OT}. Visual servoing uses H_{OT} to generate target end effector positions and orientations, and to control the robot joints to drive each H_{OE} to the desired goal. Since the arms are coupled at the rotating torso, the kinematics of the two arms are not independent – the entire robot must be considered for the alignment of the two contacts.

We expect the target load to be relatively large and unable to be held between the fingers of a traditional gripper. The force control function commands the two end effectors to apply a squeezing force for a sufficiently secure grasp, both at a stationary pose and during manipulation. In our implementation, the operator provides input using the Kinect; such a non-contact gesture-based interface is particularly attractive since the user is unfettered by mechanical constraints.

III. Related Work

This paper primarily addresses coordinated, task-based control methods for a dual-arm industrial robot. Since there is a large body of work on single-arm robotics, we narrow our scope of related work to only dual- or multi-arm robots. A recent survey listing many of the current platforms with two or more arms can be found in [16].

Our visual servoing algorithm is based on ALVAR tags [13] used in the artificial reality community. Visual servoing in itself has been heavily researched in detail and a thorough introduction to the subject was presented in [17], [18]. When applied to multi-arm systems, the arms are usually treated independently; however this system has a movable waist that couples the motion of the two arms. Therefore, the visual servoing algorithm must consider the entire robotic system at once. Burget et al. [19] and Dietrich et al. [20] both developed methods for considering the entire body during manipulation; however their algorithms deal with pre-planning motions.

Several papers have presented ways of combining motion and force control for multi-arm platforms. The typical solutions are through *hybrid position/force control*, where the position and force control loops are decoupled and treated independently, or through *impedance control*, which has the general goal of obtaining a desired dynamic interaction between the robot and object or environment [21]. We apply the hybrid position/force control method [22] to directly impart a squeeze force. Many position/force control methods are based on robot dynamic models and motor torque control [23], [24]. Since direct low-level robot control is not possible for most industrial robots, we use the kinematics-based position accommodation scheme [25].

For autonomous operations, tasks for multi-arm robots are often pre-solved using planning algorithms with known geometry information [26]–[30] in which the motion path is computed off-line and then performed in open-loop. There are also autonomous grasp planners such as *GraspIt!* and *OpenGRASP*, but they only address rigid objects. Sensor-based motion planning is possible, but due to the computational complexity it is usually reserved for relatively simple systems [31].

Other groups have looked into using the Microsoft Kinect to control robots as well, as seen in [32]–[34]. These works took advantage of the RGB-d sensor to control robotic hands, while we address a full multi-armed robotic system.

IV. System Description

Our platform centers around the Yaskawa Motoman SDA10 robot, a dual-arm 15-dof industrial robot with two 7-dof arms and a 1-dof torso, shown in Fig. 2. The robot has a built-in joint controller and allows external interface through Motoman's High Speed Controller (HSC). The HSC interface provides joint angle read and incremental commanded joint angle write at a 2ms rate, with a significant delay (more than 100ms) due to internal trajectory generation. The low-level robot control system takes a commanded correction to the current joint angle and calculates a trapezoidal trajectory profile for the joint motion. The resulting motion may be modeled sufficiently closely (except at transitions between motion segments) as a first-order-plus-dead-time system, as shown in Fig. 3. Each joint can travel up to $170°/s$. We artificially restrict the joint speed to $20°/s$ for safer operations.

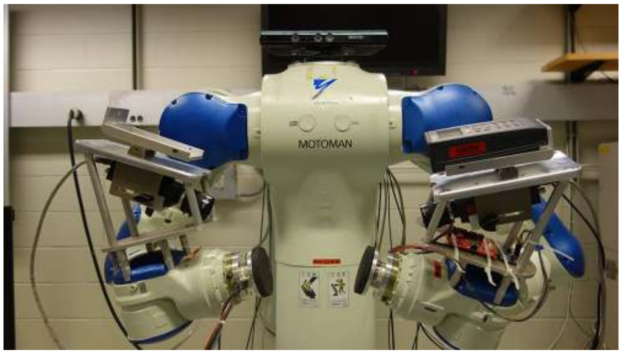


Fig. 2: The industrial robot and its peripherals.

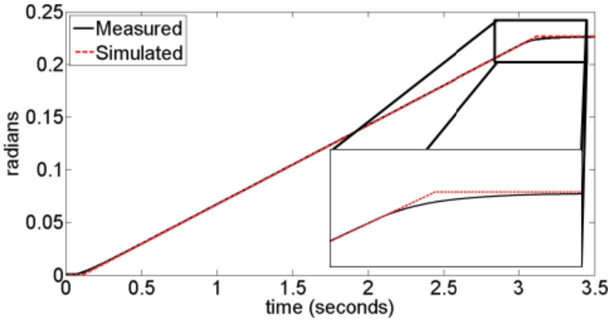


Fig. 3: The simulated and actual response of the last wrist joint of the left robot arm, commanded with a constant input (joint increment) for 3 seconds. A simple first-order-plus-dead-time plant approximation is used for the simulated response.

For visual feedback, a Sony XCD-X710CR camera is mounted on the side of each wrist and angled in such a way that the end effector is in the field of view. Directly at the end of each arm is an ATI Mini45 force/torque transducer, with a rubber disc mounted at the end for a higher-friction contact.

The components of the overall system are coordinated using our home-grown distributed control system, called Robot Raconteur [10]. The system architecture is illustrated in Fig. 4. Separate Robot Raconteur services are written (in C#) for the Motoman HSC interface, ATI force/torque sensor, cameras, image processing, and Kinect interface, residing on three separate computers linked together in a local area network via Ethernet and TCP/IP. The overall coordination is conducted by

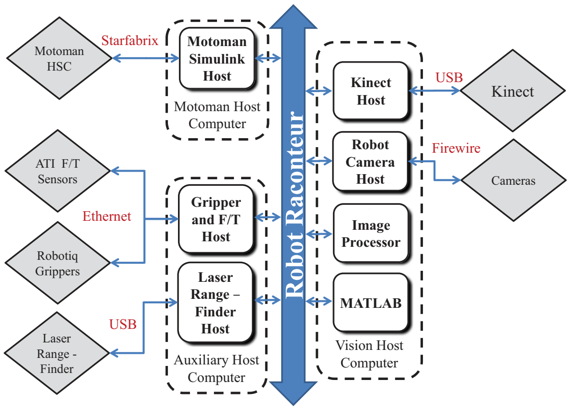


Fig. 4: Overall distributed control system architecture using the RobotRaconteur middleware.

a MATLAB script that connects to these services as a client. This MATLAB environment does not run in real time and has no explicit timing guarantee.

V. Task-Space-Based Motion Control

The input to the Motoman HSC is a vector of 15 joint corrections that then passes through a trajectory generator. With tight low-level motor servo loops, we can ignore the dynamics and reasonably approximate the robot response with a first-order-plus-dead-time plant (the dead-time is configuration dependent, varying between 100-150ms). Because the sensor provides task space information, we apply a task space kinematic control law. Let J be the 12×15 Jacobian matrix that maps the 15 joint velocities to the 12 spatial velocities (angular and linear) of both end effectors:

$$\begin{bmatrix} \nu_{E_L} \\ \nu_{E_R} \end{bmatrix} = J\dot{q} \tag{1}$$

where each spatial velocity $\nu_{E_{L/R}} = \left[\omega_{E_{L/R}}, v_{E_{L/R}}\right]^\top$ contains the angular and linear velocities of the left or right end effectors, and q is the stacked vector of q_T, q_L, and q_R, the torso, left arm, and right arm joint angles. The coordinate-free form for the ith column of J, J_i, is given by

$$J_i = \begin{cases} \begin{bmatrix} \vec{h}_i & \vec{h}_i \times \vec{p}_{iE_L} & \vec{h}_i & \vec{h}_i \times \vec{p}_{iE_R} \end{bmatrix}^\top & i = 1 \\ \begin{bmatrix} \vec{h}_i & \vec{h}_i \times \vec{p}_{iE_L} & 0 & 0 \end{bmatrix}^\top & i = 2, \ldots, 8 \\ \begin{bmatrix} 0 & 0 & \vec{h}_i & \vec{h}_i \times \vec{p}_{iE_R} \end{bmatrix}^\top & i = 9, \ldots, 15 \end{cases}$$

where the axes of revolution, $\vec{h}_i$, are shown in Fig. 5. Joint 1 is the torso, joints 2-8 correspond to the left arm, and joints 9-15 to the right arm. For computation, all vectors in J_i are represented in the base frame. Since we use the industrial controller in the inner loop, we consider $\dot{q}$ as the command input (to be sent to the HSC).

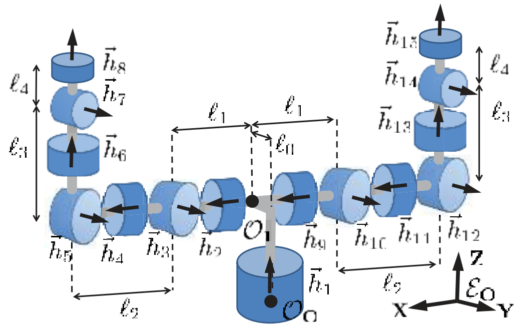


Fig. 5: The 15 axes of revolution for the Motoman robot.

Denote the desired position and pose of the end effector as (p^d_{OE}, R^d_{OE}). There are numerous possible error functions on the rotation group [35]. For simplicity, we choose a minimal representation, but other choices may be used as well. Let the position and orientation errors be

$$e_p = p_{OE} - p^d_{OE}, \;\; e_R = \sigma\left(R^d_{OE} R^\top_{OE}\right),$$

where σ is a 3-parameter representation of rotation. We choose σ to be the vector quaternion, but any other representation may be used. Let J_σ be the representation Jacobian, i.e., $\dot{\sigma} = J_\sigma \omega$. We define the mapping from spatial velocities ν to the new representation of spatial velocities, $\nu_\sigma = [\dot{\sigma}, v]^\top$, as

$$V = \text{diag}(J_{\sigma_L}, I, J_{\sigma_R}, I). \tag{2}$$

such that $\nu_{\sigma E} = V\nu_E$ and ν_E is the stacked vector of desired end effector spatial velocities $[\nu_{E_L}, \nu_{E_R}]^\top$. We can therefore adjust our original Jacobian in (1) as $J_a = VJ$.

The following damped least-squares task space controller [36] will now map the error velocities of the end effectors to joint velocities:

$$\dot{q} = -\underbrace{J_a^\top (J_a J_a^\top + \beta I)^{-1}}_{:= J_a^\dagger} Ke \tag{3}$$

where e is the stacked error vector $(e_{R_L}, e_{p_L}, e_{R_R}, e_{p_R})$, K is the feedback gain matrix, and $J_a^\dagger$ is the approximate pseudo-inverse with βI added to avoid singularity. (The singularity structure is completely characterized in Appendix A.) Note that a choice of specific units for linear and angular velocities implicitly means that gain terms for rotational error will likely be higher than the gains used for position error.

By using the Lyapunov function $V(e) = \|e\|^2$, it is easy to show that the position and orientation error will converge if we ignore feedback delay. If the steady state configuration is not singular (including both arm and representation singularities), the error will converge to zero. In the presence of delay, the gain will need to be sufficiently small with respect to the delay, on average, to guarantee stability.

For collision and joint limit avoidance, we use the standard artificial potential field approach [12]. For a task space constraint on $p_{OE} \in \mathcal{P}$, where $\mathcal{P}$ is the feasible region for the end effector position, we construct a smooth external penalty function, $\rho_p(p_{OE})$, that is zero in the feasible region and positive in the infeasible region with no local minima. For a joint space constraint on $q \in \mathcal{Q}$, where $\mathcal{Q}$ is the feasible region for the joint angles, we similarly construct an external penalty function, $\rho_q(q)$. The task space controller may now be modified as:

$$\begin{aligned} \Psi(p_{OG}, q) &= J_a^\dagger \left(\gamma_p \rho_p (\nabla \rho_p)^\top + \gamma_q \left(J_a^\top \right)^\dagger \rho_q (\nabla \rho_q)^\top \right) \\ \dot{q} &= -J_a^\dagger K e - \Psi(p_{OG}, q) \end{aligned} \tag{4}$$

where γ_p and γ_q are weightings for the repelling potential functions. This controller drives the end effectors to the desired locations while avoiding undesired configurations outside of the feasible regions $\mathcal{P}$ or $\mathcal{Q}$.

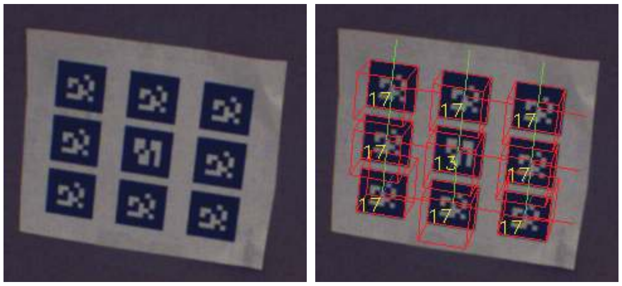
It is well known that the potential field approach may create local minima and that the robot could get stuck. In human-directed motion, the operator could modify the motion command to force the robot out of the local minimum. In autonomous motion, replanning or modification of the potential field may be necessary. Though we are aware of the possibility, this has not been an issue in our experiments.

A. Visual Servoing

There is a long history and a large body of literature on visual servoing [17], [18]. However, feature extraction under different lighting conditions, surface geometries, and textures is time-consuming and error prone. We choose a simple, yet robust, shortcut to object detection and location, by marking the grasping point with binary tags. Fiducial markers have long been used to aid object identification and location in machine vision [37]–[39]

Many such tags have been proposed [38], [40]; we choose the 2D ALVAR tags [13] as shown in Fig. 6 since the detection library is readily available. The ALVAR library determines the pose of each tag by mapping the homography between the known position of points in the tag frame to the measured pixels in the image frame. It is straightforward to use the estimated homography to recover the homogeneous transform H_{CG} given an image of a planar tag [41]. However, it is well documented [42] that a reflection ambiguity about the plane perpendicular to the optical axis creates two local minima for the orientation. To resolve this ambiguity, we include a set of "support tags" around the target tag and ascertain the orientation by majority poll. This has proven much more reliable than trying to resolve the pose ambiguity of a single tag. Once we have an estimated position and orientation of the target, we can use the task space controller in (4) to drive the end effectors to the corresponding targets.

We define infeasible regions for the end effectors as being too close to the object, too close to the body of the robot (to prevent self-collision), and poses that do not include the tag in the camera field of view. We construct virtual walls at the boundary of each of these regions, defined by their center point p_{OC} and inward normal n. Each wall has a corresponding potential function, for use in (11), determined by the projection of $p_{OG} - p_{OC}$ onto n. If the projection is negative, then ρ_p is zero, otherwise, ρ_p is increasing. The controller (4) may now be applied to drive the end effectors to their targets while avoiding the set of task space obstacles and joint limits.



(a) Tags (b) Estimation

Fig. 6: ALVAR Tags used to identify the contact point on the target. Multiple tags are employed on the same plane to get a more reliable estimation of the contact plane. In Fig. 6b, the estimation of the plane corresponding to the bottom-left and bottom-right tags was incorrect. However, most of the tags are correct and therefore we can determine the planar pose by majority vote.

B. Hybrid Position-Force Control

The visual servoing algorithm guides each end effector to align with its specified contact. To establish contact for grasping the object, we apply the task space motion controller (4) with a constant velocity towards the object while maintaining the specified orientation. The approach motion continues until a pre-specified target contact force is met.

Once the grasp is complete, based on the known locations of the tags on the object, we approximately know the contact transformations between each end effector and the object: $(p_{E_LT}, p_{E_RT}, R_{E_LT}, R_{E_LT})$, which are constant in their respective end effector's frame, $E_{L/R}$. The weight of the object, F_g, is estimated by projecting the sum of the two force measurements in the direction of gravity.

Let the forces measured by the force/torque sensors be (F_L, F_R). To maintain a stable grasp, we must control the two robot arms to exert a "squeeze" force [22]. Consider only the force components that directly oppose each other, i.e., the contact forces projected along the line between the contacts (given by the unit vector h_{LR}):

$$\eta_L = h_{LR}^\top F_L, \quad \eta_R = -h_{LR}^\top F_R. \tag{5}$$

If the object is stationary and there is no external force applied to the load, then $\eta_L = \eta_R$ is the squeeze force imparted on the object. When the object is moving, we have an additional inertia force $m a_m$ where m is the mass and a_m is the linear acceleration of the load. We regulate the squeeze force by adjusting the position of the two arms, using the standard generalized damper approach [25], which is essentially the integral force control:

$$\begin{aligned} \dot{p}_{0E_L} &= -\zeta K_f g(\eta_L - \eta^d) h_{LR}, \\ \dot{p}_{0E_R} &= -\zeta K_f g(\eta_R - \eta^d) h_{RL} \end{aligned} \tag{6}$$

where η^d is the squeeze force setpoint, ζK_f is the reciprocal of the generalized damper, and $g(\cdot)$ is a dead zone to prevent small errors from winding up the controller and generating unwanted oscillations or instability. We use ζ to avoid breaking off contact, and K_f to ensure closed-loop stability.

Since we do not have rigid grasp, force control is direction-dependent. Since the dry contact friction is proportional to the normal force, a higher squeeze force is usually more desirable (tighter grip, or more centered in the friction cone) than a lower squeeze force (which could cause a slipped grasp or broken contact), though excessive force could cause unwanted deformation or even damages. As recommended in [14] for robust integral force control of a single arm, we use a direction-dependent gain to push in hard but back off slowly (as shown in Fig. 7):

$$\zeta = \begin{cases} 1 & \eta < \eta^d \\ \frac{\eta_1^d - \zeta_f \eta^d + (1-\zeta_f)\eta}{\eta_1^d - \eta^d} & \eta^d \leq \eta < \eta_1^d \\ \zeta_f & \eta \geq \eta_1^d \end{cases} . \tag{7}$$

where the constant ζ_f must be in the range $[0, 1)$ and $\eta_1^d > \eta^d$.

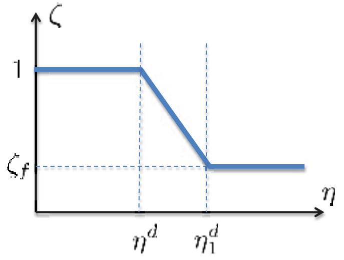


Fig. 7: Integral force feedback gain scheduling to avoid breaking off contact.

To determine a suitable value for ζ_f, we conducted a number of static force tests. Since ζ_f is inversely proportional to the imposed damping, we expect a large ζ_f to have more oscillatory behavior, and a small ζ_f to have larger steady state error. Fig. 8 confirms our expectation. Based on this experiment, we choose $\zeta_f = 0.15$ to balance between transient regulation and steady state response.

Since we project the contact forces along the line segment h_{LR}, the force control problem is essentially one-dimensional. In this case, we use a simplified mass-spring-damper model as shown in Fig. 9. The equation of motion for this system is given by

$$\begin{aligned} \eta_L &= k\left(x_L - x_M + w\right) + c(\dot{x}_L - \dot{x}_M) \\ \eta_R &= -k\left(x_R - x_M - w\right) - c(\dot{x}_R - \dot{x}_M) \\ M\ddot{x}_m &= \eta_L - \eta_R \\ &= k(x_L + x_R - 2x_M) + c(\dot{x}_L + \dot{x}_R - 2\dot{x}_M) \end{aligned} \tag{8}$$

where (x_L, x_R) are the position of the contacts and x_M is the position of the center of mass.

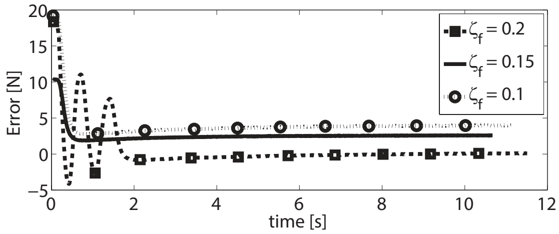


Fig. 8: Comparison of different values for ζ_f.

Ignoring the delay, and considering a constant ζ in (6), the generalized damper controller results in a third-order linear system, as shown in Appendix B. By using the Routh Stability Criterion, we show that the continuous-time system is always stable. Under sample data implementation with variable time delay, significant tuning of the parameter K_f is required to achieve stability and disturbance rejection.

It is also important to note that this model is only valid during contact. For more precise analysis, the complementarity formulation [43] should be used.

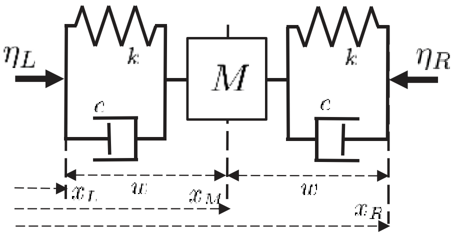


Fig. 9: Simplified 1D contact model for the target object held by two arms. Each contact is controlled to achieve desired squeeze force.

We combine the generalized damper force control law (6) with the task-space motion law (4), resulting in the following hybrid position/force control law:

$$\dot{q} = -J_a^{\dagger} K e - J^{\dagger} \zeta K_f e_f - \Psi(p_{OE}, q). \tag{9}$$

Note that the regular Jacobian J is used in force feedback since there is no additional orientation representation involved.

Experimentally we found that, for highly rigid objects and the significant time delay in our robot platform, the feedback control can be too sensitive with respect to the gains K_f and ζ_f. In this extreme case, we instead use the open loop force control ($K_f = 0$). Of course, since open loop control has no disturbance rejection capability, the force setpoint needs to be chosen sufficiently large in order to maintain contact during motion where the inertial force could be significant.

We define the undesirable region as any configuration where $\left|h_{LR}^T(p_{OE_R} - p_{OE_L})\right| - p_\Delta^d > \epsilon$, where $\epsilon > 0$ to prevent oscillations about the desired setpoint. Furthermore, since it is more desirable for the end effectors to be too close together rather than too far apart, we multiply our potential function by ζ_f from (7) to encourage a consistent grasp.

The error is now specified with respect to the position and orientation of the object. We therefore need to map the desired motion from the center of the object to the end effectors:

$$e = G^\top \begin{bmatrix} \sigma\left(R^d_{OT} R^\top_{OT}\right) \\ p_{OT} - p^d_{OT} \end{bmatrix} \tag{10}$$

where G is the grasp matrix mapping the spatial forces from each contact to the spatial force at the object frame.

Since we are working in the object frame, and treating our end effectors as rigidly attached to their contacts, we need to apply the task space potential fields with respect to the object's position. The joint space potential field function in (4) also needs to be adjusted for each arm to compensate for corrections applied to the other arm. If the robot is in an undesired region of joint space, but only for one arm, the arms may settle into an undesirable equilibrium since the closed chain constraint is not taken into account. To impose the dual-arm constraint, we relate the potential field from the affected arm to the arm that is in a comfortable joint space with the following term in Ψ:

$$\begin{aligned} H &= \begin{bmatrix} I & G^\top_{L,R} \\ G^\top_{R,L} & I \end{bmatrix} \\ \Psi(p_{OT}, q) &= J^\dagger \Big(\gamma_p \rho_p (\nabla \rho_p)^\top \\ &\qquad + \gamma_q H \left(J^\top\right)^\dagger \rho_q (\nabla \rho_q)^\top \Big), \end{aligned} \tag{11}$$

where $G_{L,R}$, and $G_{R,L}$ are the grasp matrices relating spatial forces between the left and right end effectors.

As an additional stability measure, the task space error is constrained by a saturation function to prevent excessive jerk and possible instabilities. This is particularly important in cases of large transient error such as switching between different control modes or large accelerations in the commanded signal.

Since the object is rigidly held between the arms, its pose may be estimated from the locations of the arm end effectors:

$$\begin{aligned} \hat{p}_{OT} &= \frac{1}{2}\big(p_{OE_L} + R_{OE_L} p_{E_L T} + \\ &\qquad p_{OE_R} + R_{OE_R} p_{E_R T}\big), \\ \hat{q}_{OT} &= \frac{1}{2}\left(q_{OE_L} \otimes q_{E_L T} + q_{OE_R} \otimes q_{E_R T}\right), \end{aligned} \tag{12}$$

where $\hat{p}_{OT}$ is the estimate for the object position and $\hat{q}_{OT}$ is the unit quaternion representation of the object orientation. The terms $p_{E_L T}$, $p_{E_R T}$, $q_{E_L T}$, and $q_{E_R T}$ are constant transformation terms in their respective end effector frames that were obtained during the initial grasp.

This estimation is based on the assumption that the contact-to-object transformations $(p_{E_{L/R}T}, q_{E_{L/R}T})$ are fixed, but the actual transformation will likely vary, due to object slippage at the contact, e.g., rolling about h_{LR} and out of alignment. To address this, we include an additional error term to ensure that each arm is positioned correctly:

$$\begin{aligned} e_{p_{L/R}} &= p_{OE_{L/R}} + R_{OE_{L/R}} p_{E_{L/R}T} - \hat{p}_{OT} \\ e_{R_{L/R}} &= \sigma\left(\hat{R}_{OT} R^\top_{E_{L/R}T} R^\top_{OE_{L/R}}\right) \end{aligned} \tag{13}$$

where $\hat{R}_{OT}$ is the rotation matrix representation of the unit quaternion in (12). Since this term has a different orientation error it requires a different representation Jacobian for V in (2). Therefore, we need an additional Jacobian $J_c = V_c J$ where the J_σ terms in V_c are with respect to the error (13), while the J_σ terms in J_a are with respect to the error in (10). We add this term to our controller in (9) to obtain our fully compensated controller:

$$\dot{q} = -J_a^\dagger K e - J^\dagger \zeta K_f e_f - J_c^\dagger K_c e_c - \Psi(p_{OT}, q) \tag{14}$$

where K_c is the diagonal matrix of gains, and e_c is the stacked error vector, defined in (13), $e_c = \left[e_{R_L}, e_{p_L}, e_{R_R}, e_{p_R}\right]^\top$.

C. External Load Compensation

The force feedback in (6) is based on the assumption of no external force on the load. When an external force is present, e.g., gravity load due to the weight of the object, and is coupled to the squeeze direction, h_{LR}, the force setpoints for the two arms can no longer be chosen as the same (i.e. there will be a net force on the body, resulting in motion).

Consider two-arm grasping under an external load as shown in Fig. 10. The force balance along the squeeze direction is given by

$$\underbrace{h^\top_{LR} F_L}_{\eta_L} + \underbrace{h^\top_{LR} F_R}_{\eta_R} = -\underbrace{h^\top_{LR} F_g}_{\eta_g}. \tag{15}$$

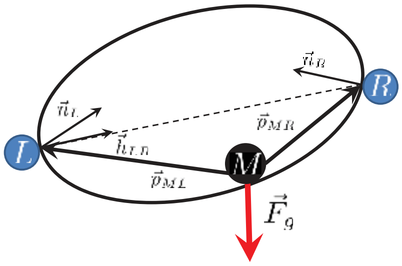


Fig. 10: Generalized squeeze grasp using 2 contacts.

We select the force control setpoint based on a nominal squeeze component η_s (chosen to achieve force closure for a secure grasp) and a distribution of the external force. If the external force is almost orthogonal to the squeeze axis, its effect on the squeeze force control is small, and it may be equally shared by the two arms. If the external force is almost aligned with the squeeze axis, i.e., in the case of the gravity load, the object is almost vertical, and the bottom arm should bear most of the load. This logic is reflected in the squeeze force setpoint rule below:

$$\begin{aligned} \eta^d_L &= \eta_s - \alpha(h^\top_{LR} h_g)\eta_g \\ \eta^d_R &= -\eta_s - \left(1 - \alpha(h^\top_{LR} h_g)\right)\eta_g \end{aligned} \tag{16}$$

where $\alpha(\cdot) \in [0, 1]$ is a function dependent on the pose of the object as shown in Fig. 13 (h_g is the unit vector in the direction of F_g):

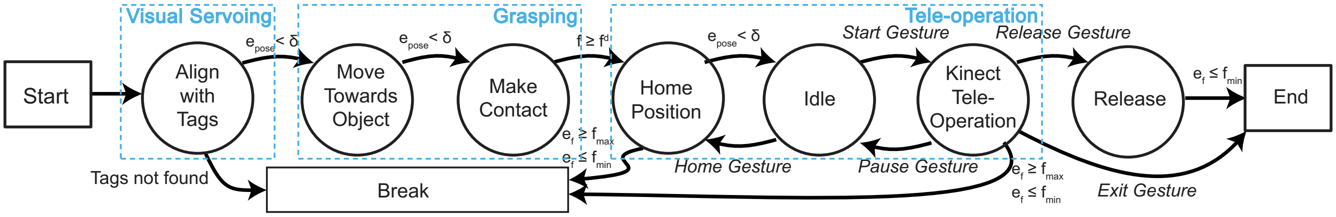
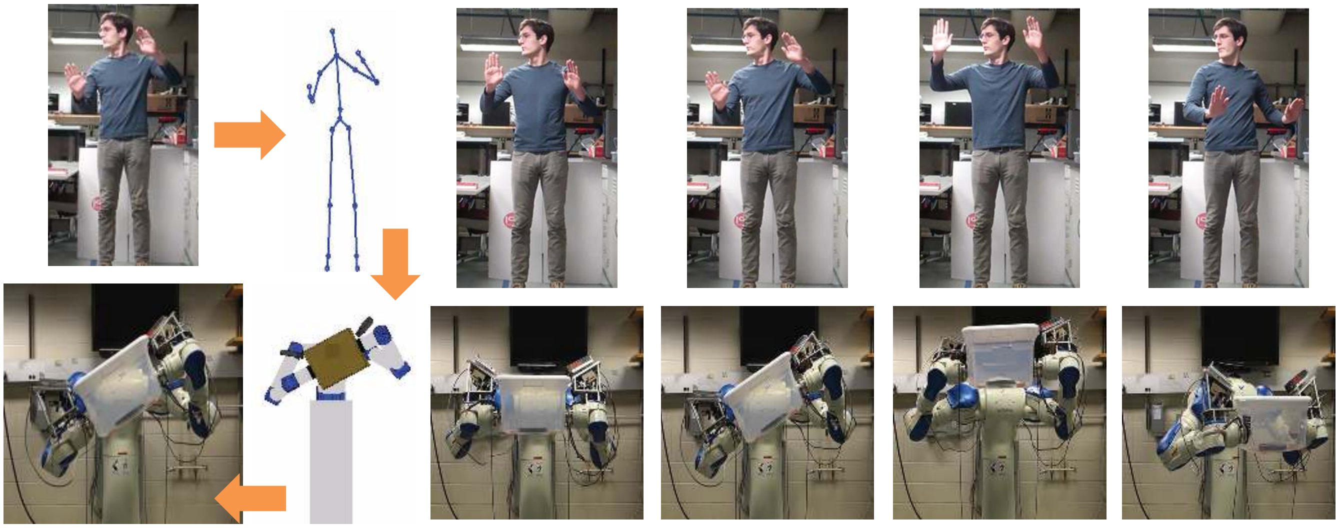


Fig. 11: Finite State Machine demonstrating modes of operation and the state changes causing one to shift to another. Loop break commands include an inability to find the object or the measured forces detecting a slipped grasp or a crushing force.



(a) Pipeline for Gestural Interface

(b) Still frames from running the Kinect Gestural Interface

Fig. 12: In (a), the operator's "skeleton" is detected using a Microsoft Kinect, which we use to interpret the desired pose for the test object. In (b), still frames are shown from running the gestural interface to command the robot.

$$\alpha(x) = \begin{cases} 0 & x > \delta \\ \frac{1}{2} - \frac{1}{2\delta}x & |x| \leq \delta \\ 1 & x < -\delta \end{cases} . \tag{17}$$

The force control setpoints for the two arms are then given by $F_L^d = \eta_L^d h_{LR}$ and $F_R^d = \eta_R^d h_{RL}$.

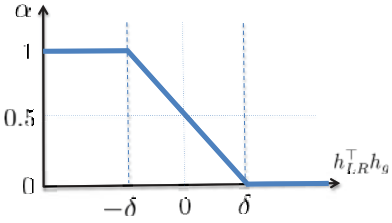


Fig. 13: Load partitioning function based on the pose of the load.

The external force η_g may be time-varying and imprecisely known. In implementation, we replace η_g in (16) with a constant estimate, $\hat{\eta}_g$. To see the effect of this error in force compensation, consider the motion of the load interacting with the external force and the imprecise force estimate:

$$m\ddot{x}_M + d\dot{x}_M + k(x_M - x_M^d) = \eta_g - \hat{\eta}_g \tag{18}$$

where m is the mass of the load, d and k are the damping and spring terms due to the low-level controller, and x_M^d is the position setpoint for the load. Since the system is stable, the effect of $\eta_g - \hat{\eta}_g$ and possibly time-varying x_M^d on the position error is (in the Laplace domain):

$$x_M - x_M^d = \frac{-(ms^2 + ds)x_M^d + \eta_g - \hat{\eta}_g}{ms^2 + ds + k}. \tag{19}$$

Under tight closed loop motion control (from the industrial robot controller), k is large, implying that the effect of varying load trajectories and imperfect force compensation would be small.

D. Gestural Interface

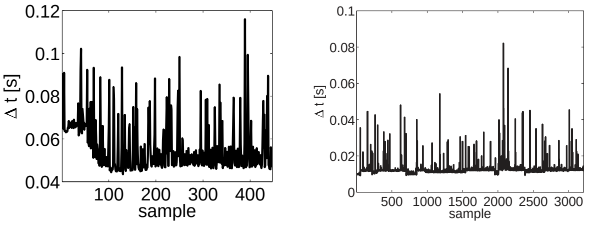
We represent the tele-robotic system as a finite state machine shown in Fig. 11, with three major components: visual servoing, stable grasping, and human-commanded motion. The transition between the states is either through motion or force control to specified locations or thresholds, or through detecting user gestures. After locating the object, positioning the end effectors, and securely grasping the object, the robot moves the object to the specified home position. The *start* gesture (left leg kick) initiates the human-directed motion using the Kinect interface. The *pause* gesture (hands together) stops the human interface and waits in place. The *home* gesture (hands far apart) returns the system to the home configuration. The *exit* gesture (left leg kick) terminates the operation. The *release* gesture (hands far apart) releases the object. As the Kinect SDK can detect and track 48 skeletal points on a human body at 30Hz, a rich vocabulary could be easily constructed to expand the finite state machine user interface. The interface is illustrated in Fig. 12, where the pipeline for interpreting the desired pose for the object is presented in Fig. 12a and some stills from running the interface on our system are shown in Fig. 12b.

An example usage of these gestures in our overall system would be for a tele-operated pick and place task. The tagged object is identified and picked up by the robot. An operator steps in front of the Kinect and takes control using the *start* gesture. The operator then demonstrates the desired trajectory for the object using their relative hand positions, coordinated in real time with the current pose of the object. Once the object reaches the desired goal, such as a conveyor belt, the *release* gesture is made to pass the object along to the next phase of manufacturing.

VI. Results

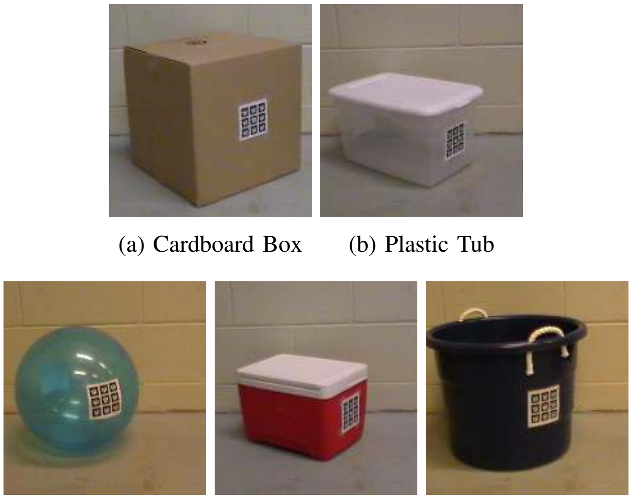
The overall system architecture is shown in Fig. 4, and involves seven main processes running on three Intel Core i7 computers with two running Windows 7 and the computer communicating with the robot running Windows XP. For the ease of development, prototyping, and data analysis, we implement the overall system coordination and control in MATLAB. The system is not real-time, since no real-time operating system is used (except for the low-level robot control to which we do not have access). However, the performance is adequate with sampling times of $\sim$ 20 Hz in visual servoing and $\sim$ 90 Hz in hybrid motion-force control (on average). The measured computation times under different operating modes are shown in Fig. 14.

As is typical in networked control systems [44], [45], the sampling delay is non-uniform. It is known that as long as the spurious long delays do not occur frequently, the closed loop system designed based on the shorter average sampling time will remain stable. The inherent delay in the industrial robot control ($\sim$100ms) is also present and needs to be taken into account. Since we only control the outer loop, the system is essentially a stable first-order system, and the effect of dead time is simply that the gain must be sufficiently small.



(a) Visual Servoing (b) Motion Control

Fig. 14: The loop rates for different operation modes within our system.



(a) Cardboard Box (b) Plastic Tub

(c) Large Ball (d) Cooler (e) Plastic Bucket

Fig. 15: The test objects.

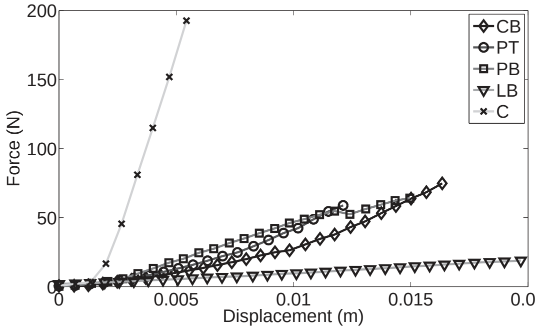


Fig. 16: The stiffness curves for each test object.

Object *[abbr]*	Mass	Surface Stiffness	Distance Between Contacts	Planar Contact	Parallel Surfaces
Cardboard Box [CB]	0.506 kg	Deformable	35.3 cm	Yes	Yes
Plastic Tub [PT]	0.622 kg	Deformable	35.5 cm	Yes	No
Large Ball [LB]	0.167 kg	Highly Deformable	33.4 cm	No	No
Cooler [C]	1.34 kg	Highly Rigid	32.0 cm	Yes	Yes
Plastic Bucket [PB]	1.08 kg	Deformable	36.0 cm	No	No

TABLE I: Physical properties of each test object.

Though stability may be assured, the low gain does affect the disturbance rejection capability, especially in force control.

We tested our control schemes using several different objects with varying characteristics such as rigidity, mass, and planarity of contact surfaces. The test objects are shown in Fig. 15. Their properties are listed in Table I and the functions for the stiffness of each contact surface are presented in Fig. 16. As an additional complication, we place weights inside the objects to move their center of mass away from the middle of their container. Please refer to the video accompanying this manuscript for recordings of the experiments described in this section.

A. Visual Servoing

For all test objects, the robot is able to locate the object, and move to the grasp location via visual servoing. The Large Ball and Plastic Tub objects have nonplanar surfaces, and consequently tags, but they do not present any difficulties.

We consider the simplest case first: two nearly parallel planar contact surfaces positioned in very accessible positions within the robot workspace. The result for the Cardboard Box is shown in Fig. 17. As expected, the position and orientation errors converge to almost zero. However, there is some residual error since the task-space potential fields, designed to keep the center tag within view, prevent the end effector from reaching the exact setpoint.

The most challenging case involves non-planar contact surfaces in a tilted orientation far from the initial robot pose. For the non-planar contact, the ALVAR tags have slightly different orientations depending on the curvature of the surface, but the majority poll approach remains robust. Results with the Large Ball are shown in Fig. 18. The position and orientation errors converge near zero as before. However, at around t =12sec, the error for θ_Y suddenly jumps. This is due to the fact that the arm is near its singularity (with the elbow fully extended).

B. Hybrid Motion-Force Control

For the first test of the hybrid motion-force controller, we specify a series of desired setpoints for the robot to drive the object towards, while maintaining its grasp. This sequence involved a combination of translation and rotation during each step, as demonstrated with the Cardboard Box in Fig. 19. The results are shown in Fig. 20. The steady state motion tracking works well for all objects. The ramp profile for the large displacement command is due to the saturation of the error magnitude in our controller, resulting in a constant velocity command to the robot.

The force responses show noticeable differences between different objects. Objects with higher mass and nonparallel contact surfaces (Plastic Bucket and Plastic Tub) experience substantial force overshoot during pose transition (spikes at t≈26sec in Fig. 20c), due to the inertial force effect. The lighter objects, Cardboard Box and Large Ball, have much tighter force regulation since the disturbance force is smaller.

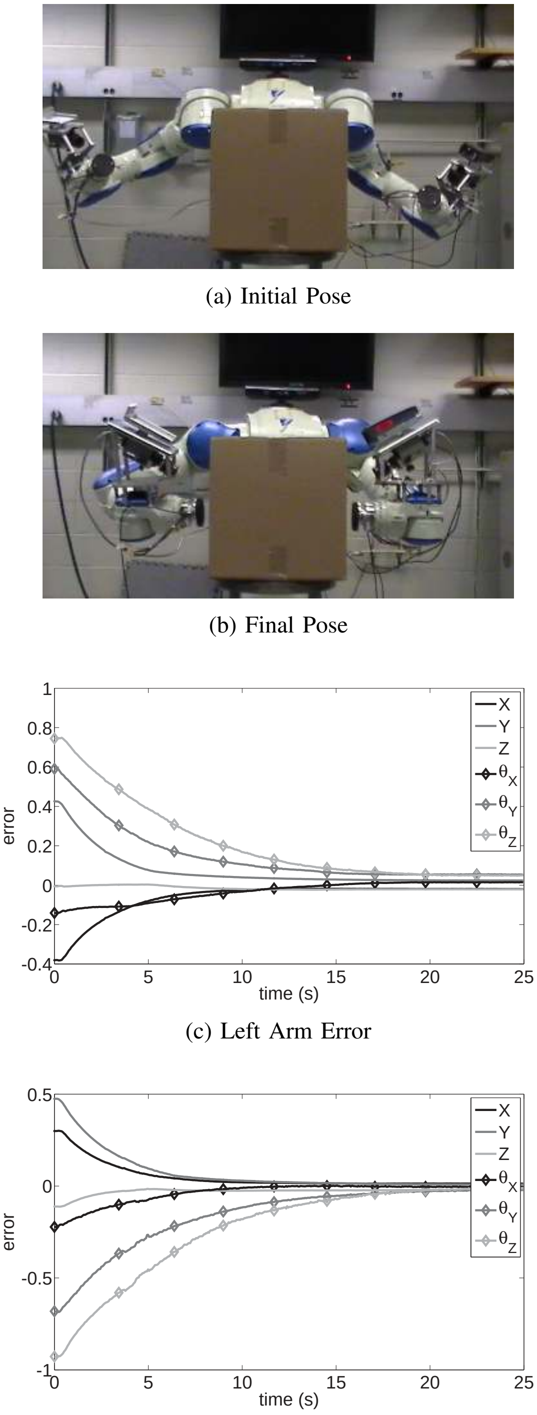
We next drive an object (in this case, the Plastic Tub) to a pose with significant orientation displacement from the initial pose, causing one force/torque sensor to be loaded with the majority of the weight. In addition, we affixed a 2.3 kg metal plate to the bottom of the Plastic Tub to bring the center of mass significantly off center. The results from this experiment are shown in Fig. 21. As the angle between gravity and the squeeze axis reduced, the uncompensated control law became more unstable until finally a failure condition was reached at $\sim$ 12 seconds. A noticeable drift in the position control, seen in Fig. 21a, occurred as a result of gravity affecting the force control.

As a more challenging scenario, we consider a moving load case. We place a 0.9 kg medicine ball in the Plastic Bucket, and then drive it to follow a sinusoid, causing the ball to shift from one side of the container to the other. Spikes occur in the force error, in Fig. 22b. Each of these spikes is due to the ball colliding into a side of the container. The motion of the arms is not significantly affected by this disturbance, due to the rigidity of the arm and the tight low-level control loop.

Finally, we command the robot motion using the Microsoft Kinect interface. The object under test is the Plastic Tub, again outfitted with a 0.9 kg mass rigidly mounted on one of its sides. As seen in the results shown in Fig. 23, the control law tries to keep up with the reference signal (from the user), despite its rapid variation, but we see the same linear convergence as in Fig. 20 due to the error saturation in our controller. For more compliant objects, the robot manages to maintain grasp stability throughout the entire process, with minimal force error as shown in Fig. 23c, even during periods with large orientation and translation excursions along the squeezing axis. For the more rigid Plastic Bucket, there is a much larger force transient from the coupling of motion-induced force to the squeeze axis. In this case, the force setpoint is increased to 60N, and the robot is able to maintain a stable grasp throughout the motion sequence.

VII. Discussion and Future Work

We presented the development and results of a dual-arm telerobotic system involving the integration of several sensors and actuators with an industrial robot. The main components of the



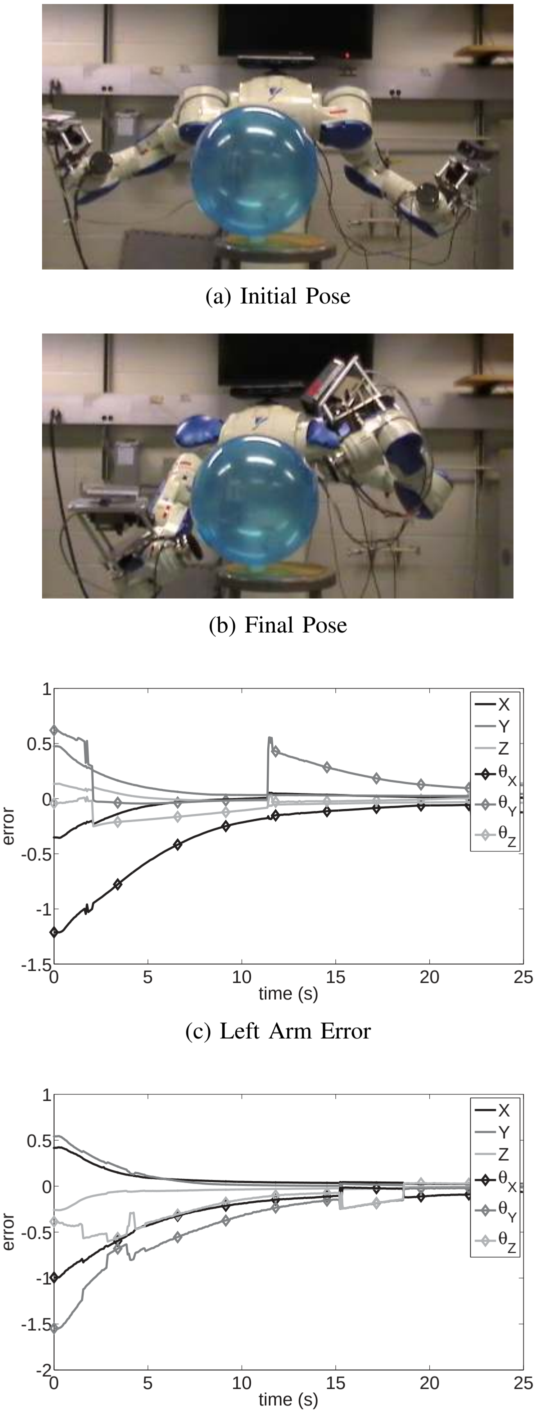
(a) Initial Pose

(b) Final Pose

(c) Left Arm Error

(d) Right Arm Error

Fig. 17: Convergence of the visual servoing control system after placing the Cardboard Box in the middle of the workspace. Position error is given in meters, while orientation error is given in radians.



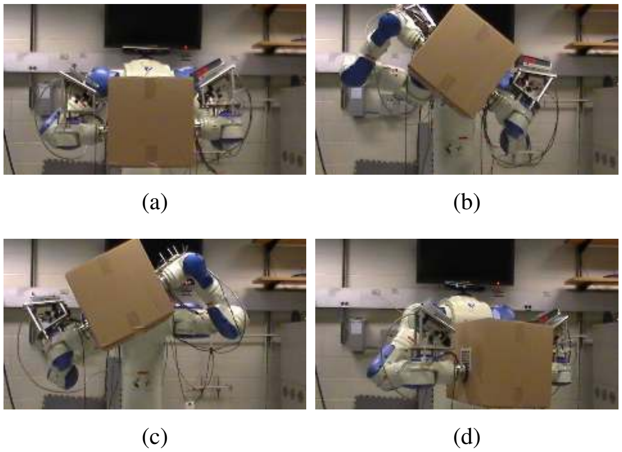
(a) Initial Pose

(b) Final Pose

(c) Left Arm Error

(d) Right Arm Error

Fig. 18: Convergence of the visual servoing control system after placing the Large Ball slightly away from center and with a significant off-angle orientation.



(a) (b)

(c) (d)

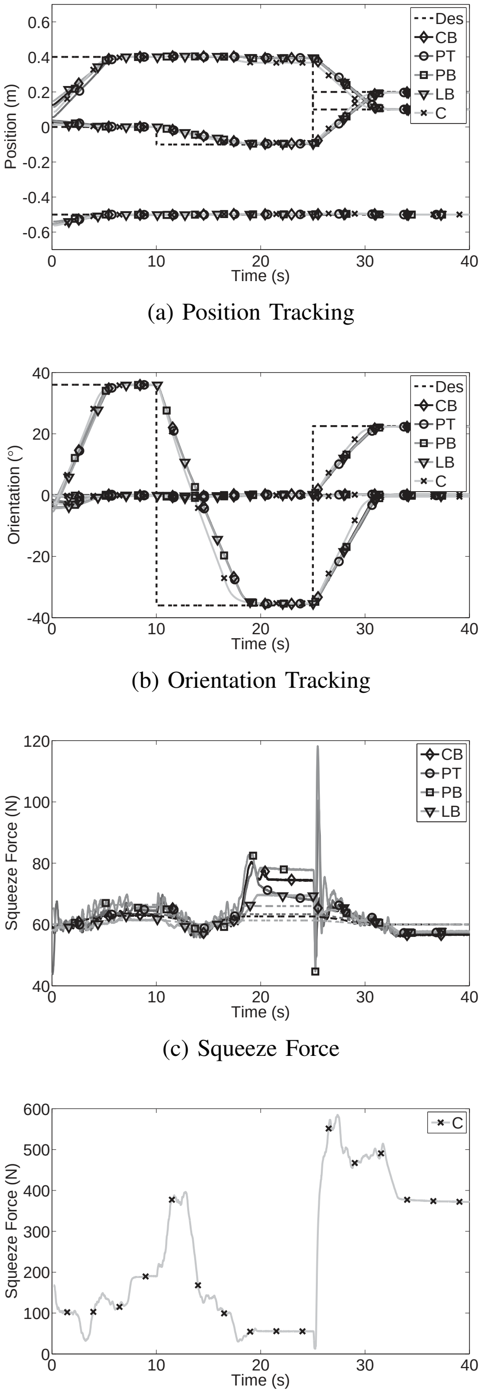
Fig. 19: Sequence of setpoints demonstrated on the Cardboard Box. Each object started in the same pose (19a), translated vertically and rotated by 36° (19b), rotated in the opposite direction by −72° with a small leftward translation (19c), and finally to a pose rotated about the vertical axis and down to the right near the robot's waist.

system include vision-guided motion control, redundancy resolution, collision avoidance, squeeze force control, load compensation, and human gestural interface. We integrated these interconnected components in a robust and versatile distributed control and communication architecture, and demonstrated its effectiveness in manipulating a variety of objects with different shapes, surface textures, weights, and mass distributions.

While the implementation and demonstration is for a specific platform, we mostly draw on off-the-shelf components and software, so the approach is easily extensible to other robots and platforms. We used an industrial robot controller, and despite its significant time delay, we were able to achieve robust performance for complex motion and force objectives.

A limitation to this implementation is that it is implicitly only effective as a local planner; so in the future we will incorporate global planning methods to handle local equilibria and introduce more sophisticated redundancy resolution. In the current system, we used high-friction contact pads designed for non-rigid grasping to have high friction . We are also investigating the use of modern, articulated grippers in the types of manipulation tasks studied here in which an enveloping grasp is impossible.

To demonstrate the generality of our approach, we are currently extending our implementation to other industrial dual-arm systems such as the Baxter by Rethink Robotics. We are interested in different use cases for this dual-arm system, such as human-directed assembly, bimanual pick and place for large or heavy objects, and two-handed household tasks for assistive robots. Motivated by fabric layups in composites manufacturing, we are also investigating a variation on the earlier discussed complementarity force control problem in which, instead of applying a squeeze force on a rigid body, the



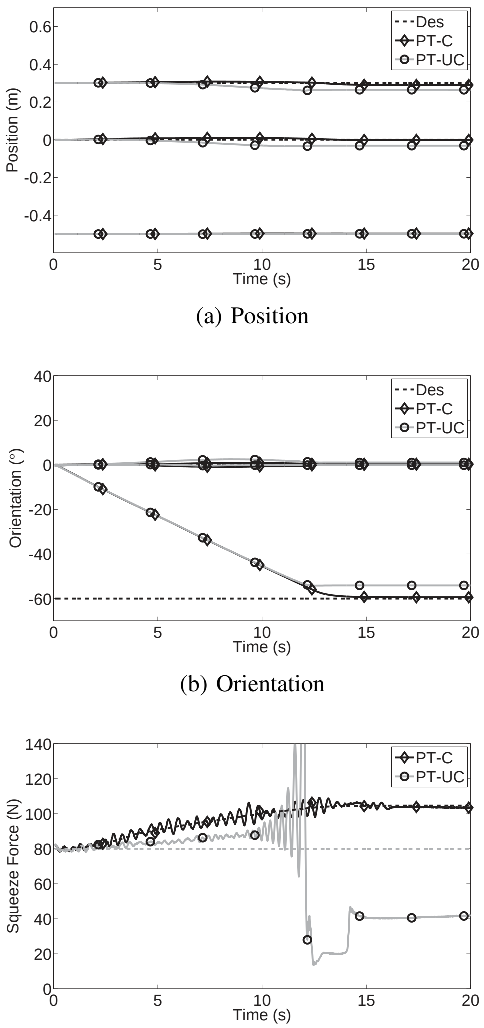
(a) Position Tracking

(b) Orientation Tracking

(c) Squeeze Force

(d) Squeeze Force for Cooler

Fig. 20: The motion and force results for all test objects. Position and orientation have almost identical performance. The Plastic Bucket showed a large spike in the force control during a large transient due to its more rigid surface. The Cooler was controlled using the potential field approach and had much wilder fluctuations in its force control.



(a) Position

(b) Orientation

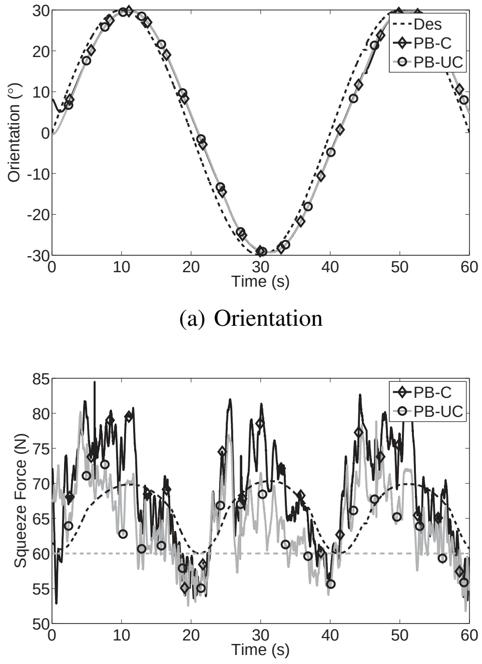
(c) Squeeze Force

Fig. 21: Results from experiments with and without load compensation for the Plastic Tub under a large angular setpoint. The uncompensated run is denoted PT-UC, and the compensated run is denoted PT-C. The uncompensated run failed at ~ 12 seconds due to force overshoot.

robot must maintain a prescribed tension in a flexible object during motion.

Appendix A
Singularities of the Motoman Dual-Arm Robot

The Motoman dual-arm robot has 15 rotational degrees of freedom. Since the task space involves both arms, the corresponding Jacobian is 12×15. To analyze the singularity, we will consider two sub-cases: the 7-dof single arm and the 8-dof arm and torso. Then two cases are then combined for the full dual-arm system. We use the shorthand notation s_i for $\sin\theta_i$ and c_i for $\cos\theta_i$.



(a) Orientation

(b) Squeeze Force

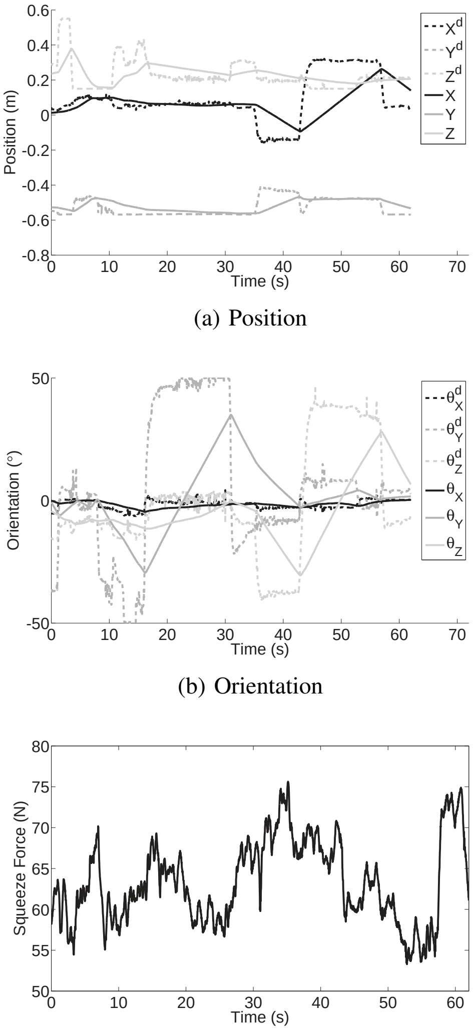
Fig. 22: Results from running two trials with a sinusoidal trajectory and a dynamic load. Both runs operated on the Plastic Bucket, with one trial incorporating the gravity compensation to its desired force setpoint (denoted by PB-C), while the other did not (denoted by PB-UC). The large spikes in the force error, located at ~ 5, 27, and 48 seconds, show when the moving load impacted a wall of the container.

A. Singularity of 7-DOF Single Arm

Consider the left arm (joints 2-8) and the corresponding coordinate frame assignment shown in Fig. 5. As in [46], choose the reference task frame, denoted $\mathbf{A}$, to be at the wrist spherical joint where axes 6, 7, and 8 intersect. The corresponding Jacobian in coordinate-free form is

$$J_A = \Big[\begin{matrix} \vec{h}_2 & \vec{h}_3 & \vec{h}_4 & \vec{h}_5 \\ \vec{h}_2 \times \vec{p}_{46} & \vec{h}_3 \times \vec{p}_{46} & \vec{h}_4 \times \vec{p}_{46} & \vec{h}_5 \times \vec{p}_{56} \end{matrix} \quad \begin{matrix} \vec{h}_6 & \vec{h}_7 & \vec{h}_8 \\ 0 & 0 & 0 \end{matrix} \Big]. \tag{20}$$

First consider $[\vec{h}_6 \ \ \vec{h}_7 \ \ \vec{h}_8]$ at full rank. Then J_A loses rank if and only if $J_{A_1} = [\vec{h}_2 \times \vec{p}_{46} \ \ \vec{h}_3 \times \vec{p}_{46} \ \ \vec{h}_4 \times \vec{p}_{46} \ \ \vec{h}_5 \times \vec{p}_{56}]$ loses rank. If the shoulder spherical joint is not singular, i.e., $s_3 \neq 0$, then J_{A_1} loses rank if and only if $\vec{p}_{46} \cdot \vec{h}_5 \times \vec{p}_{56} = 0$ which is equivalent to $c_5 = 0$. When $s_3 = 0$, the first and third columns are linearly dependent, so we only need to consider the last three columns. The singularity condition may be obtained geometrically from $\vec{p}_{56}$ collinear with $\vec{h}_4$ (or $c_5 = 0$) or $\vec{h}_3$, $\vec{h}_4$, $\vec{p}_{46}$ coplanar (or $c_4 = 0$).



(a) Position

(b) Orientation

(c) Squeeze Forces

Fig. 23: Results of a user commanding the position setpoint using a Microsoft Kinect. The object under manipulation was the Plastic Tub with a 0.9 kg mass mounted to an interior wall.

Now consider the remaining case where $\begin{bmatrix}\vec{h}_6 & \vec{h}_7 & \vec{h}_8\end{bmatrix}$ is rank-deficient. For this case, choose the reference task frame **B** to be at the shoulder spherical joint, i.e., the intersection of $\vec{h}_2$, $\vec{h}_3$, $\vec{h}_4$. The corresponding Jacobian is

$$J_B = \begin{bmatrix} \vec{h}_2 & \vec{h}_3 & \vec{h}_4 & \vec{h}_5 & \vec{h}_6 & \vec{h}_7 & \vec{h}_8 \\ 0 & 0 & 0 & \vec{h}_5 \times \vec{p}_{54} & \vec{h}_6 \times \vec{p}_{64} & \vec{h}_7 \times \vec{p}_{64} & \vec{h}_8 \times \vec{p}_{64} \end{bmatrix}. \tag{21}$$

We only need to consider $s_3 \neq 0$. By using the same reasoning as before, we arrive at the singularity condition of $c_5 = 0$ when $s_7 \neq 0$, and, for the case $s_7 = 0$, the singularity condition is $c_5 c_6 = 0$. When both the shoulder and wrist are singular, $s_3 = 0$ and $s_7 = 0$, the arm is also singular.

To summarize, the singularity condition for the 7-dof arm may be written compactly as

$$c_5(s_3 + c_4)(s_7 + c_6)(s_3 + s_7) = 0. \tag{22}$$

B. Singularity of 8-DOF Arm and Torso

When the 7-dof arm is singular, the torso degree of freedom may remove the singularity. First consider $c_5 = 0$. When either $s_7 \neq 0$, the torso adds a term $\vec{h}_1 \times p_{16}$ to possibly make J_{A_1} full rank. The singularity condition in this case becomes $\ell_0 c_3 + \ell_1 s_2 s_3 = 0$. The same condition is obtained when $s_3 \neq 0$. When $s_3 = 0$, we need to consider the full matrix, and the singular condition becomes $s_4 = 0$. Similarly, when $s_7 = 0$, the singular condition becomes $s_6 = 0$. When both shoulder and wrist are at singularity, $s_3 = s_7 = 0$, the Jacobian matrix may be made full rank with torso motion, and the singularity condition becomes $c_5(\ell_0(c_4 s_6 + s_4 s_5 c_6) - (\ell_1 + \ell_2) c_2 c_5 c_6) = 0$. Next consider the shoulder singularity $s_3 = c_4 = 0$. The addition of $\vec{h}_1 \times \vec{p}_{16}$ due to torso rotation changes the singularity condition to the rank deficiency of $\begin{bmatrix}\vec{h}_1 \times \vec{p}_{16} & \vec{h}_3 \times \vec{p}_{46} & \vec{h}_5 \times \vec{p}_{56}\end{bmatrix}$, which simplifies to $(\ell_1 + \ell_2) c_2 c_5 - \ell_0 s_5 = 0$. Similarly, for the wrist singularity case, $s_7 = c_6 = 0$, the torso adds a column $\vec{h}_1 \times \vec{p}_{12}$. The singularity condition becomes $\ell_0 c_3 + \ell_1 s_2 s_3 = 0$. Combining all cases above, the singularity condition for the 8-dof arm may be written as

$$\begin{aligned} &(c_5 + (\ell_0 c_3 + \ell_1 s_2 s_3)(s_3 + s_4)(s_6 + s_7)) \\ &\cdot(s_3 + s_7 + c_5(\ell_0(c_4 s_6 + s_4 s_5 c_6) - (\ell_1 + \ell_2) c_2 c_5 c_6)) \\ &\cdot(s_3 + c_4 + ((\ell_1 + \ell_2) c_2 c_5 - \ell_0 s_5)) \\ &\cdot(s_7 + c_6 + (\ell_0 c_3 + \ell_1 s_2 s_3)) = 0. \end{aligned} \tag{23}$$

C. Singularity of Full 15-DOF Dual-Arm System

For the 15-dof dual-arm system, the torso may remove the singularity of either the right or the left arm. Therefore, the combined arm is singular (with 12-dof task degree of freedom) if both of the following conditions are satisfied

1. The 8-dof left arm or the 7-dof right arm is singular.
2. The 7-dof left arm or the 8-dof right arm is singular.

Appendix B
Stability of Force Control of a Compliant Object

To simplify the stability analysis, we consider a constant linear feedback in the position accommodation force control law (6). Considering only along the squeeze direction, we have

$$\dot{x}_L = -\zeta K_f(\eta_L - \eta^d) \quad \dot{x}_R = \zeta K_f(\eta_R - \eta^d). \tag{24}$$

Substituting in the expression of the contact force in (8), we have

$$\begin{aligned} \dot{x}_L &= -\zeta K_f(k(x_L - x_m + w) + c(\dot{x}_L - \dot{x}_m) - \eta^d) \\ \dot{x}_R &= \zeta K_f(-k(x_R - x_m - w) - c(\dot{x}_R - \dot{x}_m) - \eta^d). \end{aligned} \tag{25}$$

Summing the two we have

$$\begin{aligned} &\dot{x}_L + \dot{x}_R \\ &= -\zeta K_f(k(x_L + x_R) + c(\dot{x}_L + \dot{x}_R) - 2kx_m - 2c\dot{x}_m) \\ &= -\zeta K_f(1 + \zeta K_f c)^{-1}(k(x_L + x_R) - 2kx_m - 2c\dot{x}_m). \end{aligned} \tag{26}$$

Represented in the Laplace domain, we have

$$x_L + x_R = \frac{2\zeta K_f(cs + k)}{s(1 + \zeta K_f c) + \zeta K_f k} x_m. \tag{27}$$

The equation of motion in (8) in the Laplace domain is

$$(Ms^2 + 2cs + 2k)x_m = (cs + k)(x_L + x_R). \tag{28}$$

Combining the two equations, we have the third-order closed loop characteristic polynomial (the right and left arm work in unison, reducing the order by one) given by

$$\pi(s) = s(M(1 + \zeta K_f c)s^2 + (2c + Mk\zeta K_f)s + 2k). \tag{29}$$

The pole at zero corresponds to the position dynamics. Since we are focusing only on the force control, the system may be shifted anywhere on a line. A motion loop feedback will stabilize this pole. The two stable poles correspond to the force control loop, showing the mass-spring-damper system will reach the equilibrium under the integral force control.

Acknowledgment

The authors would like to thank John Wason, the developer and architect of Robot Raconteur, for the development of a powerful and versatile software package and his continuing help with our work. The authors would also like to thank Glenn Saunders and Ken Myer Jr. for the design, fabrication, and installation of the camera, force/torque sensor, and end effector mounts. Matt Krolick implemented the Robot Raconteur services for the Robotiq hands and ATI force/torque sensors. Alex Nolet implemented the Robot Raconteur interface for the Microsoft Kinect sensor. Lu Lu corrected several typos in the stability analysis.

This work was supported in part by the Center for Automation Technologies and Systems (CATS) under a block grant from the New York State Empire State Development Division of Science, Technology and Innovation (NYSTAR) and in part by the National Science Foundation (NSF) Smart Lighting Engineering Research Center (EEC-0812056).

References

[1] T. Brogårdh, "Present and future robot control development: an industrial perspective," *Annual Reviews in Control*, vol. 31, no. 1, pp. 69–79, 2007.

[2] B. Shimano, C. Geschke, and C. Spalding III, "VAL-II: A new robot control system for automatic manufacturing," in *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 278–292, IEEE, 1984.

[3] N. Nayak and A. Ray, "An integrated system for intelligent seam tracking in robotic welding. II. design and implementation," in *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1898–1903, IEEE, 1990.

[4] E. Castro, S. Seereeram, J. Singh, A. A. Desrochers, and J. Wen, "A real-time computer controller for a robotic filament winding system," *Journal of Intelligent and Robotic Systems*, vol. 7, no. 1, pp. 73–93, 1993.

[5] A. Blomdell, G. Bolmsjo, T. Brogardh, P. Cederberg, M. Isaksson, R. Johansson, M. Haage, K. Nilsson, M. Olsson, T. Olsson, and A. Robertsson, "Extending an industrial robot controller," *IEEE Robotics & Automation Magazine*, vol. 12, pp. 85–94, Sept. 2005.

[6] T. Olsson, M. Haage, H. Kihlman, R. Johansson, K. Nilsson, A. Robertsson, M. Björkman, R. Isaksson, G. Ossbahr, and T. Brogårdh, "Cost-efficient drilling using industrial robots with high-bandwidth force feedback," *Robotics and Computer-Integrated Manufacturing*, vol. 26, no. 1, pp. 24–38, 2010.

[7] Unimation Incorporated, A Westinghouse Company, *User's Guide to VAL II Programming Manual, Ver. 2.0*, Aug. 1986.

[8] ABB Flexible Automation, *RAPID reference manual*, 2005.

[9] "DARPA Robotics Challenge Trials 2013." www.theroboticschallenge.org, 2012. Accessed: 12/24/2013.

[10] J. Wason and J. Wen, "Robot Raconteur: A communication architecture and library for robotic and automation systems," in *IEEE International Conference on Robotics and Automation (ICRA)*, (Trieste, Italy), Aug. 2011.

[11] S. Edwards, "ROS for industrial applications." www.rosindustrial.org, 2012. Accessed: 9/12/2012.

[12] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," *International Journal of Robotics Research*, vol. 5, no. 1, pp. 90–98, 1986.

[13] VTT Technical Research Centre of Finland, "ALVAR-a library for virtual and augmented reality." http://virtual.vtt.fi/virtual/proj2/multimedia/alvar/, 2013. Accessed: 02/20/2013.

[14] L. Wilfinger, J. Wen, and S. Murphy, "Integral force control with robustness enhancement," *IEEE Control Systems Magazine*, vol. 14, pp. 31–40, Feb. 1994.

[15] D. Kruse, R. Radke, and J. Wen, "A sensor-based dual-arm telerobotic manipulation platform," in *IEEE International Conference on Automation Science and Engineering (CASE)*, pp. 350–355, 2013.

[16] C. Smith, Y. Karayiannidis, L. Nalpantidis, X. Gratal, P. Qi, D. V. Dimarogonas, and D. Kragic, "Dual arm manipulation: A survey," *Robotics and Autonomous Systems*, vol. 60, pp. 1340–1353, Oct. 2012.

[17] F. Chaumette and S. Hutchinson, "Visual servo control. I. Basic approaches," *IEEE Robotics & Automation Magazine*, vol. 13, pp. 82–90, Dec. 2006.

[18] F. Chaumette and S. Hutchinson, "Visual servo control. II. Advanced approaches [tutorial]," *IEEE Robotics & Automation Magazine*, vol. 14, pp. 109–118, Mar. 2007.

[19] F. Burget, A. Hornung, and M. Bennewitz, "Whole-body motion planning for manipulation of articulated objects," *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1656–1662, May 2013.

[20] A. Dietrich, T. Wimbock, and A. Albu-Schaffer, "Dynamic whole-body mobile manipulation with a torque controlled humanoid robot via impedance control laws," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3199–3206, IEEE, Sept. 2011.

[21] T. Yoshikawa, "Multifingered robot hands: Control for grasping and manipulation," *Annual Reviews in Control*, vol. 34, pp. 199–208, Dec. 2010.

[22] J. Wen and K. Kreutz-Delgado, "Motion and force control of multiple robotic manipulators," *Automatica*, vol. 28, pp. 729–743, July 1992.

[23] N. Xi, T.-J. Tarn, and A. K. Bejczy, "Intelligent Planning and Control for Multirobot Coordination: An Event-Based Approach," *IEEE Transactions on Robotics and Automation*, vol. 12, no. 3, pp. 439–452, 1996.

[24] T. Yoshikawa, "Virtual Truss Model for Characterization of Internal Forces for Multiple Finger Grasps," *IEEE Transactions on Robotics and Automation*, vol. 15, no. 5, pp. 941–947, 1999.

[25] M. T. Mason, “Compliance and force control for computer controlled manipulators,” *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 11, no. 6, pp. 418–432, 1981.

[26] F. Basile, F. Caccavale, P. Chiacchio, J. Coppola, and C. Curatella, “Task-oriented motion planning for multi-arm robotic systems,” *Robotics and Computer-Integrated Manufacturing*, vol. 28, no. 5, pp. 569 – 582, 2012.

[27] K. Hauser and V. Ng-Thow-Hing, “Randomized multi-modal motion planning for a humanoid robot manipulation task,” *International Journal of Robotics Research*, vol. 30, pp. 678–698, Dec. 2010.

[28] J.-S. You, D.-H. Kim, S.-J. Lim, S.-P. Kang, J. Y. Lee, and C.-S. Han, “Development of manipulation planning algorithm for a dual-arm robot assembly task,” *IEEE International Conference on Automation Science and Engineering (CASE)*, 2012.

[29] B. Cohen, S. Chitta, and M. Likhachev, “Search-based planning for dual-arm manipulation with upright orientation constraints,” *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3784–3790, May 2012.

[30] S. Dalibard, A. Nakhaei, F. Lamiraux, and J.-P. Laumond, “Manipulation of documented objects by a walking humanoid robot,” *IEEE-RAS International Conference on Humanoid Robots*, pp. 518–523, Dec. 2010.

[31] S. Hayati, T. S. Lee, K. S. Tso, P. G. Backes, and J. Lloyd, “A unified teleoperated-autonomous dual-arm robotic system,” *IEEE Control Systems Magazine*, vol. 11, no. 2, pp. 3–8, 1991.

[32] J. L. Raheja, R. Shyam, U. Kumar, and P. B. Prasad, “Real-Time Robotic Hand Control Using Hand Gestures,” *International Conference on Machine Learning and Computing (ICMLC)*, pp. 12–16, 2010.

[33] G. Du, P. Zhang, J. Mai, and Z. Li, “Markerless Kinect-Based Hand Tracking for Robot Teleoperation,” *International Journal of Advanced Robotic Systems*, p. 1, 2012.

[34] P. Gil, C. Mateo, and F. Torres, “3D Visual Sensing of the Human Hand for the Remote Operation of a Robotic Hand,” *International Journal of Advanced Robotic Systems*, p. 1, 2014.

[35] J. Wen and K. Kreutz-Delgado, “The attitude control problem,” *IEEE Transactions on Automatic Control*, vol. 36, pp. 1148–1162, Oct. 1991.

[36] C. W. Wampler, “Manipulator inverse kinematic solutions based on vector formulations and damped least-squares methods,” *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 16, no. 1, pp. 93–101, 1986.

[37] C. B. Bose and J. Amir, “Design of fiducials for accurate registration using machine vision,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 12, pp. 1196–1200, 1990.

[38] M. Fiala, “Designing highly reliable fiducial markers,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 7, pp. 1317–1324, 2010.

[39] E. Olson, “AprilTag: A robust and flexible visual fiducial system,” in *IEEE International Conference on Robotics and Automation (ICRA)*, 2011.

[40] A. Rice, A. Beresford, and R. Harle, “Cantag: an open source software toolkit for designing and deploying marker-based vision systems,” in *IEEE International Conference on Pervasive Computing and Communications (PerCom)*, p. 10, IEEE, 2006.

[41] Z. Zhang, “A flexible new technique for camera calibration,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, pp. 1330–1334, Nov. 2000.

[42] G. Schweighofer and A. Pinz, “Robust pose estimation from a planar target,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, pp. 2024–2030, Dec. 2006.

[43] J.-S. Pang and J. C. Trinkle, “Complementarity formulations and existence of solutions of dynamic multi-rigid-body contact problems with Coulomb friction,” *Mathematical Programming*, vol. 73, no. 2, pp. 199–226, 1996.

[44] W. Zhang, M. S. Branicky, and S. M. Phillips, “Stability of networked control systems,” *IEEE Control Systems Magazine*, vol. 21, no. 1, pp. 84–99, 2001.

[45] T. C. Yang, “Networked control system: a brief survey,” *IEEE Proceedings-Control Theory and Applications*, vol. 153, no. 4, pp. 403–412, 2006.

[46] A. Fijany and A. K. Bejczy, “Efficient Jacobian inversion for the control of simple robot manipulators,” in *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 999–1007, IEEE, 1988.

Daniel Kruse received his B.S. in Electrical Engineering from Virginia Commonwealth University in 2011, and is currently a Ph.D. student at Rensselaer Polytechnic Institute in the Electrical, Computer, and Systems Engineering department. His research interests include autonomous and shared control systems for humanoid robots to perform bimanual or coordinated tasks using force and vision feedback.

John T. Wen received his B.Eng. from McGill University in 1979, M.S. from University of Illinois in 1981, and Ph.D. from Rensselaer Polytechnic Institute in 1985, all in Electrical Engineering. From 1981-1982, he was a system engineer at Fisher Controls. From 1985-1988, he was a member of technical staff at the Jet Propulsion Laboratory. Since 1988, he has been with Rensselaer Polytechnic Institute where he is currently a professor and head of the Industrial and Systems Engineering Department. From 2005-2013, he served as the Director of Center for Automation Technologies and Systems (CATS).

Richard J. Radke is a Full Professor in the Electrical, Computer, and Systems Engineering department at Rensselaer Polytechnic Institute. He has B.A. and M.A. degrees in computational and applied mathematics from Rice University, and M.A. and Ph.D. degrees in electrical engineering from Princeton University. His research interests include computer vision problems related to range imagery, camera networks, and radiotherapy applications. Dr. Radke is a Senior Member of the IEEE and an Associate Editor of *IEEE Transactions on Image Processing*. His textbook *Computer Vision for Visual Effects* was published by Cambridge University Press in 2012.