Pulling Open Novel Doors and Drawers with Equilibrium Point Control

Advait Jain
Healthcare Robotics Laboratory, Georgia Tech
advait@cc.gatech.edu

Charles C. Kemp
Healthcare Robotics Laboratory, Georgia Tech
charlie.kemp@bme.gatech.edu

***Abstract*— A large variety of doors and drawers can be found within human environments. Humans regularly operate these mechanisms without difficulty, even if they have not previously interacted with a particular door or drawer. In this paper, we empirically demonstrate that equilibrium point control can enable a humanoid robot to pull open a variety of doors and drawers without detailed prior models, and infer their kinematics in the process.**

Our implementation uses a 7 DoF anthropomorphic arm with series elastic actuators (SEAs) at each joint, a hook as an end effector, and low mechanical impedance. For our control scheme, each SEA applies a gravity compensating torque plus a torque from a simulated, torsional, viscoelastic spring. Each virtual spring has constant stiffness and damping, and a variable equilibrium angle. These equilibrium angles form a joint space equilibrium point (JEP), which has a corresponding Cartesian space equilibrium point (CEP) for the arm's end effector.

We present two controllers that generate a CEP at each time step (ca. 100ms) and use inverse kinematics to command the arm with the corresponding JEP. One controller produces a linear CEP trajectory and the other alters its CEP trajectory based on real-time estimates of the mechanism's kinematics. We also present results from empirical evaluations of their performance (108 trials). In these trials, both controllers were robust with respect to variations in the mechanism, the pose of the base, the stiffness of the arm, and the way the handle was hooked. We also tested the more successful controller with 12 distinct mechanisms. In these tests, it was able to open 11 of the 12 mechanisms in a single trial, and successfully categorized the 11 mechanisms as having a rotary or prismatic joint, and opening to the right or left. Additionally, in the 7 out of 8 trials with rotary joints, the robot accurately estimated the location of the axis of rotation.

I. INTRODUCTION

A large variety of doors and drawers can be found within human environments. Operating these mechanisms plays a role in many daily activities, such as moving within an environment or retrieving an object that has been stored. Being able to operate these same mechanisms would help service robots assist with similar activities.

Humans regularly operate these mechanisms without difficulty, even if they have not previously interacted with a particular door or drawer. This would be an advantageous capability for robots, and some progress has been made in this direction [1], [2]. So far, however, researchers have either employed relatively complex control schemes or used pre-existing models of the specific door or drawer. Moreover, researchers have yet to empirically demonstrate that their methods can succeed given real-world variation. Within this paper, we present a straightforward form of impedance control, which we call *equilibrium point control* (EPC). We demonstrate that EPC in conjunction with low mechanical impedance can be used to pull open novel doors and drawers, and that our implementation is robust to variations in the mechanism, the pose of the base, the stiffness of the arm, and the way the handle is hooked.

A. Low-Impedance Manipulation

Researchers have made compelling arguments for the benefits of robots with low mechanical impedance [3], [4]. As has often been noted, these arguments are particularly relevant for manipulation within human environments, since robots are likely to be uncertain about the state of the world. At minimum, low-impedance manipulation can reduce the forces and moments resulting from contact, and thus reduce the risk of damage to the robot, the environment, and nearby people.

In addition, we believe that low-impedance manipulation can improve the performance of robots with respect to common tasks within human environments. Although low-impedance manipulation often results in poorer performance when moving the end effector through a pre-defined trajectory [5], we believe this type of evaluation fails to capture the challenges of manipulation in human environments.

The common task we address in this paper (opening novel doors and drawers) illustrates forms of real-world task variation and uncertainty that a service robot is likely to encounter. For example, our system requires that the robot be able to haptically explore the environment and make unexpected contact with rigid components of the environment (e.g., when haptically finding and hooking onto the handle). Likewise, our system requires that the robot accommodate unexpected displacements (e.g., the constrained trajectory of the handle). At their core, these examples require that the world be able to move the robot's arm, and do so without generating large forces and torques. As such, we believe that low-impedance manipulation is well-matched to the task.

Our robot's arm has low mechanical impedance at all links. The robot arm achieves this through both passive and active means [6]. A service robot in daily operation may not be able to restrict its contact with the world to its end effector, and may intentionally make contact with proximal links in order to perform tasks. As such, we believe there is value in using robots and control schemes that provide low mechanical impedance for contact at any point along the arm.

Within our research, the end effector stiffness of the manipulator of our robot is relatively low when compared to other impedance controlled arms. For example, it is lower by around a factor of five compared to work on door opening with Cartesian space impedance control using the DLR-Lightweight-Robot-II [7]. It also uses joint stiffnesses that are comparable to stiffness estimates for joints in the human arm during planar manipulation [8], [9].

B. The Equilibrium Point Hypothesis

The equilibrium point hypothesis (EPH) inspired our control method. The EPH originates in biomechanical models of the spring-like properties of neuro-muscular systems [10]. It is a well-known hypothesis about biological motor control, which posits that motion is controlled by adjusting the equilibrium point of a biomechanical system over time [10]. These sequences of equilibrium points are sometimes referred to as virtual trajectories [11]. In this context, the equilibrium point refers to the configuration to which the mechanical system would settle in the absence of externally applied forces other than gravity. The EPH has often been presented as a model of biological control that does not require explicit compensation for dynamics.

Long-standing debates continue about whether or not the EPH is true for human motor control [12], [13]. For this work, however, we only use it as inspiration for controlling humanoid robots with equilibrium point control (EPC).

C. Equilibrium Point Control

For EPC, motion of the robot's arm is commanded by adjusting the position of a Cartesian-space equilibrium point (CEP) that denotes where the robot's end effector would settle in the absence of externally applied forces other than gravity. For our implementation, this is achieved through the use of virtual visco-elastic springs at the robot's joints along with gravity compensation. For any commanded CEP, we find an associated joint space equilibrium point (JEP) that defines the equilibrium settings for the virtual springs that would result in the robot's end effector settling at the CEP. The robot could also adjust the stiffnesses of these virtual springs, but we keep them constant in this work.

In previous work, we have shown that behaviors that use linear trajectories for the CEP can operate a door handle and push open a door, and are robust to variations in the control parameters and the environment [14]. We hypothesized that simple CEP trajectories might be sufficient to perform a wide variety of tasks. In this paper, we build on that work by showing that a humanoid robot can use similar CEP trajectories to pull open a variety of mechanisms without detailed prior models.

Equilibrium point control offers a promising alternative to other forms of control. As we demonstrate, it can be effectively used for both freespace reaching trajectories and mechanically constrained manipulation tasks. In contrast to position control methods, our implementation does not require high-fidelity estimates of the kinematics of the handle and door [15]. Unlike some approaches to force control and impedance control, we do not explicitly model the dynamics of the arm nor the impedance at the end effector [7]. We also do not use inverse dynamics [16]. As such, equilibrium point control is relatively simple to use.

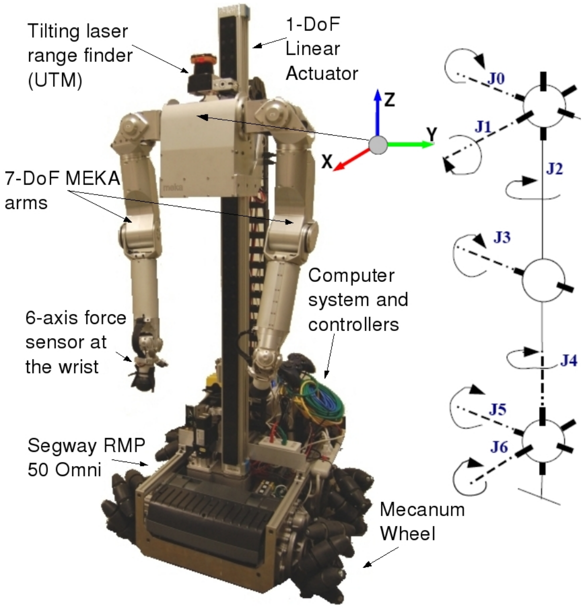
Tilting laser range finder (UTM)
1-DoF Linear Actuator
Z
Y
X
J0
J1
J2
J3
J4
J5
J6
7-DoF MEKA arms
Computer system and controllers
6-axis force sensor at the wrist
Segway RMP 50 Omni
Mecanum Wheel

Fig. 1. The mobile manipulator used in this paper, the coordinate frame attached to the torso, and the orientation of joint axes for the 7 joints of each arm (taken from MEKA Robotics datasheets).

Previous robotics research has looked at similar robotic control strategies in simulation [17], in freespace motions [18], in legged locomotion [19], in rhythmic manipulation from a fixed based [20], and in the design and control of compliant actuators [21], [22]. However, few researchers have looked at this form of control in the context of task-oriented mobile manipulation. Coupled with our robot's low mechanical impedance, we have found EPC to be easy to work with, easy to implement, and surprisingly robust.

II. The Robot

The robot is a new, as yet unnamed, statically stable mobile manipulator that our lab, the Healthcare Robotics Lab, assembled in early 2009 (see Figure 1). It consists of arms from MEKA Robotics (MEKA A1), an omni-directional mobile base from Segway (RMP 50 Omni), and a 1-DoF linear actuator from Festo that can lift the manipulator and sensors from ground level to 1.2m above the ground. Distinctive features of this robot include the use of series elastic actuators [6] in all 14 DoF of the two arms (7 DoF each) and four Mecanum wheels for the base.

For this work, a hook serves as the end effector (Figure 2). We designed the hook, printed it with a 3D printer, and then applied rubber to its surfaces to increase friction. One can think of this as a model of the human hand when a person uses a finger or fingers to hook around a handle and pull something open (Figure 2). We also took inspiration

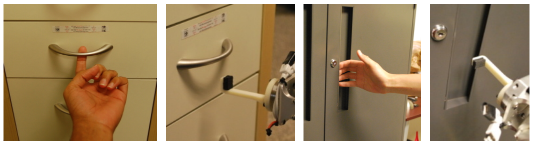

Fig. 2. **Left:** Examples of a human using his hand as a hook. **Right:** The two orientations for the robot's hook, Left and Up.

from prosthetic hooks, which have been successfully used with remarkable versatility and effectiveness. A hook has the advantage of being effective for a variety of handles, including recessed handles that would be difficult to grasp.

A. The Software and the Sensors

A Mac Mini running Ubuntu GNU/Linux performs all of the computation for sensing and high-level control. There is also a Dell Studio Hybrid that runs Ubuntu GNU/Linux with a kernel patched with RTAI for real-time operation. It performs computations for the MEKA arms. We have written all our software in Python and make use of a variety of open source packages including SciPy [23], KDL, ROBOOP, and ROS [24].

For this work, the robot only uses haptic and proprioceptive feedback. The robot senses forces and torques using a wrist-mounted 6-axis force/torque sensor (ATI Mini40 from ATI Industrial Automation). The arm's joints also sense torque, but the current behaviors only use this sensing implicitly in the context of virtual spring control.

III. EQUILIBRIUM POINT CONTROL

Figure 3 shows the control structure of the system presented in this paper. All the manipulation behaviors in this work move the Cartesian equilibrium point (CEP) in a coordinate frame attached to the torso of the robot with the X axis pointing out the front of the robot, the Y axis pointing to the robot's left, and the Z axis pointing up towards the ceiling, as shown in Figure 1.

We have written the behaviors in Python and they run on a Mac Mini running Ubuntu GNU/Linux. At approximately 10Hz, these behaviors compute a new CEP (x_{eq}) using 6-axis force feedback from the wrist-mounted force/torque sensor (F), and estimates of the mechanism kinematics using the history of end effector positions (x_{ee}).

The behaviors then use the inverse kinematics (IK) solver from KDL[1] to compute a joint-space equilibrium point (JEP, q_{eq}) corresponding to the CEP. When computing q_{eq}, the IK solver is seeded with the previous JEP in the trajectory. If no previous JEP exists, the robot uses a look-up-table to find a configuration of the arm with which to seed the IK solver.

On the Dell Studio Hybrid, a control loop simulates virtual viscoelastic springs for all joints of the manipulator except two wrist joints (J5 and J6 in Figure 1). At 1kHz, this control loop computes a torque vector $\tau = [\tau_0 ... \tau_4]$, which are the torques applied to the three joints in the shoulder (J0, J1, J2), one joint in the elbow (J3), and the wrist roll joint (J4). τ is computed as the sum of two torque vectors:

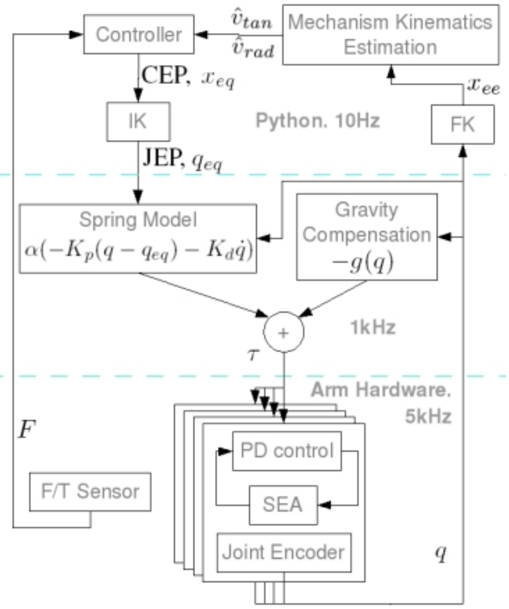

Fig. 3. Block diagram showing the overall control structure.

$$\tau \quad = \quad -g(q) + \alpha(-K_p(q - q_{eq}) - K_d\dot{q}) \qquad (1)$$

The first torque vector, $g(q)$, is the torque due to gravity as a function of the current joint angles q. Subtracting it provides gravity compensation. The second torque vector, $-K_p\tilde{q} - K_d\dot{q}$, simulates a torsional, viscoelastic spring with constant stiffness and damping at each joint. K_p and K_d are diagonal stiffness and damping matrices. q_{eq} is the joint space equilibrium point (JEP), and α is a scalar that we use to increase and decrease the overall stiffness and damping in some of the experiments. Changing α is equivalent to scaling K_p and K_d. As α increases, the arm gets stiffer and behaves more like a position-controlled arm. In practice α can only be scaled within a limited range, such that even with the highest values of α the arm has considerable compliance relative to a traditional industrial arm.

For the wrist joints J5 and J6, the robot uses position control that relates the motor output to joint encoder readings and ignores torque estimates from the deflection of the springs. Consequently the wrist is held stiff, except for the passive compliance of the SEA springs and cables connecting the SEA to the joints. Throughout the experiments, the robot kept its wrist pointed away from the torso and normal to the torso's front surface.

IV. TWO EQUILIBRIUM POINT CONTROLLERS FOR PULLING OPEN DOORS AND DRAWERS

In this section, we describe two controllers that generate Cartesian equilibrium point (CEP) trajectories that enable the robot to pull open novel doors and drawers. Both controllers

[1] Kinematics and Dynamics Library (http://www.orocos.org/kdl)

share a similar structure:

$$\begin{aligned} x_{eq}[t] &= x_{eq}[t-1] + m[t] + h[t] \quad (2)\\ q_{eq}[t] &= \text{Inverse Kinematics}(x_{eq}[t]) \quad (3) \end{aligned}$$

At each time step, t, they compute the CEP, $x_{eq}[t]$, by adding a vector intended to operate the mechanism, $m[t]$, and a vector intended to keep the hook from slipping off of the handle, $h[t]$, to the previous CEP, $x_{eq}[t-1]$. $x_{eq}[t]$ is then used to find $q_{eq}[t]$, the joint space equilibrium point (JEP). $x_{eq}[-1]$ is the CEP once the robot has a firm hook on the handle, by executing the behavior described in Section VI-A. We have written these two controllers in Python and they run at approximately 10Hz.

While updating q_{eq}, the robot looks for three types of stop conditions. If the magnitude of the force measured by the wrist force-torque sensor exceeds a maximum force threshold $F_{th}[t]$, the robot stops. Likewise, if the magnitude of the force drops below 1N, the robot assumes that its hook end effector has slipped off the handle and stops. Finally, if the end effector's trajectory or the CEP trajectory leaves the workspace of the arm, the robot stops. Under some circumstances, the CEP trajectory should be allowed to leave the workspace, but we do not address these situations in this work.

Equations 4 and 5 show how the robot computes the maximum force threshold, $F_{th}[t]$.

$$\begin{aligned} t_{mv} &= min\{t \text{ s.t. } \|x_{ee}[t] - x_{ee}[0]\| \geq 0.1m\} \quad (4)\\ F_{th}[t] &= \begin{cases} 80N & t \leq t_{mv} \\ min(\|F[t_{mv}]\| + 30N, 80N) & \text{otherwise} \end{cases} \quad (5) \end{aligned}$$

It is initialized to 80N. The robot first detects motion of the mechanism at time t_{mv}, when the end effector has moved by a distance greater than the length of the hook. After this, it adapts the maximum force threshold for stopping to be 30N greater than the magnitude of the force measured using the wrist force torque sensor at time step t_{mv}.

A. Controller 1: Pull-Linear

Equations 6 and 7 show the two vectors $m[t]$ and $h[t]$ for this controller. $m[t]$ is a constant vector which results in a linear CEP trajectory and moves the CEP towards the robot's torso by 1cm at each time step. An example of the resulting linear CEP trajectory is shown in Figure 4. $x_{eq}[-1]$ is determined by the behavior that attempts to get a firm hook on the handle (Section VI-A). This is the reason why there is an initial offset between the CEP and the actual position of the end effector in Figure 4. The controller is:

$$\begin{aligned} m[t] &= (-1cm, 0, 0) \quad (6)\\ h[t] &= (0, 0, 0) \quad (7) \end{aligned}$$

B. Controller 2: Pull-Radial-Force

This controller assumes that the mechanism will either be a prismatic joint or a rotary joint with the axis of rotation parallel to gravity. It uses an estimate of the location of the axis of rotation and the radius of the mechanism using the algorithm detailed in Section V.

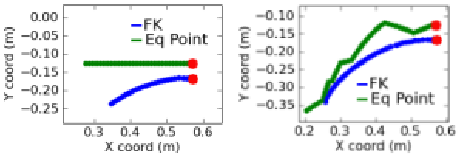

Fig. 4. Figure showing the CEP trajectory in green and the actual motion of the end effector in blue for the Pull-Linear **(left)** and Pull-Radial-Force **(right)** controllers. The red circles denote the CEP and the end effector position at $t = 0$.

Based on this estimate, the controller defines tangential and radial unit vectors for the estimated rotary motion, $(\hat{v}_{tan}[t], \hat{v}_{rad}[t])$. This is comparable to estimating a task frame [1], [25], [26]. Starting at $t = 0$, and until the mechanism kinematics estimation algorithm of Section V does not have enough points to estimate the kinematics, the Pull-Radial-Force controller sets these vectors to $((-1, 0, 0), (0, 1, 0))$.

Using the tangential and radial unit vectors, the controller factors the force measured by the wrist force torque sensor into estimated tangential and radial components, $(\hat{F}_{tan}[t], \hat{F}_{rad}[t])$.

$m[t]$ is a vector of constant magnitude oriented in the direction of the estimated motion tangent (see Equation 8). By itself, $m[t]$ would tend to create an CEP trajectory that looks similar to the trajectory traced out by the handle of the mechanism. $h[t]$ is a vector of constant magnitude that is parallel to the radial unit vector, $\hat{v}_{rad}[t]$. The sign of the vector is determined by a bang-bang controller that attempts to keep the radial force applied to the handle by the hook at 5N, as shown in Equation 9. The controller is:

$$\begin{aligned} m[t] &= 1cm \cdot \hat{v}_{tan}[t] \quad (8)\\ h[t] &= \begin{cases} -0.25cm \cdot \hat{v}_{rad}[t] & \text{if } \hat{F}_{rad}[t] < 5N \\ +0.25cm \cdot \hat{v}_{rad}[t] & \text{if } \hat{F}_{rad}[t] > 5N \\ (0,0,0) & \text{otherwise} \end{cases} \quad (9) \end{aligned}$$

An example of the resulting CEP trajectory is shown in Figure 4.

V. Mechanism Kinematics Estimation

The mechanism kinematics estimation algorithm returns an estimate of the location of the axis of rotation of the mechanism that the robot is operating and its radius. It initially computes two models using the end effector trajectory in the XY plane (via forward kinematics). One model is for rotary mechanisms that open to the right and the second for rotary mechanisms that open to the left. The algorithm does not have a separate model for prismatic joints and we assume that a prismatic joint will be estimated as a rotary joint with a very large radius. The algorithm runs an optimization for each model to compute:

$$(c, r)^* = argmin(\sum_t (\|x_{ee}[t] - c\| - r)^2) \quad (10)$$

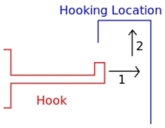

Fig. 5. Figure illustrating the two compliant motions that the robot executes to try to get a firm hooking grasp: 1) motion towards the mechanism. 2) lateral motion towards the handle.

where r is the radius of the mechanism, c is the location of its axis in the XY plane, and $x_{ee}[t]$ is the end effector trajectory. The initial guess for r is set to 1.0m and the initial guess for c is ($x_{ee}[0]_x$,$-r$) and ($x_{ee}[0]_x$,r) for optimizations for door models that open to the right and left respectively. $x_{ee}[0]_x$ is the x coordinate of the end effector at $t = 0$.

We use an implementation of the BFGS optimization algorithm from SciPy [23]. Out of the two circle models, the kinematics estimation algorithm selects the model with the lower residual error as the current estimate of the mechanism's kinematics. This gives new estimates at approximately 5Hz.

VI. Sources of Variation

The two pulling controllers depend on several factors: the way in which the handle has been hooked, the stiffness settings of the joints, the initial posture of the arm, and the pose of the body relative to the handle. Within this section, we describe how we set these parameters for the experiments.

A. A Behavior for Hooking the Handle

First, performance depends on the way in which the robot has hooked the handle. For example, if the end effector slips off the handle, or along the handle, the performance of the system will degrade. In order to represent the natural variations that might occur when autonomously hooking a handle, we created a hooking behavior.

Since we only use haptic sensing in this work, we do not address the problem of finding a good place to hook onto a door or drawer. Instead, we provide the robot with a 3D location to which the hook should be moved and the orientation for the hook (see Figure 2). For the experiments, we achieved this by fixing the arm posture, orienting the hook, and then driving the base and elevating the torso with a gamepad such that the hook was poised to haptically reach forward and hook the handle.

Once the robot is in position, it attempts to firmly hook the handle before executing a pull behavior. It does this through two compliant motions. First, it moves the arm with a linear CEP trajectory along the positive X axis until it detects contact with the mechanism using the wrist force torque sensor. The robot then moves the end effector laterally toward the handle with a linear CEP trajectory until either a force threshold is achieved, indicating contact, or the hook has moved a distance of 5cm. These two motions are illustrated in Figure 5.

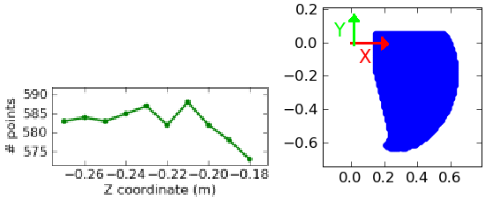

Fig. 6. **Left:** The area of the workspace in a plane parallel to the ground, estimated as the number of points in a grid of 2cm resolution that have an inverse kinematics solution, is reasonably constant when the Z-coordinate is between -0.18m and -0.27m. **Right:** The workspace of the manipulator when the Z-coordinate in -0.21m

B. Selecting the Stiffness Values

For all of the experiments, we used the same relative stiffnesses of the five joints, and only altered the scaling factor α. With α set to 1.0 and the robot's right arm held to its side, the stiffness gains for the virtual viscoelastic springs for J0, J1, and J2 at the shoulder were 20, 50, and 15Nm/rad, respectively. For J3 at the elbow the stiffness gain was 25Nm/rad, and for J4 at the wrist it was 2.5Nm/rad (see Figure 1). Qualitatively, we found that the robot's performance was insensitive to changes in the relative stiffness, as was the computationally estimated stiffness ellipse at the end effector over the end effector's workspace. These stiffness settings are similar to what we used in our previous work on unlatching and pushing doors open [14]. They are also comparable to estimates of human joint stiffness [8], [9].

C. The Pose of the Body Relative to the Handle

During autonomous activities, the position of the base and torso relative to the handle will be likely to vary due to uncertainties (e.g., perception), limited precision (e.g., motion of the base), task constraints (e.g., obstacles), and other challenges that accompany real-world operation. We wish to verify empirically that our proposed controllers are robust to these forms of variation.

1) Height of the Torso: Qualitatively, we found that system performance was not sensitive to the height of the torso relative to the handle. Consequently, we chose to fix the height of the torso relative to the handle. We chose this height by searching for a value that would maximize the area of the planar Cartesian workspace of the end effector. We estimated the size of the workspace by using the IK solver to sample over achievable end effector positions. A graph of the results of this optimization, Figure 6, confirms that the area of the workspace does not vary significantly with height.

2) Planar Location and Orientation: To test the performance of the controllers with variation in the position and orientation of the base relative to the handle of the mechanism we carried out multiple trials on the three mechanisms. We varied the angle over 30° (±15° of error) and the distance by 10cm, as shown in Figure 8. The results of these trials are detailed in Section VII.

In a second set of trials, we selected the position and orientation of the robot from the first set of trials that resulted

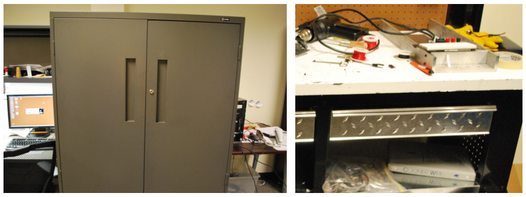

Fig. 7. The three mechanisms used for the experiments of Section VII-A. **Left:**Two cabinet doors with recessed handles. **Right:** Drawer with a recessed handle.

in the maximum opening angle and carried out one trial each on 12 different mechanisms. Section VIII presents the performance of the robot in these trials.

For all the trials, the robot initially pulled towards its torso, as though it believed the initial motion should be normal to its torso.

D. The Initial Posture of the Arm

Given the fixed pose of the wrist, the height of the torso, and the planar position and orientation of the base, the arm must reach the handle such that it is hooking it. For our tests, this leaves one DoF remaining in the 7DoF arm. We initialize the posture of the arm such that the plane formed by the shoulder, elbow and wrist is close to vertical with the elbow tilting away from the torso.

VII. Testing The Two Controllers on Three Mechanisms

To test the performance of the controllers with variation in the position and orientation of the base relative to the handle of the mechanism and with variation in the stiffness of the manipulator, we evaluated the two controllers when operating the left and right doors of a cabinet with rotary joints, and a drawer with a prismatic joint, shown in Figure 7. During these tests, we varied the pose of the robot's base relative to the handle and the overall stiffness of the arm.

A. Comparison of Pulling Controllers

In this section, we report on 108 trials that we conducted to empirically compare the performance of the Pull-Linear and Pull-Radial-Force equilibrium point controllers. For these trials, we systematically varied the mechanism being operated, the pose of the base, and the overall stiffness of the arm. For each of the three mechanisms, we tested both controllers with the base in six different poses and the arm's stiffness scaled by three different values of α (i.e., $3 \times 2 \times 6 \times 3 = 108$ trials). Figure 7 shows the three mechanisms. Figure 8 shows the six poses of the base. We set α to 0.8, 1.0, and 1.2.

Table I summarizes the performance of the two pulling controllers subject to these variations. Both Pull-Linear and Pull-Radial-Force opened all three mechanisms through an average angle greater than 30° for the rotary joints and an average distance greater than 20cm for the prismatic joint. Also, the robot correctly distinguished between rotary and prismatic joints and estimated the correct direction of rotation with both of the controllers. For this work, we interpret a radius greater than 2m to be a prismatic joint.

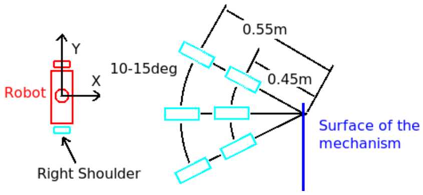

Fig. 8. Layout of the experiments of Section VII-A. We placed the robot in approximately the 6 different positions shown in the figure with an angle between -15° and +15° with the surface of the mechanism and the X-coordinate of the end effector being either 0.45m or 0.55m.

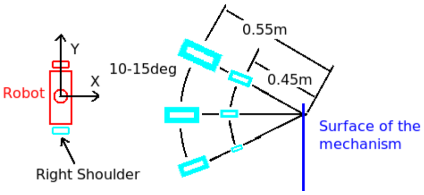

Fig. 9. Figure showing the variation of the opening angle for a rotary joint with the position of the robot. The larger the size of the light blue rectangle, the greater the angle through which the robot opened the door for the experiments of Section VII-A.

For the door that opens to the right, both of the control methods obtained a good estimate of the radius of the joint with the average estimated radius being 0.33m for Pull-Linear (13.2% relative error) and 0.357m for Pull-Radial-Force (6% relative error). The radius of the cabinet door, measured by hand, is 0.38m. The error and the standard deviation of the estimated radius for Door 2 (which opens to the left) are quite high. This is because the end effector tended to slip off the handle, resulting in the termination of the CEP trajectory. As a result, the trajectory from forward kinematics was not indicative of the kinematics of the mechanism.

The average of the maximum interaction force is higher for Pull-Linear as compared to Pull-Radial-Force. This is expected since Pull-Radial-Force actively adjusts the CEP via a bang-bang controller to keep $\hat{F}_{rad}$, the component of the force along the radial direction, at 5N. In this work, the mobile base is stationary while the robot manipulates the mechanism. Due to this, and the workspace limits of the robot arm, the position of the robot relative to the mechanism had a significant impact on the angle through which the robot opened a door, and the distance through which it pulled a drawer. The effect of base position on the extent to which the robot opened a mechanism is shown in Figure 9.

The performance of the two controllers was similar for the prismatic joint. Scaling the stiffness and damping to 0.8, 1.0, and 1.2 did not result in a significant change in the task performance.

VIII. Testing One Controller on 12 Mechanisms

For this experiment, we used the Pull-Radial-Force controller on 12 different mechanisms: four drawers, five doors that open to the right, and three doors that open to the left. We selected the position of the robot relative to the mechanism that resulted in the maximum opening angle for

TABLE I

COMPARISON OF THE TWO PULLING CONTROLLERS.

Mechanism		Pull-Linear	Pull-Radial-Force
Door 1 (Opens right)	Avg max force	20.5N	9.0N
	Std max force	4.9N	1.8N
	Max angle opened	41.6°	79.0°
	Avg angle opened	32.8°	47.2°
	Std angle opened	4.8°	16.4°
	Avg estimated radius	0.331m	0.357m
	Std estimate radius	0.028m	0.018m
Door 2 (Opens left)	Avg max force	9.5N	7.6N
	Std max force	4.0N	3.0N
	Max angle opened	43.4°	42.0°
	Avg angle opened	31.2°	30.6°
	Std angle opened	7.4°	4.2°
	Avg estimated radius	0.680m	0.600m
	Std estimate radius	0.397m	0.322m
Drawer	Avg max force	15.8N	14.1N
	Std max force	2.3N	1.4N
	Max distance opened	0.290m	0.340m
	Avg distance opened	0.223m	0.243m
	Std distance opened	0.051m	0.056m

TABLE II

PERFORMANCE OF PULL-RADIAL-FORCE ON 12 MECHANISMS.

Mechanism	Angle/Distance pulled	Estimated Radius	Measured Radius
Right Door 1	39.0°	0.51m	0.57m
Right Door 2	84.6°	0.30m	0.34m
Right Door 3	84.4°	0.37m	0.40m
Right Door 4	20.0°	0.67m	0.80m
Right Door 5	50.0°	0.89m	0.17m-0.44m
Left Door 1	54.1°	0.42m	0.17m-0.44m
Left Door 2	30.0°	0.60m	0.57m
Left Door 3	38.1°	0.54m	0.41m
Drawer 1	0.3m	45.63m	N/A
Drawer 2	Failed	Failed	N/A
Drawer 3	0.3m	9.69m	N/A
Drawer 4	0.3m	7.34m	N/A

the rotary joints in the experiments of Section VII-A (see Figure 9), and we set $\alpha = 1.0$.

Table II presents the performance of the robot on each mechanism and Figure 10 shows the robot after it has operated each of the 12 mechanisms. The order of the mechanisms in Table II and Figure 10 is consistent. The image of the drawer without the robot is the only mechanism that the robot failed to open. While attempting to pull open this drawer, the hook slipped off before the drawer opened. This drawer initially requires a very large force to open ($> 50N$), which implies that $h[t]$ should apply higher forces to keep the hook on the handle. The robot correctly distinguished between prismatic and rotary joints and the direction of rotation in the remaining 11 trials.

Figure 11 shows the relative error for each of the eight rotary mechanisms. There were larger errors in the mechanism kinematics estimation when the robot operated the mechanisms through smaller angles (such as Right Door 4), or if the hook slid along the handle (such as Right Door 5 and Left Door 1). In our experiments, small opening angles occurred either on doors that opened to the left or on doors with large radii. Currently, the robot only uses its right arm and keeps its mobile base stationary. As a result, it can open doors that open to the left and doors with large radii through a smaller angle as compared to doors that open to the right.

The robot estimated the radius of the rotary joints with an average relative error of 16.3%. This average error ignores a misestimation of the radius for the cabinet shown in Figure 11. This unusual cabinet has its handle horizontally oriented on top, which gives a lot of room for the hook to slide along the handle in the XY plane. When opening the cabinet, the hook slid along the handle, which our current methods for kinematic estimation do not accommodate. Consequently, although the robot correctly categorized the door as being a rotary joint that opens to the right, the robot misestimated the radius as being 89cm, while hand measurement reported it to be between 17cm and 44cm.

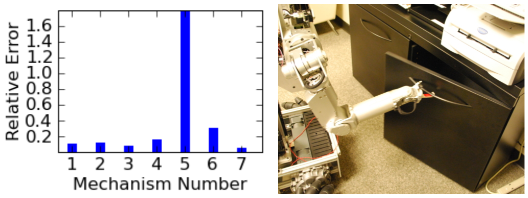

Fig. 11. **Left:** Relative error in the radius estimation for the eight rotary mechanisms. **Right:** A wide horizontal handle, coupled with rotary motion in the XY plane resulted in the hook slipping along the handle as the robot opened this cabinet. As a result, the trajectory from forward kinematics was not informative about the radius of the joint and the location of its axes.

IX. DISCUSSION AND CONCLUSION

Our results indicate that manipulation with low mechanical impedance and equilibrium point control can enable a robot to open a number of different novel doors and drawers. Even with a simple linear trajectory, the robot was able to operate the mechanisms and make inferences about them, so we would expect for a variety of equilibrium point controllers to succeed. Furthermore, we believe that this work provides additional support for our conjecture that manipulation with low mechanical impedance and equilibrium point control is sufficient for a broad set of real-world tasks.

At the same time, the Pull-Radial-Force controller provides an example of how a more sophisticated equilibrium point controller can improve performance. In our tests, it was superior to Pull-Linear, both in terms of the greater extent to which the mechanisms were opened and the lower resultant forces measured at the end effector. Further exploration of the space of possible controllers seems warranted. More generally, efforts to compare the performance of equilibrium point control with more common forms of force and impedance control could help elucidate the advantages and disadvantages of these related approaches.

The success of our methods and the accompanying kinematic inference also suggests that the use of simple compliant trajectories could be valuable for haptically exploring and learning about the world. We would expect this type of approach to be complementary to recent work on kinematic inference using other sensory modalities [27], [28].

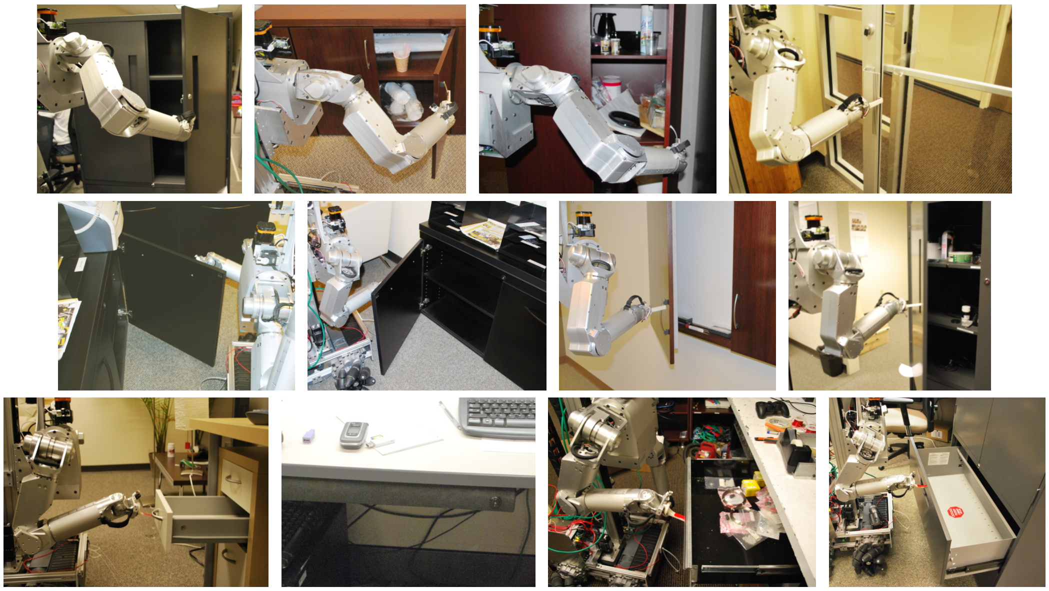

Fig. 10. Images showing the robot after it has operated 11 mechanisms with the Pull-Radial-Force controller. The image of the drawer without the robot is the mechanism for which the hook slipped off before the drawer opened. The order of the mechanisms matches the results presented in Table II

X. Supplementary Material

Supplementary material including the Python code, video and a CAD model of the hook is available at: `www.hsi.gatech.edu/hrl/epc-humanoids09.shtml`

XI. Acknowledgements

We thank Chad Jenkins and Mike Stilman for helpful discussions during this research. We gratefully acknowledge support from Willow Garage and NSF grant IIS-0705130.

References

[1] M. Prats, S. Wieland, T. Asfour, A. del Pobil, and R. Dillmann, "Compliant interaction in household environments by the Armar-III humanoid robot," in *Humanoids*, 2008.

[2] R. Diankov, S. Srinivasa, D. Ferguson, and J. Kuffner, "Manipulation Planning with Caging Grasps," in *Humanoids*, 2008.

[3] G. Pratt, "Low impedance walking robots 1," *Integrative and Comparative Biology*, vol. 42, no. 1, pp. 174–181, 2002.

[4] S. Buerger, "Stable, High-Force, Low-Impedance Robotic Actuators for Human-Interactive Machines," Ph.D. dissertation, Massachusetts Institute of Technology, 2006.

[5] M. Zinn, O. Khatib, B. Roth, and J. Salisbury, "Playing it safe [human-friendly robots]," *IEEE Robotics & Automation Magazine*, 2004.

[6] G. Pratt and M. Williamson, "Series elastic actuators," in *IROS*, 1995.

[7] C. Ott, B. Baeuml, C. Borst, and G. Hirzinger, "Autonomous opening of a door with a mobile manipulator: A case study," *IFAC Symposium on Intelligent Autonomous Vehicles*, 2007.

[8] H. Gomi and M. Kawato, "Equilibrium-point control hypothesis examined by measured arm stiffness during multijoint movement," *Science*, vol. 272, no. 5258, p. 117, 1996.

[9] F. Mussa-Ivaldi, N. Hogan, and E. Bizzi, "Neural, mechanical, and geometric factors subserving arm posture in humans," *Journal of Neuroscience*, vol. 5, no. 10, pp. 2732–2743, 1985.

[10] R. Shadmehr, "Equilibrium point hypothesis. In (Ed.), M. AA, editor, Handbook of brain theory and neural networks," 2002.

[11] N. Hogan, "The mechanics of multi-joint posture and movement control," *Biological cybernetics*, vol. 52, no. 5, pp. 315–331, 1985.

[12] E. Bizzi, N. Hogan, F. Mussa-Ivaldi, and S. Giszter, "Does the nervous system use equilibrium-point control to guide single and multiple joint movements?" *Behavioral and brain sciences(Print)*, 1992.

[13] M. Hinder and T. Milner, "The case for an internal dynamics model versus equilibrium point control in human movement," *The Journal of Physiology*, vol. 549, no. 3, pp. 953–963, 2003.

[14] A. Jain and C. C. Kemp, "Behavior-based door opening with equilibrium point control," in *RSS Workshop: Mobile Manipulation in Human Environments*, 2009.

[15] M. Quigley, S. Batra, S. Gould, E. Klingbeil, Q. Le, A. Wellman, and A. Y. Ng, "High-Accuracy 3D Sensing for Mobile Manipulation: Improving Object Detection and Door Opening," in *ICRA*, 2009.

[16] L. Sentis and O. Khatib, "Synthesis of whole-body behaviors through hierarchical control of behavioral primitives," *International Journal of Humanoid Robotics*, 2005.

[17] X. Gu and D. Ballard, "An equilibrium point based model unifying movement control in humanoids," in *RSS*, 2006.

[18] M. Williamson, "Postural primitives: Interactive behavior for a humanoid robot arm," in *Proceedings of the Fourth International Conference on Simulation of Adaptive Behavior*, 1996.

[19] S. Migliore, "The Role of Passive Joint Stiffness and Active Knee Control in Robotic Leg Swinging: Applications to Dynamic Walking," Ph.D. dissertation, Georgia Institute of Technology, 2009.

[20] M. Williamson, "Robot arm control exploiting natural dynamics," Ph.D. dissertation, Massachusetts Institute of Technology, 1999.

[21] Y. Mukaibo, S. Park, and T. Maeno, "Equilibrium Point Control of a Robot Arm with a Double Actuator Joint," *International Simposium on Robotics and Automation*, 2004.

[22] D. Clapa, E. Croft, and A. Hodgson, "Equilibrium point control of a 2-DOF manipulator," *Journal of Dynamic Systems, Measurement, and Control*, vol. 128, p. 134, 2006.

[23] E. Jones, T. Oliphant, P. Peterson, *et al.*, "SciPy: Open source scientific tools for Python," *URL http://www. scipy. org*, 2001.

[24] M. Quigley, B. Gerkey, K. Conley, J. Faust, T. Foote, J. Leibs, R. W. Eric Berger, and A. Ng, "ROS: an open-source Robot Operating System," in *Open-Source Software workshop of (ICRA)*, 2009.

[25] M. 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] H. Bruyninckx and J. De Schutter, "Specification of force-controlled actions in the "Task Frame Formalism": A Synthesis," *Robotics and Automation, IEEE Transactions on*, vol. 12, no. 4, pp. 581–589, 1996.

[27] D. Katz, Y. Pyuro, and O. Brock, "Learning to Manipulate Articulated Objects in Unstructured Environments Using a Grounded Relational Representation," *Robotics Science and Systems*, 2008.

[28] J. Sturm, V. Pradeep, C. Stachniss, C. Plagemann, K. Konolige, and W. Burgard, "Learning kinematic models for articulated objects," in *Proc. of the Int. Conf. on Artificial Intelligence (IJCAI)*, 2009.