



Reaching in clutter with whole-arm tactile sensing

The International Journal of
Robotics Research
32(4) 458–482



© The Author(s) 2013
Reprints and permission:
sagepub.co.uk/journalsPermissions.nav
DOI: 10.1177/0278364912471865
ijr.sagepub.com



Advait Jain^{1*}, Marc D Killpack^{1*}, Aaron Edsinger² and Charles C Kemp¹

Abstract

Clutter creates challenges for robot manipulation, including a lack of non-contact trajectories and reduced visibility for line-of-sight sensors. We demonstrate that robots can use whole-arm tactile sensing to perceive clutter and maneuver within it, while keeping contact forces low. We first present our approach to manipulation, which emphasizes the benefits of making contact across the entire manipulator and assumes the manipulator has low-stiffness actuation and tactile sensing across its entire surface. We then present a novel controller that exploits these assumptions. The controller only requires haptic sensing, handles multiple contacts, and does not need an explicit model of the environment prior to contact. It uses model predictive control with a time horizon of length one and a linear quasi-static mechanical model. In our experiments, the controller enabled a real robot and a simulated robot to reach goal locations in a variety of environments, including artificial foliage, a cinder block, and randomly generated clutter, while keeping contact forces low. While reaching, the robots performed maneuvers that included bending objects, compressing objects, sliding objects, and pivoting around objects. In simulation, whole-arm tactile sensing also outperformed per-link force-torque sensing in moderate clutter, with the relative benefits increasing with the amount of clutter.

Keywords

Manipulation, force and tactile sensing, mechanics, design and control

1. Introduction

Research on robot manipulation has often emphasized collision-free motion with occasional contact restricted to the robot's end effector. In essence, most of the manipulator's motion is intended to be free-space motion and unintended contact is considered to be a failure of the system. In contrast, animals often appear to treat contact between their arms and the world as a benign event that does not need to be avoided. For example, humans make extensive contact with their forearms even during mundane tasks, such as eating or working at a desk.

In this paper, we present progress towards new foundational capabilities for robot manipulation that take advantage of contact across the entire arm. Our primary assumption is that, for a given robot, environment, and task, contact forces below some value have no associated penalty. For example, when reaching into a bush, moderate contact forces are unlikely to alter the robot's arm or the bush in undesirable ways. Likewise, even environments with fragile objects, such as glassware on a shelf, can permit low contact forces. While some situations merit strict avoidance of contact with an object, we consider these to be rare, and instead focus on control methods that allow contact.

Our main contribution in this paper is a novel controller that enables a robot arm to move within an environment while regulating contact forces across its entire surface. The controller assumes that contact forces can be sensed across the surface of the arm, and that the arm's joints can be modeled as linear torsional springs. The controller uses model predictive control (MPC) with a time horizon of length one and a linear quasi-static mechanical model. At each time-step, the controller constructs a model and solves an associated quadratic programming problem that minimizes the predicted distance to a goal, subject to constraints on the predicted changes in contact forces.

We empirically evaluated our controller's performance with respect to the task of haptically reaching to a goal location in high clutter (see Figure 1). The clutter could consist of a variety of fixed, movable, and deformable objects, and the robot was not given a model of the environment. For our

¹Healthcare Robotics Lab, Georgia Institute of Technology, USA

²Meka Robotics, San Francisco, USA

*These authors contributed equally to this work.

Corresponding author:

Advait Jain, Healthcare Robotics Lab, Georgia Institute of Technology,
828 West Peachtree Street NW, Atlanta, GA 30308, USA.
Email: advaitjain@gmail.com



Fig. 1. (Left) View of foliage from the robot’s perspective. Two rigid blocks of wood are occluded by the leaves. **(Right)** Image of the robot after it has successfully reached the goal location using the controller we present in this paper. The red circle denotes the position of the end effector.

tests, we used a simulated robot, a real robot with simulated tactile sensing, and a real robot with real tactile sensing across its forearm. These robots successfully reached to locations in clutter while regulating contact forces across their arms. While doing so, they autonomously performed maneuvers, such as pushing into a compliant object until the contact force was too high and then pivoting around it. In an automated test with artificial foliage, our controller outperformed a baseline controller in terms of success rate and low contact forces. In simulation, our controller had higher performance with whole-arm tactile sensing than with per-link force–torque sensing in moderate clutter, and the relative value of whole-arm tactile sensing increased with higher clutter.

1.1. Biological inspiration

Animals have inspired our approach with their adept manipulation in cluttered environments. As illustrated in Figure 2, animals reach for objects in high clutter with poor visibility when foraging for food (Dominy, 2004; Iwaniuk and Whishaw, 1999; Catania, 1999). While doing so, they are able to handle contact at multiple locations across their arms and often appear to treat contact as a benign event.

Whole-body tactile sensing and compliant actuation are notable characteristics of biological organisms. Organisms across spatial scales, from small nematodes to insects and mammals, can sense forces across their entire bodies (Bianchi, 2007; Lumpkin et al., 2010; Lederman and Klatzky, 2009 and <http://www.wormbook.org/>). Compliant actuation is also common and has inspired numerous roboticists (Hogan, 1984; Alexander, 1990; Migliore et al., 2005).

1.2. Benefits of contact, whole-body tactile sensing, and low-stiffness actuation

Allowing contact between the robot and the environment has potential benefits for manipulating in clutter, as does whole-body tactile sensing and low-stiffness actuation.

Allowing contact with the arm can increase the effective range of motion of the manipulator. The decrease in the effective workspace due to avoiding contact is exacerbated by safety margins and an inability to apply forces that compress or move objects. Similarly, if the robot has a compliant exterior, avoiding contact forfeits the additional range of motion achievable by compressing its exterior.

Whole-body tactile sensing has advantages in terms of distinguishing between distinct force distributions and contact configurations. For example, a force distributed over a small area results in higher stress with greater potential for damaging materials than the same force distributed over a large area. Also, the geometries associated with distinct contact regions, such as a rigid point, line, or plane, imply distinct options for movement. Prior research has attempted to use joint torques, the geometry of the links, and measurements from force–torque sensors to estimate contact properties (e.g. Eberman and Salisbury (1990); Bicchi et al. (1993); Kaneko and Tanie (1994); De Schutter et al. (1999)). However, interpretation of data from these sensors can often be ambiguous in multi-contact situations (Salisbury, 1984). In practice, the estimation can also be sensitive to the state of the manipulator, the fidelity of the torque estimates, joint friction, and other properties of the manipulator (Eberman, 1989; Dogar et al., 2010).

Robotics researchers have demonstrated that low-stiffness joints lower interaction forces during incidental contact and can be beneficial for unmodeled interactions (Pratt and Williamson, 1995; Pratt and Pratt, 2001; Pratt, 2002; Buerger and Hogan, 2007). Within rigid clutter, low stiffness at the joints may also mitigate problems due to jamming and wedging, much like compliance can be beneficial in peg-in-hole problems (Whitney and Nevins, 1979; Whitney, 1982; Dupont and Yamajako, 1994; Mason, 2001).

1.3. Challenges associated with reaching in high clutter

For this paper, we focus on reaching to a goal location in high clutter, which entails challenges such as the following:

- *Lack of non-contact trajectories:* As clutter increases, approaches that avoid contact with the environment will have a diminishing set of feasible trajectories.
- *Contact with only the end effector may be inefficient or infeasible:* Removing or rearranging the clutter by serially making contact with the end effector may be inefficient or infeasible.
- *Creating detailed geometric models in advance may be infeasible:* Clutter can consist of unique objects and configurations that have not been encountered before.
- *Observation of geometry is obstructed:* Occlusion can prevent conventional line-of-sight sensors, such as cameras and laser range finders, from observing the geometry of the clutter in advance (see Figure 1).

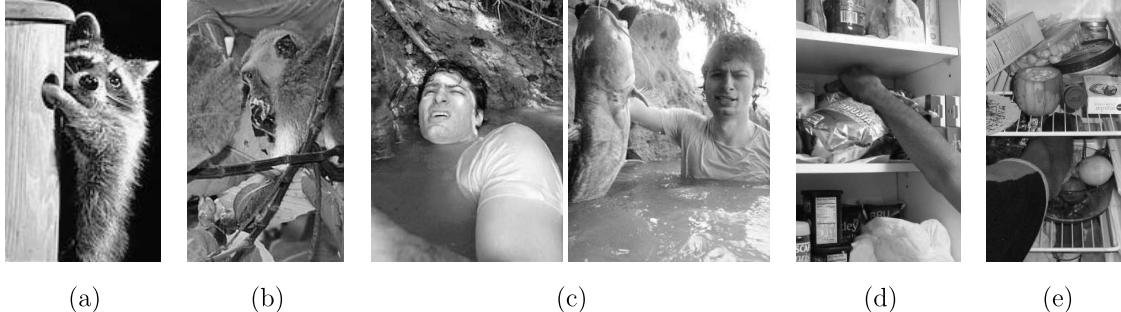


Fig. 2. Animals reach into clutter while foraging. (a) A raccoon reaches into a bird house to find eggs and young (<http://www.nativeamerica.com/research.html>). (b) A Long-tailed Macaque grasps fruit in dense foliage (<http://wildshores.blogspot.com/2009/06/wild-monkeys-at-breakfast-in-admiralty.html>). (c) When noodling, people find catfish holes from which to pull fish out (<http://www.ethantw.com/noodling.html>). (d)–(e) A person makes contact along his forearm while reaching for objects in a cluttered cabinet and refrigerator. (All images used with permission)

- *Mechanics are difficult to infer without contact:* Non-contact sensing provides limited ability to infer the mechanical properties of the clutter, such as whether or not an object can be bent or moved out of the way.

Notably, many approaches to manipulation are poorly matched to the challenges of high clutter. For example, approaches that rely on pre-existing detailed models, estimation of models via conventional line-of-sight sensing, or collision-free motions with the arm would fare poorly in many high-clutter real-world situations (e.g. Stilman et al. (2007b); Kavraki and LaValle (2008); Saxena et al. (2008); Srinivasa et al. (2009)).

In contrast, our approach directly addresses these challenges by allowing multiple contacts across the entire surface of the arm, not requiring a detailed model of the environment prior to contact, and only requiring contact-based sensing.

1.4. Organization of this paper

In Section 2 we discuss related research. In Section 3, we present our model predictive controller. We then describe the three testbeds we used to evaluate our controller (Section 4), the approaches that we used for comparison (Section 5), and the experiments that we ran (Sections 6 and 7). We discuss some limitations in Section 8 and conclude in Section 9.

2. Related work

2.1. Manipulation in clutter

In this paper, we focus on the task of reaching to a goal location in high clutter. Previous robotics research has addressed the tasks of generating collision-free trajectories (e.g. Lozano-Perez (1987); LaValle and Kuffner (2001); Kavraki and LaValle (2008)), generating reaching motions in free space (e.g. Hersch and Billard (2006); Stulp et al. (2009); Metta et al. (2011)), and manipulating objects

in uncluttered environments (e.g. Natale and Torres-Jara (2006); Saxena et al. (2008); Hsiao et al. (2010a); Jain and Kemp (2010a); Pastor et al. (2011); Romano et al. (2011)).

Most prior research on manipulation in clutter restricts contact between the robot and its environment to the end effector. Leeper et al. (2012) presented a method to assist remote operators that find collision-free poses for the end effector in clutter. Dogar and Srinivasa (2011) and Stillman et al. (2007a, 2007b) presented motion planners that use models of objects in the world to enable robots to rearrange clutter by pushing, grasping, and moving objects with their end effectors. The planned trajectories were executed without sensor feedback. Mason et al. (2011) described a simple end effector design that can grasp a single marker from a cluttered pile of markers, and haptically estimate its pose in the gripper. Although the task is different, aspects of the approach are similar in spirit to ours: the robot interacts with high clutter, does not use a detailed model of the environment prior to contact, allows multiple contacts across the surfaces of the end effector, uses haptic sensing, and retries several times.

2.2. Multi-contact manipulation

Park and Khatib (2008) presented a framework for controlling a robot with multiple contacts along the links. It generalized previous direct force control methods (Raibert and Craig, 1981; Khatib, 1987) to allow for contact at points other than the end effector. It also considers the dynamic stability of the robot. This method requires a full dynamic model of the robot, models contact as being rigid, and assumes that the robot has at least six degrees of freedom (DOF) for each contact to control the contact force and torque vector (Sentis et al., 2010). Using this framework, results have been shown in simulation (Sentis and Khatib, 2005; Sentis et al., 2010), and on a real robot in relatively controlled settings (Petrovskaya et al., 2007). No results have been shown in cases where the robot makes additional

unpredicted contact with the environment or loses contact at some locations.

Researchers have demonstrated that a humanoid robot can make contact at multiple, predetermined locations on its body to better achieve dynamic stability while performing tasks, such as taking a large step (Legagne et al., 2011), climbing a ladder (Hauser et al., 2005), and sitting down in a chair (Escande and Kheddar, 2009). These approaches require a complete geometric model of the world, assume rigid contacts, and do not incorporate sensor feedback as the robot executes the planned kinematic trajectory.

Other work on multi-contact manipulation exists in the context of using the surfaces of a multi-fingered hand, or the entire body to grasp and manipulate objects (e.g. Bicchi (1993); Bicchi and Kumar (2000); Platt Jr et al. (2003); Hsiao and Lozano-Perez (2006); Hsiao et al. (2010b)). Researchers in this area often focus on the task of finding and achieving stable grasps for objects.

Research on whole-arm collision detection has used a model of the dynamics of the robot arm to detect deviations from the expected torques to detect arbitrary collisions across the entire arm (e.g. De Luca and Mattone (2004); De Luca et al. (2006); Haddadin et al. (2008)). Contact conditions that result in loads that cancel one another on a link could go undetected with these methods. Joint torques alone are also insufficient to discriminate among multi-contact conditions in general.

2.3. Tactile exploration

Researchers have used tactile sensing at the end effector to haptically explore objects (Okamura and Cutkosky, 2001; Kraft et al., 2009) and environments (Haddadin et al., 2011; Ganesh et al., 2012), detect features (Okamura and Cutkosky, 1999), recognize objects (Bierbaum et al., 2009; Pezzementi et al., 2011), localize objects (Petrovskaya and Khatib, 2011; Platt Jr et al., 2011), and estimate object properties (Tanaka et al., 2003; Chitta et al., 2011).

Most research has looked at tactile sensing at the end effector for haptic exploration, but there are exceptions, such as research that investigates head-mounted whiskers for bio-inspired exploration (Prescott et al., 2010).

2.4. Motion planning with deformable objects

Manipulation research often assumes that any objects that the robot interacts with are rigid. There is some research on motion planners that allow the robot to make contact with, and push into deformable objects (e.g. Rodriguez et al. (2006); Frank et al. (2011); Patil et al. (2011)). These approaches assume knowledge of the specific configuration of the objects and require accurate and detailed models of how the objects deform.

2.5. Robot locomotion

Our approach to robot manipulation has similarities to approaches that have been successful for robot locomotion. For example, researchers have developed robots that locomote in cluttered environments without detailed geometric models of the terrain nor planning over long time horizons (Saranli et al., 2001; Raibert et al., 2008). Likewise, whole-body contact, and contact in general, has not been considered undesirable. For example, robots have used contact over their entire bodies to traverse the ground and swim in granular media (McKenna et al., 2008; Maladen et al., 2010). Additionally, the use of simple mechanical models, compliance, and force sensing is common for robot locomotion (Garcia et al., 1998; Pratt and Pratt, 2001; Pratt, 2002; Raibert et al., 2008).

2.6. Model predictive control

One of the initial application areas for MPC was chemical process control (Garcia et al., 1989). It is often referred to as receding horizon control when used for control of aerial vehicles (Bellingham et al., 2002; Abbeel et al., 2010). MPC has also been used in robot locomotion research (e.g. Wieber (2006); Erez et al. (2011); Manchester et al. (2011)). In terms of robot manipulation, MPC has recently been used for bouncing a ball (Kulchenko and Todorov, 2011), generating manipulator trajectories to compensate for inertial forces on a boat (From et al., 2011), controlling a six-DOF cable-driven parallel manipulator (Duchaine et al., 2007), and reaching in free space (Ivaldi et al., 2010).

3. The model predictive controller

The controller that we have developed uses linear MPC with a time horizon of length one. Specifically, using the notation of Morari and Lee (1999), our controller uses a linear discrete-time model of the system,

$$x(k+1) = Ax(k) + Bu(k), \quad (1)$$

where $x(k)$ is the state and $u(k)$ is the control input.

At each time-step, k , the controller computes a sequence of control inputs, $u^*(i), i = k \dots (k+N-1)$, to minimize a quadratic objective function of $x(k), \dots, x(k+N)$ and $u(k), \dots, u(k+N-1)$, subject to linear inequality constraints on $x(k), \dots, x(k+N)$ and $u(k), \dots, u(k+N-1)$, where N is the length of the time horizon of the model predictive controller. This defines a quadratic program (Morari and Lee, 1999). The controller then uses only the first control input by setting $u(k) = u^*(k)$, and reformulates the quadratic program at the next time-step. In this paper, we use a time horizon of length one ($N = 1$), and recompute the A and B matrices in equation (1) at each time-step.

In the rest of this section, we describe our model predictive controller for manipulation with multiple contacts. First, in Section 3.1, we give an overview of the controller that we have developed. Next, we present the hierarchy

of controllers in Section 3.2. In Section 3.3, we describe the linear quasi-static model that our model predictive controller uses, and detail the quadratic program that the controller solves at each time-step in Section 3.4. We then describe extensions to the quadratic program in Section 3.5.

3.1. Overview of the one-step model predictive controller

Our model predictive controller attempts to move the end effector closer to a specified position, subject to constraints on the predicted contact forces. It explicitly allows the robot to apply any force less than a *don't care force threshold*, $f_{c_i}^{\text{thresh}}$, at each contact, c_i .

Our controller has the following parameters that influence its behavior:

- *Goal location* ($x_g \in \mathbb{R}^3$): This is the Cartesian position to which the controller attempts to move the end effector. We do not address achieving a goal orientation in this paper.
- *Contact stiffness matrices* ($K_{c_i} \in \mathbb{R}^{3 \times 3}$): These are the controller's estimates of the stiffness matrices for each contact location along the arm. In this paper, we assume that the stiffness at each contact is non-zero along the direction of the contact force sensed by the real or simulated sensors and is zero in the other directions.
- *Don't care force thresholds* ($f_{c_i}^{\text{thresh}} \in \mathbb{R}^3$): The controller attempts to keep the force at each contact below this value, and there is no penalty or constraint for contact forces below this threshold.
- *Maximum rate of change for contact forces* ($\Delta f_{c_i}^{\text{rate}} \in \mathbb{R}^3$): This term limits the predicted change in the contact force over one time-step.
- *Safety force threshold* ($f_{c_i}^{\text{safety}} \in \mathbb{R}^3$): If the contact force, f_{c_i} , exceeds this safety threshold value, our controller stops and reports a failure.

3.2. Control structure

Figure 3 shows our hierarchical control structure. A joint-space impedance controller runs at 1 kHz. Our implementation of our model predictive controller runs in Python on a conventional desktop computer at a rate that varies between approximately 50 and 100 Hz. Some of this variation depends on the number of contact locations.

3.2.1. "Simple" impedance controller: For a detailed description and analysis of this form of joint-space impedance control, we refer the reader to Hogan and Buerger (2005). The input to the 1 kHz "simple" impedance controller, ϕ , is called a virtual trajectory. The controller uses feedback from the joint encoders to command torques at the joints, τ , that are given by

$$\tau = K_j(\phi - \theta) - D_j\dot{\theta} + \hat{\tau}_g(\theta). \quad (2)$$

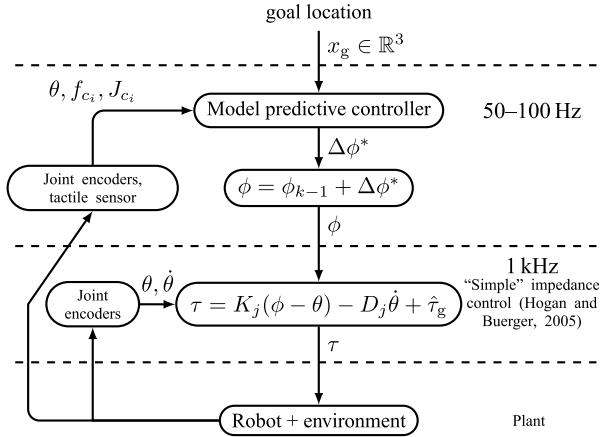


Fig. 3. Block diagram showing the hierarchical control structure. Details are in Section 3. The controller frequencies are specific to our implementation.

For this paper, K_j and D_j are constant $m \times m$ diagonal joint-space stiffness and damping matrices, $\theta \in \mathbb{R}^m$ and $\dot{\theta} \in \mathbb{R}^m$ are the current joint angles and joint velocities, and $\hat{\tau}_g \in \mathbb{R}^m$ is a gravity-compensating torque vector that is a function of θ . The robot arm has m joints.

As a result, the closed-loop system behaves as if the arm is connected to the joint-space virtual trajectory, ϕ , via torsional viscoelastic springs at the joints. If ϕ is held constant, "simple" impedance control can be shown to result in stable interaction with passive environments for contacts all over the arm (Hogan and Buerger, 2005; Hogan, 1988).

"Simple" impedance control, sometimes referred to as equilibrium point control, does not explicitly model the dynamics of the arm, unlike other approaches to force control and impedance control (Albu-Schaffer et al., 2003; Sentis et al., 2010). A model of the dynamics could potentially improve performance. In practice, we have found "simple" impedance control to be stable, reliable, and sufficient to reach into clutter and perform other tasks (Edsinger and Kemp, 2007a, 2007b; Jain and Kemp, 2009a, 2009b, 2010b). It also has the advantage of being straightforward to implement without the need for system identification. Other researchers have looked at similar robotic control strategies for humanoid motion in simulation (Gu and Ballard, 2006) for free-space motion (Williamson, 1996), legged locomotion (Migliore, 2009), and rhythmic manipulation from a fixed base (Williamson, 1999).

Extension 1 is a video showing the "simple" impedance controller running and the low stiffness of the robot's joints.

3.2.2. Model predictive controller: The input to our model predictive controller is a Cartesian goal position, $x_g \in \mathbb{R}^3$. The controller uses feedback from the joint encoders and whole-arm tactile sensing to compute $\Delta\phi^* \in \mathbb{R}^m$, an incremental change in the virtual joint-space trajectory. This $\Delta\phi^*$ is the control input, $u(k)$, of equation (1).

3.3. Linear discrete-time model

In this section, we derive a linear discrete-time quasi-static model, similar to equation (1), for the arm and its interaction with the world. The model is of the form

$$\theta(k+1) = \theta(k) + B\Delta\phi(k), \quad (3)$$

where $\theta \in \mathbb{R}^m$ is the state of the system (vector of joint angles for a robot with m joints), the control input $\Delta\phi \in \mathbb{R}^m$ is the incremental change in the joint-space virtual trajectory of the impedance controller, and $B \in \mathbb{R}^{m \times m}$.

We begin by assuming that the robot has a fixed and statically stable mobile base and the arm is in contact with the world at n locations. We denote the i^{th} contact as c_i . The equations of motion in joint-space are

$$M(\theta)\ddot{\theta} + C(\theta, \dot{\theta})\dot{\theta} + \sum_{i=1}^n J_{c_i}^T(\theta)f_{c_i} + \tau_g(\theta) = \tau, \quad (4)$$

where $M \in \mathbb{R}^{m \times m}$ is the inertia matrix, $C \in \mathbb{R}^{m \times m}$ is the Coriolis matrix, $J_{c_i} \in \mathbb{R}^{3 \times m}$ is the Jacobian matrix for the i^{th} contact, $f_{c_i} \in \mathbb{R}^3$ is the force at contact c_i , $\tau_g \in \mathbb{R}^m$ is the vector of torques due to gravity at each joint, and $\tau \in \mathbb{R}^m$ is the vector of torques applied by the actuators at the joints. Equation (4) ignores properties such as friction at the joints, but is commonly used in robotics (Featherstone and Orin, 2008).

Combining the equations of motion (equation (4)) with the impedance control law (equation (2)) gives the model of the arm and its interaction with the world,

$$M\ddot{\theta} + C\dot{\theta} + \sum_{i=1}^n J_{c_i}^T f_{c_i} + \tau_g = K_j(\phi - \theta) - D_j\dot{\theta} + \hat{\tau}_g. \quad (5)$$

In this paper, as an approximation, we assume that the gravity-compensating torques are perfect and that the dynamics are negligible, since the robot is moving slowly. We remove all terms with $\ddot{\theta}$ or $\dot{\theta}$ from equation (5), and set $\hat{\tau}_g = \tau_g$ to obtain

$$\sum_{i=1}^n J_{c_i}^T f_{c_i} = K_j(\phi - \theta), \quad (6)$$

which is a quasi-static model. These approximations help us obtain a linear discrete time model that results in a computationally favorable quadratic program. In equation (6), the torques at the joints due to the contact forces (left-hand side) balance the torques applied by the actuators in the joints (right-hand side).

For the contact model, we ignore friction at the contacts and assume that each contact behaves like a linear spring normal to the surface of the robot arm. These assumptions are similar to the Hertzian contact model (Johnson, 1985; Kao et al., 2008). This results in a mechanical model with torsional springs at the joints and linear springs at the contacts as illustrated in Figure 4.

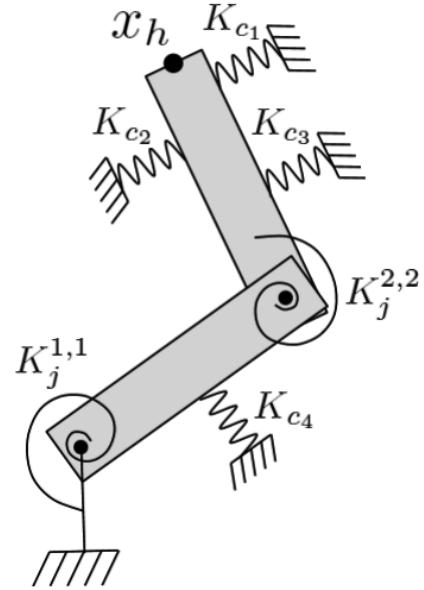


Fig. 4. Graphical representation of a planar version of the quasi-static mechanical model with torsional springs at the joints of the robot and linear springs at contacts that our model predictive controller uses, as we describe in Section 3.3.

If we take the difference between equation (6) at times k and $k+1$, we obtain

$$\begin{aligned} \sum_{i=1}^n J_{c_i}^T(k+1)f_{c_i}(k+1) - J_{c_i}^T(k)f_{c_i}(k) \\ = K_j(\phi(k+1) - \phi(k) - \theta(k+1) + \theta(k)). \end{aligned} \quad (7)$$

We assume that the change in the configuration of the arm in one time step, $\theta(k+1) - \theta(k)$, is small and we approximate $J_{c_i}(k+1)$ with $J_{c_i}(k)$. We also assume that no new contacts occur and no contacts are broken. This reduces equation (7) to

$$\begin{aligned} \sum_{i=1}^n J_{c_i}^T(k)(f_{c_i}(k+1) - f_{c_i}(k)) \\ = K_j(\Delta\phi(k) - \theta(k+1) + \theta(k)), \end{aligned} \quad (8)$$

where $\Delta\phi(k) = \phi(k+1) - \phi(k)$ is the control input of our model predictive controller, see equation (3) and Figure 3.

Using the linear elastic spring model for the contacts,

$$f_{c_i}(k+1) - f_{c_i}(k) = K_{c_i}J_{c_i}\Delta\theta(k), \quad (9)$$

where $\Delta\theta(k) = \theta(k+1) - \theta(k)$. We can now use equation (9) to rewrite equation (8) as

$$\theta(k+1) = \theta(k) + \left(K_j + \sum_{i=1}^n J_{c_i}^T K_{c_i} J_{c_i} \right)^{-1} K_j \Delta\phi(k). \quad (10)$$

$(K_j + \sum_{i=1}^n J_{c_i}^T K_{c_i} J_{c_i})$ is the sum of a positive definite matrix, K_j , and positive semi-definite matrices, $J_{c_i}^T K_{c_i} J_{c_i}$, and is therefore positive definite and invertible.

Equation (10) is the linear discrete-time model of the system that our controller generates and uses at each time-step. The controller computes the parameters for this model, J_{c_i} and K_{c_i} , based on the current contact locations and a stiffness for each contact. Note that the contact forces, f_{c_i} , do not appear in this model, which relates change in the virtual trajectory to change in the joint angles. The contact forces do appear in the inequality constraints of the quadratic program, and are used in a quadratic penalty term if they exceed the *don't care force threshold*, $f_{c_i}^{\text{thresh}}$. In practice, measured forces are also used to detect contact and estimate contact locations. Detecting contact is especially important, since false positives would result in the controller hallucinating contact and modeling it with a spring.

Equation (10) is in the same form as equations (1) and (3). As we describe next, its linear form allows us to formulate the optimization as a quadratic program (QP) that can be solved efficiently at each time-step.

3.4. Quadratic program to compute $\Delta\phi^*$

Using the terminology of Boyd and Vandenberghe (2004), our optimization variable is $\Delta\phi$, an incremental change in the joint-space virtual trajectory, and we minimize a quadratic objective function subject to linear equality and inequality constraints. We use the open source OpenOpt framework to solve this quadratic program (<http://openopt.org>).

Our objective function is of the form

$$\sum_i \alpha_i g_i, \quad (11)$$

where g_i are quadratic functions of the optimization variable $\Delta\phi$ and α_i are empirically tuned scalar weights. We set $\alpha_1 = 1$ and list the values of the other scalar weights in Section 3.5.

We set up the quadratic program such that the solution, $\Delta\phi^*$, will result in the predicted position of the end effector that is closest to a desired position subject to constraints on the predicted change in the joint angles and contact forces. Below are the components of this quadratic program.

3.4.1. Move to a desired position: The first term of the quadratic objective function of equation (11) attempts to move the end effector to a desired position. It is of the form

$$g_1 = \|\Delta x_d - \Delta x_h\|^2, \quad (12)$$

where $\Delta x_h = x_h(k+1) - x_h(k)$ is the predicted motion of the end effector (hand) and $\Delta x_d \in \mathbb{R}^3$ is the desired change in the end effector position in one time-step.

We assume the change in the joint angles will be small, and use the kinematic relationship

$$\Delta x_h = J_h \Delta\theta, \quad (13)$$

where $J_h \in \mathbb{R}^{3 \times m}$ is the Jacobian at the end effector (or hand), and $\Delta\theta = \theta(k+1) - \theta(k)$ is the change in the joint angles predicted by equation (10). We express the objective function g_1 as a quadratic function of $\Delta\phi$:

$$g_1 = \left\| \Delta x_d - J_h \left(K_j + \sum_{i=1}^n J_{c_i}^T K_{c_i} J_{c_i} \right)^{-1} K_j \Delta\phi \right\|^2. \quad (14)$$

For this paper, at each time-step, a simple mid-level controller provides a waypoint, $x_h + \Delta x_d$, that attempts to move the end effector from its current position, x_h , toward the final goal position, $x_g \in \mathbb{R}^3$, along a straight line, where

$$\Delta x_d = \begin{cases} d_w \frac{x_g - x_h}{\|x_g - x_h\|} & \text{if } \|x_g - x_h\| > d_w \\ x_g - x_h & \text{if } \|x_g - x_h\| \leq d_w \end{cases}, \quad (15)$$

and d_w is a small constant distance. Although we do not present alternatives in this paper, other mid-level controllers could potentially provide waypoints in order to perform other actions, such as following a surface or moving along a planned trajectory.

3.4.2. Joint limits: We also add two linear inequality constraints to keep the predicted joint angles within the physical joint limits. These are of the form

$$\Delta\theta_{\min} \leq \Delta\theta \leq \Delta\theta_{\max}, \quad (16)$$

where $\Delta\theta_{\min}$ and $\Delta\theta_{\max}$ are the difference between the minimum and maximum joint limits and the current configuration of the robot. Using equation (10), we can rewrite the inequalities of equation (16) as linear inequalities in $\Delta\phi$.

3.4.3. Contact forces: For each contact, we attempt to restrict the contact force f_{c_i} to be below a *don't care force threshold*, $f_{c_i}^{\text{thresh}}$, and limit the predicted change of the contact force, $\Delta f_{c_i} = f_{c_i}(k+1) - f_{c_i}(k)$, in one time-step. This results in two inequalities for each contact,

$$\Delta f_{\min} \leq \Delta f_{c_i} \leq \Delta f_{\max}, \text{ where} \quad (17)$$

$$\Delta f_{\min} = -f_{c_i}^{\text{rate}}, \text{ and} \quad (18)$$

$$\Delta f_{\max} = \min(f_{c_i}^{\text{rate}}, f_{c_i}^{\text{thresh}} - f_{c_i}). \quad (19)$$

$f_{c_i}^{\text{rate}}$ is a threshold on the maximum allowed predicted change in the contact force in one time-step. We limit the maximum change to keep the motion of the arm smooth. For simplicity, we used the same value of $f_{c_i}^{\text{rate}}$ for both increasing and decreasing the contact force, but this is not a requirement. The term $(f_{c_i}^{\text{thresh}} - f_{c_i})$ in equation (19) explicitly allows contact forces below $f_{c_i}^{\text{thresh}}$.

From equations (9) and (10), the inequalities of equation (17) can be expressed as linear inequalities in $\Delta\phi$.

3.5. Extensions to the quadratic program

In this section, we describe three extensions to the quadratic program of the previous section (Section 3.4).

3.5.1. Squared magnitude of $\Delta\tau$: To discourage large changes in the joint torques in one time-step, we add the term

$$\begin{aligned} g_2 &= \|\Delta\tau\|^2 \\ &= \Delta\phi^T K_j^T K_j \Delta\phi \end{aligned} \quad (20)$$

to the objective function after multiplying it with a scalar weight α_2 , see equation (11). For this paper, we set $\alpha_2 = 0.00001$ for both the real and simulated robot. This weight penalizes large motions of redundant degrees of freedom, but making α_2 too large discourages all motion of the arm.

3.5.2. Decreasing contact forces above the don't care threshold: Due to modeling errors, the contact forces can go above the *don't care force threshold* ($f_{c_i} > f_{c_i}^{\text{thresh}}$). This can lead to an infeasible quadratic program. For example, Δf_{\max} can become less than Δf_{\min} .

For each contact with $f_{c_i} > f_{c_i}^{\text{thresh}}$, we replace equation (19) with $\Delta f_{\max} = 0$. This requires the predicted contact force to either decrease or remain the same. Leaving the predicted contact forces the same by not changing the virtual trajectory, $\Delta\phi = 0$, is always feasible.

We also add an additional term g_3 to the objective function that encourages the controller to decrease these contact forces

$$g_3 = \sum_i \|\Delta f_{c_i}^d - \Delta f_{c_i}\|^2 \text{ if } f_{c_i} > f_{c_i}^{\text{thresh}}, \quad (21)$$

where $\Delta f_{c_i}^d$ is the desired change for the contact force in one time-step and Δf_{c_i} is the change in the predicted contact force. We set $\Delta f_{c_i}^d$ as a force with a constant magnitude and a direction opposite to f_{c_i} . Using equations (9) and (10), we can express g_3 as a quadratic function of $\Delta\phi$.

We set the scalar weight $\alpha_3 = 1$ and $\|\Delta f_{c_i}^d\| = 0.2$ N. These two parameter values enable the controller to reduce the force in a controlled manner in the software simulation and on the real robot.

3.5.3. Limits on the virtual trajectory: On the robot Cody, described in Section 4.2, the joint-space impedance controller limits the virtual trajectory to be within the physical joint limits. To account for this, we add two additional linear constraints on $\Delta\phi$:

$$\Delta\phi_{\min} \leq \Delta\phi \leq \Delta\phi_{\max}. \quad (22)$$

3.6. Stopping our model predictive controller

In the event of a contact force above the safety threshold, $f_{c_i}^{\text{thresh}}$, our model predictive controller stops sending changes to the virtual trajectory, while the “simple”

impedance controller continues to run. Theoretically, if the environment is passive, the arm will then be stable due to the “simple” impedance controller’s stability properties (Hogan and Buerger, 2005; Hogan, 1988).

Additionally, for the experiments in this paper, a higher level controller stops our model predictive controller if the position of the end effector does not change significantly over a period of time, indicating that our model predictive controller is stuck in a local minimum, or if the controller runs for a time greater than a timeout value.

4. Experimental testbeds

We evaluated our model predictive controller using three different testbeds: 1) a software simulation testbed with a three-DOF planar arm, 2) a hardware-in-the-loop testbed that simulates whole-arm tactile sensing for a real seven-DOF arm, and 3) a real seven-DOF arm with a real tactile sensor covering its forearm. The same MPC code written in Python runs on all three experimental testbeds. For visualization, we use the rviz program, which is part of the Robot Operating System (Quigley et al., 2009).

4.1. The simulated robot

This testbed allows us to simulate a large number of trials. We use the open source physics simulation library Open Dynamics Engine (<http://www.ode.org>) to simulate a planar arm with three rotational joints, a 1 kHz joint-space impedance controller, and tactile sensors covering the entire surface of the arm with 100 taxels per meter. Figure 5 shows a visualization of the simulated robot and taxels. The simulation estimates contact forces using linear complementarity constraints (Cottle and Dantzig, 1968; Cottle et al., 2009) and linear friction cone approximations. It also treats contact between two convex objects as a single contact, such as when a cylindrical link of the robot arm makes contact with a cylindrical object. The physics simulation and model predictive controller run in separate processes, but we synchronize their execution so that our model predictive controller consistently runs at very close to 100 Hz with respect to the simulated clock.

The simulated three-link planar arm has kinematics and joint limits similar to a human holding his hand straight out and manipulating in a plane parallel to the ground and at shoulder height. We define the end effector of the arm to be the most distal point on its most distal link, which corresponds with the finger tips of this outstretched hand. The lengths (19.6, 33.4 and 28.8 cm) and masses (2.8, 2.3 and 1.32 kg) of the different links are similar to a human torso and arm, while the diameter of the cylindrical links (3 cm) is similar to a human hand held flat or a small human wrist (Grosso et al., 1989; Tilley and Dreyfuss, 2001). The three joints, with stiffnesses of 30, 20 and 15 Nm/rad, correspond to the torso, shoulder, and elbow. The joint stiffness values

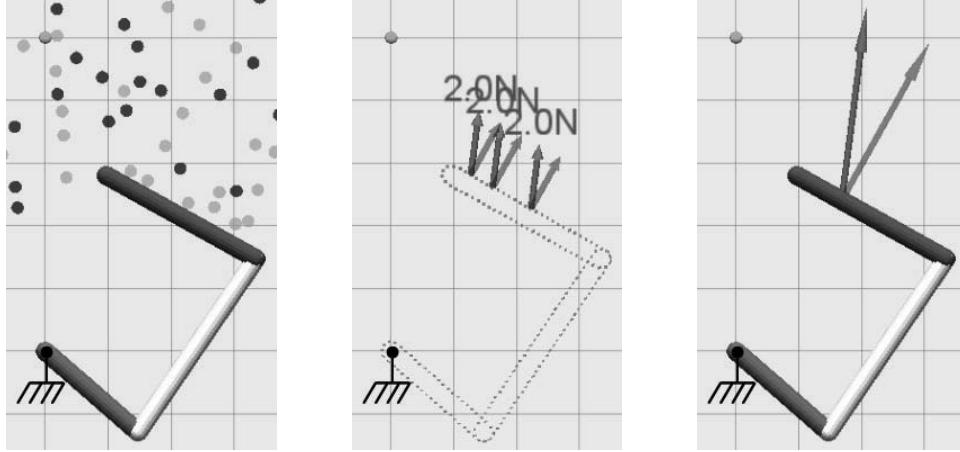


Fig. 5. (Left) Visualization of the three link planar arm reaching to a goal location (cyan) in a volume consisting of rigid cylinders that are either fixed (red) or movable (yellow). The base of the arm is rigidly fixed to the world. **(Middle)** Visualization of the whole-arm tactile sensor. The orange points are 1 cm apart and represent the centers of the taxels. The green arrows are the contact force vectors and each red arrow is the sensed component of the contact force normal to the surface of the arm. **(Right)** Visualization of the contact force and contact location inferred from per-link force-torque sensing (Section 5.2.1). The green arrow is the inferred contact force and the red arrow is its normal component.

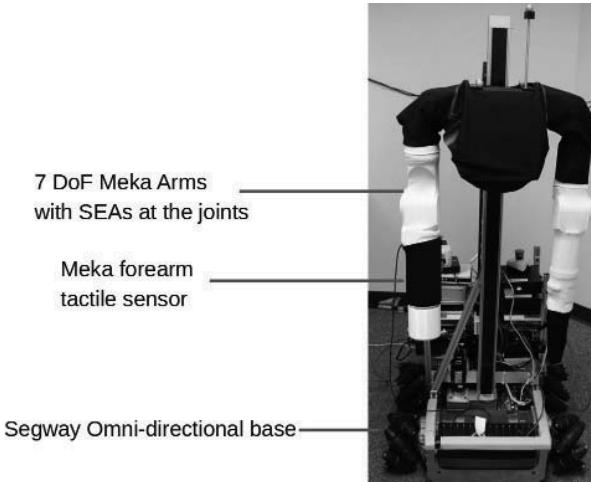


Fig. 6. The robot Cody with two compliant seven-DOF arms and a tactile sensor covering its right forearm.

are similar to measured stiffnesses of humans during planar reaching motions (Shadmehr, 1993).

4.2. The real robot – *Cody*

Figure 6 shows the robot Cody that we use for experiments in this paper. Cody has a Segway omni-directional mobile base, which we control with a PID controller that uses visual odometry (Killpack et al., 2010). Cody also has two seven-DOF arms from Meka Robotics with kinematics similar to human arms and a series elastic actuator (SEA) at each degree of freedom. The passive elastic elements of the SEAs are stiff relative to the active compliance we specify for all joints, except two wrist joints. The passive elastic elements help protect the actuators from shock loads (Pratt

and Williamson, 1995) and may have advantages in wedging and jamming conditions in high-stiffness clutter, but we do not investigate their role in this paper. For this paper, except for two wrist joints, the actuators can be thought of as torque sources.

On Cody, “simple” impedance control runs at 1 kHz within an RTAI real-time thread on a computer running Ubuntu Linux. We use the same stiffness settings as our previous work (Jain and Kemp, 2009b, 2010b). We set the stiffnesses for the shoulder flexion/extension, shoulder abduction/adduction, shoulder internal/external rotation, elbow flexion/extension, and forearm pronation/supination motions at 20, 50, 15, 25 and 2.5 Nm/rad, respectively. The joint stiffness values are also similar to measured stiffnesses of humans during planar reaching motions (Shadmehr, 1993). We use position control for the wrist flexion/extension and abduction/adduction, since the passive compliance of the SEA springs and the cables connecting the SEAs to these two joints is significant.

4.3. Hardware-in-the-loop simulation of tactile sensing

Since we did not have whole-arm tactile sensing on a real robot, we built a hardware-in-the-loop testbed to simulate whole-arm tactile sensing. Figure 7 shows the current implementation of this testbed. The clutter field consists of rigidly mounted posts made of extruded aluminum (80/20® T-Slot Framing) covered with bubble wrap to make them cylindrical and deformable. These posts are similar to the fixed cylinders in the software simulation.

The testbed estimates contact locations on the robot’s arm using geometric collision detection from OpenRAVE (Diankov and Kuffner, 2008) and models of the robot

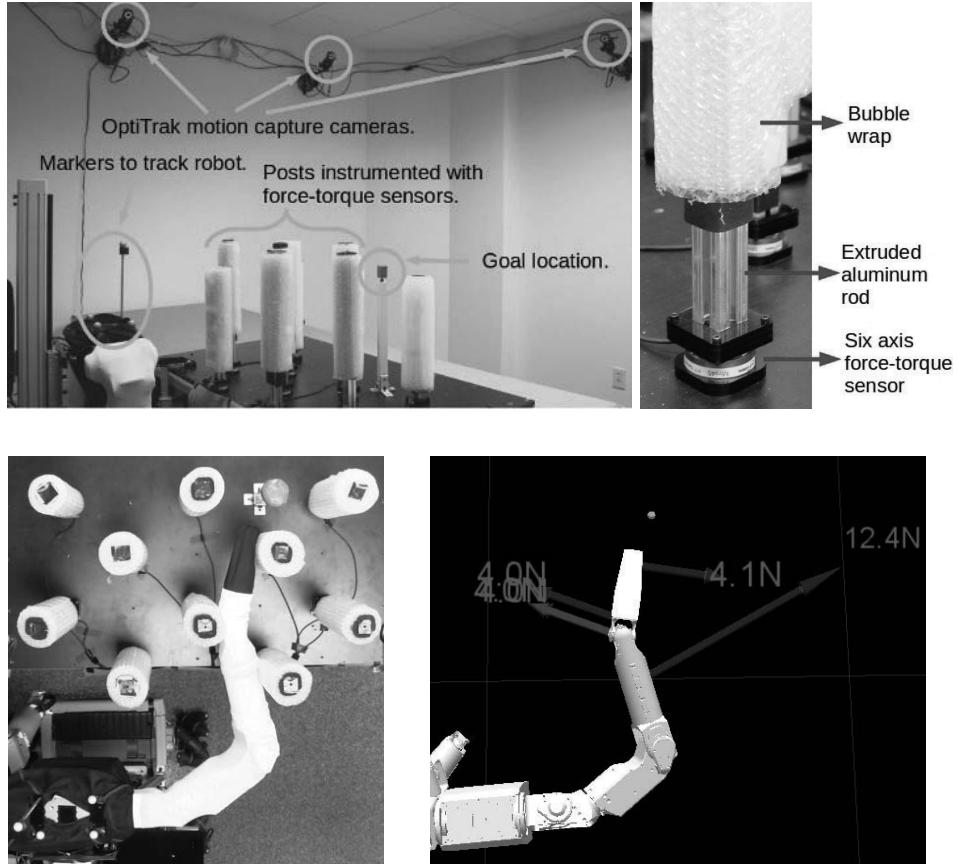


Fig. 7. (Top left) Different components of the hardware-in-the-loop testbed. **(Top right)** Close-up of one instrumented post showing the force–torque sensor at the base of an extruded aluminum rod which we have covered in bubble wrap. **(Bottom left)** Cody attempting to reach to a goal location (green). **(Bottom right)** Visualization of the simulated tactile sensing.

arm and the posts. We use an OptiTrak motion tracking system to estimate the positions of the posts and the pose of the robot with respect to a common frame of reference.

To estimate contact forces, we instrumented each of the posts with a six-axis force–torque sensor (ATI Mini45) at its base. For each instrumented post and robot link in contact with one another, the simulated tactile sensor produces a single contact location, which is the centroid of the estimated contact region, and an associated force vector from the post’s force–torque sensor. If multiple links make contact with the same post, it divides the force magnitude equally among them.

4.4. Real tactile sensing forearm

Figure 8 shows the tactile sensor that covers the forearm of the robot Cody. Meka Robotics and the Georgia Tech Healthcare Robotics Lab developed the forearm tactile sensor, which is based on Stanford’s capacitive sensing technology, as described in Cutkosky and Ulmen (2012).

The forearm sensor consists of 384 taxels arranged in a 16×24 array. There are 16 taxels along the length of the

cylindrical forearm and 24 taxels along the circumference. Each taxel has a dimension of 9×9 mm and a sensing range of 0–30 N. We can obtain the 16×24 taxel array sensor data at 100 Hz using Robot Operating System (ROS) drivers.

On the robot Cody, we added two layers of material on top of the forearm sensor to cover the open parts of the joints, protect the sensor, and make the exterior of the arm low friction. These are shown on the right in Figure 8. The white sleeve is a thin neoprene McDavid compression arm sleeve (Model No. 656T), and the black layer is a padded Ergodyne neoprene elbow sleeve (Model No. 650) designed for athletes.

When using the real tactile sensor, we mounted a rigid 3D printed cylindrical cover over the wrist of the robot, shown in Figure 8. The cover serves to detect contact forces across the entire surface of the most distal part of the arm using the wrist-mounted force–torque sensor. It also creates a smooth surface over the wrist joint. The cover is flush with the tactile sensing forearm such that the last two wrist joints are unable to rotate when it is mounted.

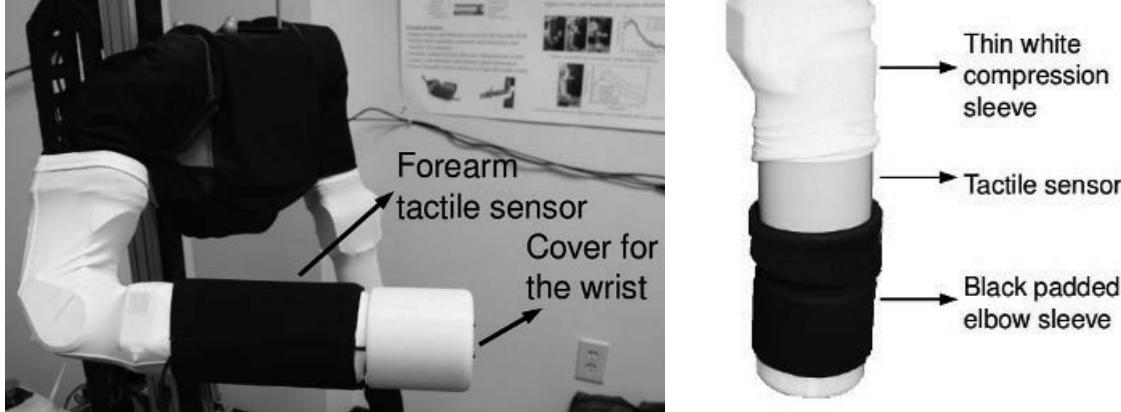


Fig. 8. (Left) Tactile sensor on the right forearm of Cody (underneath the black neoprene sleeve) as well as a 3D printed cover for the wrist. (Right) Two layers of material (thin white compression sleeve and black padded sleeve) that we added on top of the tactile sensor (blue).

4.5. Detecting contact

As mentioned in Section 3.3, the controller uses the measured force to detect a contact. Specifically, the controller only adds a contact to its model if the force magnitude is greater than a threshold.

This threshold is 0.5 N for the simulated robot and 1 N in the hardware-in-the-loop testbed. For the real tactile sensor, we precompute the standard deviation of the sensor noise for each taxel when the arm is not in contact with the environment. The threshold for detecting contact is six times the standard deviation of the sensor noise or 1 N, whichever is greater. For contacts on the wrist cover, the threshold is 1 N.

4.6. Tactile sensing limitations

Due to limitations in our implementations, there are differences in tactile sensing with the three testbeds.

The tactile sensors for the software simulated robot and the real tactile sensing forearm match the assumptions we used in our controller derivation. They report the geometric centers of the taxels at which contact is detected along with the total normal force measured at each taxel.

While testing the real tactile sensing forearm, the plastic cover on the distal part of Cody's arm also reports contact forces. These forces are not handled in the same way as our derivation. Instead, the contact location is the center of the force-torque sensor, and the contact force is the total resultant force measured by the force-torque sensor. The controller treats this force vector as though it were a normal force applied at the center of the force-torque sensor, which implies that it will not ignore frictional forces. In other words, the spring associated with this contact in the mechanical model is oriented along the direction of this force vector rather than a surface normal.

Similarly, the hardware-in-the-loop simulation of tactile sensing does not report normal forces, and instead provides the centroid of the estimated contact region as the contact location, and a scaled version of the total resultant force

vector measured by the object's force-torque sensor as the contact force.

The primary consequence of these differences is that frictional forces will influence the controller rather than being ignored. This influence is mitigated by the smooth, low-friction surfaces of the plastic cover and Cody's arm. We also put neoprene McDavid compression arm sleeves on Cody's arm when it operates in the hardware-in-the-loop system.

5. Approaches used for comparison

In this section, we describe four methods against which we compared our controller: 1) a baseline controller; 2) our model predictive controller with per-link force-torque sensing; 3) our model predictive controller without force sensing, and 4) a state-of-the-art geometric motion planner that has full knowledge of the environment.

5.1. Baseline controller

Our baseline controller uses the same joint-space impedance control as our model predictive controller to maintain low stiffness at the joints. However, it does not use feedback from the tactile sensor, except to define a safety stopping criterion. Specifically, this controller computes

$$\Delta\phi^* = (J_h^T J_h)^{-1} J_h^T \Delta x_d, \quad (23)$$

where $\Delta\phi^* \in \mathbb{R}^m$ is the incremental change in the joint-space virtual trajectory (see Figure 3), $J_h \in \mathbb{R}^{3 \times m}$ is the Jacobian at the robot's end effector (hand), and $\Delta x_d \in \mathbb{R}^3$ is the desired Cartesian motion of the end effector computed from equation (15). The baseline controller monitors the tactile sensor values and stops if the force at any contact goes above the safety force threshold, $f_{c_i}^{\text{safety}}$.

Equation (23) is the solution to the quadratic program for our model predictive controller, if the manipulator is not in

contact with anything, the constraints are ignored, and only g_1 (see equation (12)) is used for the objective function. In free space, both controllers behave similarly, since they attempt to move the end effector along a straight line to the goal with identical low-stiffness settings at the joints.

5.2. Our model predictive controller with different sensors

In order to compare whole-arm tactile sensing with other sensors, we simulated an arm with force-torque sensors at the base of each link and an arm with no force sensors.

5.2.1. Simulated force-torque sensors: For the arm with simulated force-torque sensors at each link, the robot interprets any set of contacts as at most one contact force per link. For each link, the simulated robot first uses Poinsot's theorem to calculate the resultant force and its line of action due to all external forces applied to the link (Murray et al., 1994). It then estimates the associated contact location as the intersection of this line of action with the link's axis. Notably, this contact location may not be on the body of the link. For example, a moment due to canceled forces can result in the contact location being off the link. Additionally, the contact force's direction need not be normal to the link's perimeter. Instead, the spring associated with the contact in the controller's mechanical model is oriented along the direction of the resultant force. As we briefly discuss later, this method resulted in the best overall performance of our controller with per-link force-torque sensing out of several more elaborate methods. Figure 5 shows a visualization of the feedback from the simulated force-torque sensors.

Our simulated force-torque sensors are idealized, giving an upper bound on sensor performance. The sensing does not have any noise or drift besides the numerical noise in the software simulation. In addition, the force-torque sensing only reports contact forces directly due to contact with the environment, since forces due to accelerating the mass of the links, friction, and other phenomena are ignored. This is akin to perfectly estimating and compensating for all phenomena that could interfere with measurements from per-link force-torque sensors.

5.2.2. No force sensing: The arm with no force sensing performs “simple” impedance control, but provides no contact locations nor contact forces to the model predictive controller. Consequently, the controller believes it is operating in free space, and will move the end effector directly toward the goal location, subject to the remaining constraints and objective function terms. Although it does not use force sensing for maneuvering, if a contact force exceeds the safety threshold, the model predictive controller stops, as described in Section 3.6.

5.3. Conservative assumptions about the number of contacts

Our controller attempts to keep each contact force below the *don't care force threshold*. When using per-link force-torque sensing, it makes the conservative assumption that the load associated with each link results from a single contact point. It does this, since it does not know the actual number of contacts on a link over which the load is being distributed. With tactile sensing it makes a similar conservative assumption by treating all contacts applied to a single taxel as a single contact.

5.4. Motion planner with full knowledge

For experiments in the software simulation testbed, we also compared against a bi-directional RRT motion planner (Bi-RRT) as implemented in OpenRAVE (Diankov and Kuffner, 2008). The motion planner has complete knowledge of the fixed cylinders in the cluttered environment and we remove all the movable cylinders. We use Bi-RRT to estimate whether or not a solution exists for a given goal location and configuration of the fixed cylinders. We use this to estimate an upper bound for the success rate for a given set of trials. This is an approximate upper bound, since there may be no way to reach the goal when the movable cylinders are present.

6. Software simulation experiments

In this section, we describe experiments with the software simulation testbed (see Section 4.1). For all experiments, the simulated arm reached to a goal location in a clutter field starting from the initial configuration. We considered a reach successful if its end effector (the most distal point on the most distal link) reached to within 2 cm of the goal location. Extension 9 is the code and instructions to reproduce the results of this section.

6.1. Simulating clutter

Our simulated clutter fields are composed of rigid cylinders with a diameter of 2 cm that are either fixed to the ground or movable. In isolation, a movable cylinder slides with the application of approximately 2 N (static friction and kinetic friction are set to be equal).

To generate a clutter field, we randomly place a predefined number of fixed and movable cylinders into a 1.2×0.6 m rectangular region. For each cylinder, we uniformly sample a two-dimensional location from the rectangular region (see Figure 9), redrawing until we find a location for which the associated cylinder does not collide with the cylinders that have already been placed.

We use this simplified model of clutter to randomly generate clutter fields with different ratios of fixed to movable clutter, and different densities of clutter. This clutter model shares some similarities with dense foliage, which often

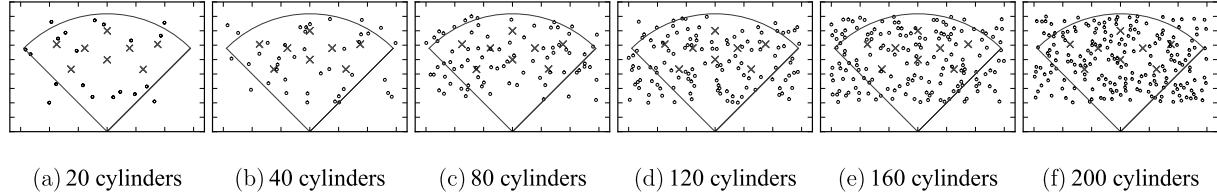


Fig. 9. An example of the six different levels of clutter that we used in the evaluation of Section 6.3. The cylinders are the black circles, the eight goal locations are green crosses, and the blue curve outlines the area swept by the fully extended arm when its proximal joint moves from 45° to 135° . The physical limits for the proximal joint are at -150° and 150° .

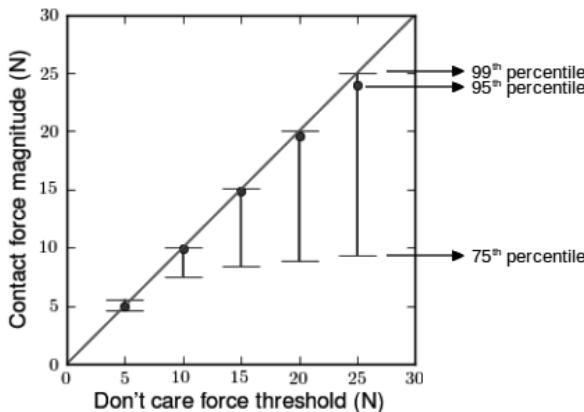


Fig. 10. 75^{th} , 95^{th} , and 99^{th} percentile scores for the contact force from 200 trials of single reaches to eight different goal locations in 25 distinct clutter fields for different values of the *don't care force threshold*. Details are in Section 6.2.

consists of stiff branches and movable leaves. Reaching into this type of clutter results in a diverse set of contact conditions with numerous point contacts occurring across the arm, some of which result in high forces when pushed against and others of which result in lower forces. Making contact with this simulated clutter also results in secondary effects due to contact between the cylinders.

6.2. Regulating contact forces

To test the influence of the *don't care force threshold*, $f_{c_i}^{\text{thresh}}$, we generated 25 clutter fields, each with 20 fixed and 20 movable cylinders and 8 constant goal locations as shown in Figure 9. We then ran our model predictive controller with five different *don't care force threshold* values on these 200 trials (25 fields \times 8 goals).

Figure 10 shows the 75^{th} , 95^{th} , and 99^{th} percentile scores for the magnitude of the contact force for each value of $f_{c_i}^{\text{thresh}}$. The correlation coefficient between $f_{c_i}^{\text{thresh}}$ and the 95^{th} percentile score was ≥ 0.999 , providing evidence that the $f_{c_i}^{\text{thresh}}$ parameter can be used to predictably influence the contact forces.

For the contact force statistics we report here and in other simulation experiments, we use the raw contact forces from the physics simulation. The physics simulation computes contact forces between all objects at every time-step. For our results, we use all of the raw contact forces applied

to the manipulator by the clutter objects, rather than measurements from the simulated robot's sensors or the normal components of the raw forces.

6.3. Reaching trials

We used the simulation testbed to compare our model predictive controller with whole-arm tactile sensing to a version with per-link force-torque sensing, a version with no force sensors, and Bi-RRT, as described in Section 5.2.

We varied the level of clutter with six different values for the total number of cylinders (20, 40, 80, 120, 160 and 200). For this paper, we refer to 20 and 40 cylinders as “low clutter”, 80 and 120 cylinders as “moderate clutter”, and 160 and 200 cylinders as “high clutter”. We also changed the composition of the clutter to have movable cylinders constituting 0, 50 or 100% of the clutter with the rest fixed. Overall, we ran $3 \times 6 \times 3 \times 150 \times 8 = 64800$ trials of the simulated robot reaching in clutter with 3 types of sensor feedback, 6 levels of clutter, 3 ratios of movable to fixed cylinders, 150 randomly generated clutter field configurations, and 8 constant goal locations. Figure 9 shows examples of the different levels of clutter and the goal locations.

For all trials, we set $f_{c_i}^{\text{thresh}} = 5\text{ N}$ and $f_{c_i}^{\text{safety}} = 50\text{ N}$ for each contact c_i . If the end effector of the arm did not move more than 1 mm for a period of 10 s, the trial ended as a failure.

6.3.1. Success rates for three controllers and Bi-RRT: Figure 11 shows the performance of the three controllers, and our estimated upper bound for the success percent via Bi-RRT, which we will refer to as the estimated optimal success rate. As expected, the estimated optimal success rate and the controllers' success rates decrease with increasing clutter.

We also tested the statistical significance of the differences between the success rates for whole-arm tactile sensing versus per-link force-torque sensing for 18 different conditions (6 levels of clutter \times 3 ratios of movable to fixed objects). To do so, we used a standard significance test for comparing two proportions (McClave et al., 2008). We found statistical significance with p -values less than 0.01 for moderate and high clutter composed of 100% and 50% movable objects. We did not find significance for any condition with 100% fixed objects due to the low number of successful trials. By running 3600 additional trials with 80

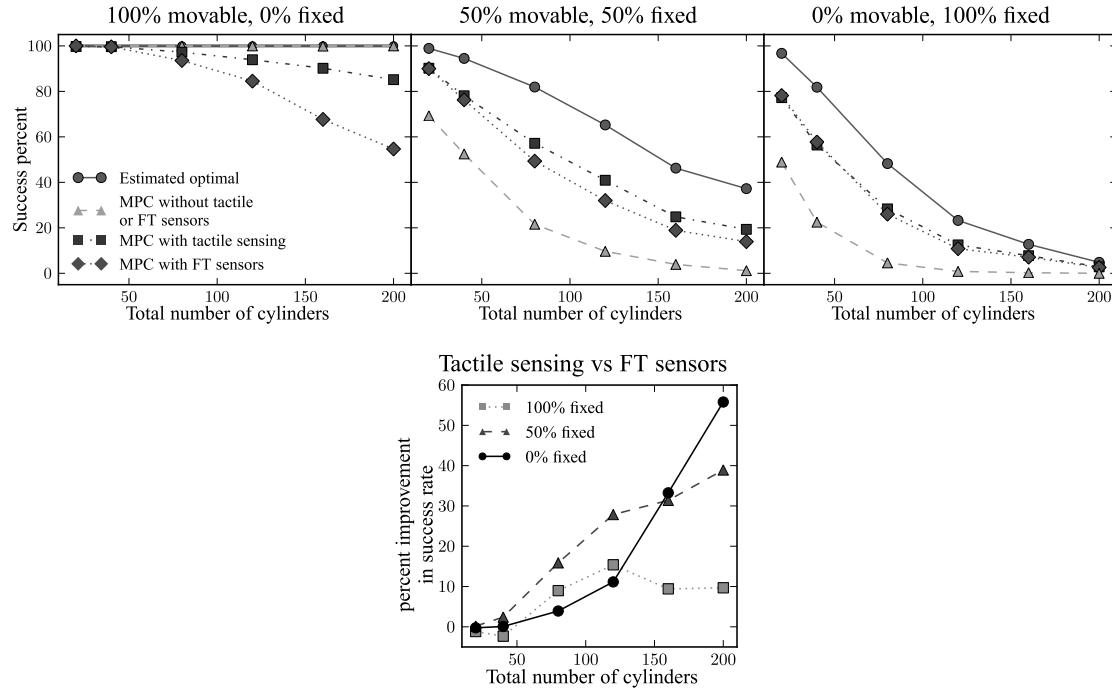


Fig. 11. (Top) Success rate for a single reach attempt in different levels of clutter and ratios of fixed to movable clutter. Each data point is the success rate over 1200 trials with a single reach towards the goal per trial. **(Bottom)** Percent improvement in the success rate when the robot uses whole-arm tactile sensing instead of per-link force-torque sensing. Details are in Section 6.3.1.

fixed objects, we found statistical significance ($p < 0.05$) for relative performance comparable to that shown in Figure 11. Specifically, when including these additional trials, whole-arm tactile sensing performed 9.36% better than per-link force-torque sensing with 80 fixed objects. Finding statistical significance for higher levels of clutter would have required more trials than we could perform.

For 100% and 50% movable objects, higher clutter resulted in larger effect sizes. As shown in the far right graph of Figure 11, the relative improvement of whole-arm tactile sensing over per-link force-torque sensing generally increased as the level of clutter increased, resulting in large improvements at the highest level of clutter (200 objects).

With fixed objects, both the controller with whole-arm tactile sensing and per-link force-torque sensing had higher success rates than the controller with no sensing. However, with 100% movable objects, the controller with no sensing performed better, since it could push into the movable objects with arbitrarily large forces to achieve a 100% success rate.

6.3.2. Examples of whole-arm tactile sensing succeeding when force-torque sensing fails: Figure 12 shows two examples of multi-contact situations for which per-link force-torque sensing failed, but whole-body tactile sensing succeeded. In the middle example, the robot with per-link force-torque sensing became stuck due to contact with a fixed cylinder and with a movable cylinder in contact with

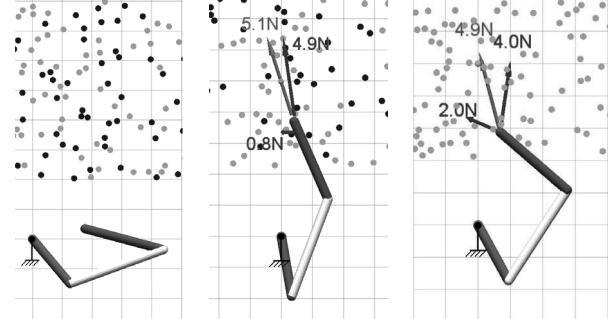


Fig. 12. (Left) The initial configuration of the simulated arm for our reaching trials. The first joint remained fixed to the ground at the same location throughout all of our tests. **(Middle and Right)** Two multi-contact situations in which our model predictive controller became stuck with per-link force-torque sensing and succeeded with whole-arm tactile sensing. The cyan circle denotes the goal location, the blue arrows show the actual contact forces, and the green arrow shows the contact force and contact location inferred from per-link force-torque sensing. **(Middle)** The distal link of the robot makes contact with a fixed cylinder and a movable cylinder in contact with a fixed cylinder. **(Right)** The distal link makes contact with two movable cylinders.

a fixed cylinder. In the example on the right, the robot with per-link force-torque sensing became stuck due to contact with two movable cylinders, one of which was in contact with an additional movable cylinder. For both of

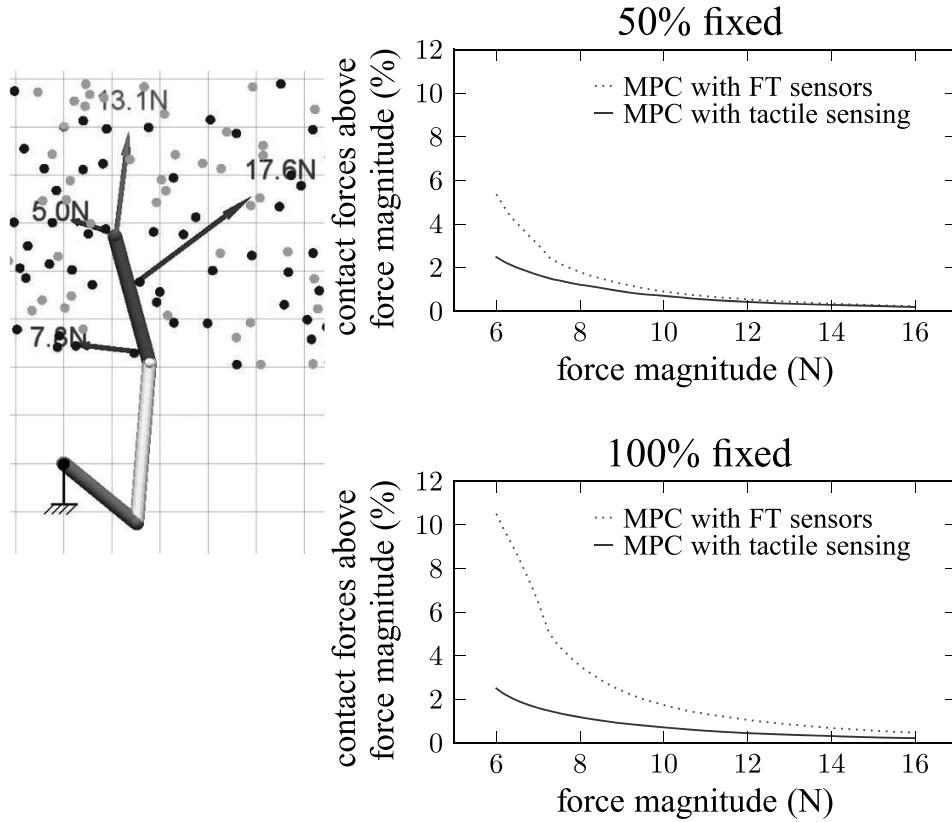


Fig. 13. (Left) An example where our model predictive controller with per-link force–torque sensing resulted in the robot applying large forces. The blue arrows are the actual contact forces and the green arrow is the inferred contact force. **(Right)** A comparison in contact force control between per-link force–torque sensing and whole-arm tactile sensing using the percentage of contact forces that are above a force magnitude at all levels of clutter. Composition of clutter for the top graph is 50% fixed, 50% movable and is 100% fixed for the bottom graph.

these multi-contact situations, per-link force–torque sensing infers a single contact force and associated contact location.

Extensions 2 and 3 are videos of these two trials. In the videos, the robot starts from the same initial configuration in free space, fails with per-link force–torque sensing and succeeds with whole-arm tactile sensing. Additionally, for these two trials, we first ran the controller with per-link force–torque sensing until it became stuck, and then switched to whole-arm tactile sensing. The robot was then able to successfully reach the goal starting from the state at which it had been stuck.

6.3.3. Contact forces for three controllers: In order to compare the contact forces that occurred during reaching with the three controllers, we calculated the percentage of contact forces with magnitudes above or below various force magnitudes (see Figure 13 and Table 1). These results can be interpreted as the chance that the magnitude of a contact force will be above or below a given value, if the magnitude of each contact force were generated by an independently and identically distributed random variable.

As shown in Table 1, using both whole-arm tactile and per-link force–torque sensing, the controller successfully regulated most contact forces that occurred while reaching in 50% movable clutter to be below or slightly above the *don't care force threshold*, $f_{c_i}^{\text{thresh}}$, of 5 N. Without sensing, however, a large percentage of contact forces were above 6 N. We also found the differences among all of these values to be statistically significant with p -values less than 0.01 using a significance test for comparing two proportions (McClave et al., 2008).

As shown in Figure 13, per-link force–torque sensing resulted in higher contact forces than whole-arm tactile sensing. With per-link force–torque sensing, a higher percentage of contact forces were above a given force magnitude threshold for all values of the threshold between 6 N and 16 N. We also found the differences over this range with increments of 0.125 N to be statistically significant with p -values less than 0.01 using a significance test for comparing two proportions (McClave et al., 2008). The differences were greater in clutter with 100% fixed objects versus 50%

Table 1. Percentage of contact forces for which haptic sensing with our controller is able to maintain the forces below the given threshold for a 50% fixed and 50% movable object ratio.

	Percentage of contact forces below	
	5 N	6 N
50% movable, 50% fixed	5 N	6 N
Tactile sensors	86.8%	97.5%
Force-torque sensors	84.8%	94.6%
Without tactile or FT sensors	48.8%	50.6%

With 100% fixed objects, per-link force–torque sensing resulted in over four times as many contact forces above 6 N in magnitude than whole-arm tactile sensing. Additionally, with both per-link force–torque sensing and no force sensing the percentage of contact forces above 6 N was higher in moderate and high clutter than in low clutter, but remained low with whole-arm tactile sensing. In other words, with 100% fixed objects, whole-arm tactile sensing had more relative value in higher clutter. For example, with whole-arm tactile sensing 2.18%, 2.51% and 2.5% of contact forces were above 6 N with 20, 80 and 200 objects, respectively. These percentages were 6.16%, 10.95%, and 11.02% for per-link force–torque sensing, and 80.65%, 86.68% and 88.89% for no force sensing.

6.3.4. Average velocity for three controllers: The average end effector velocities for failed trials were dominated by the 10 s timeout. For successful trials, the average end effector velocities were 2.57 cm/s, 2.40 cm/s and 2.85 cm/s for our model predictive controller with whole-arm tactile sensing, per-link force–torque sensing, and no force sensing, respectively. No force sensing was fastest, but did not regulate the contact forces and was successful over far fewer trials. Whole-arm tactile sensing was faster than per-link force–torque sensing, even though it achieved lower contact forces and higher success rates.

6.4. Local minima

Figure 14 shows an example of a local minimum in which our controller can become stuck. In order to escape from local minima, some form of high-level control is required. For some of our experiments, we used a high-level controller that attempts to move the end effector in a straight line towards the goal, detects when the arm has stopped making progress (reached a local minimum), pulls the arm back to a new configuration, and then tries again. We treat this high-level controller as a black box for the current paper. Extension 4 is a video showing this additional control layer in a simplified environment.

Using this additional control layer, we tested our controller in simulation with whole-arm tactile sensing on a clutter ratio of 50% fixed and 50% movable objects with six levels of clutter. We used a total of 7200 trials during

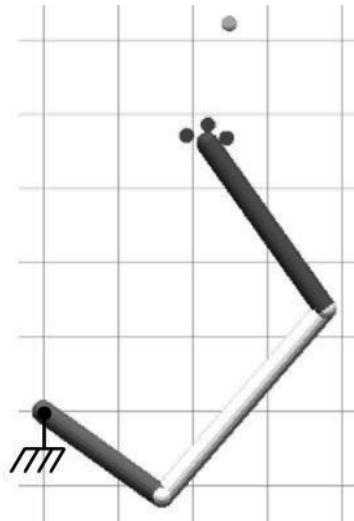


Fig. 14. An example of our greedy model predictive controller stuck in a local minimum. The arm can reach the goal location (cyan circle) if it goes around the cylinders (red circles) but this would involve an increase in the cost function.

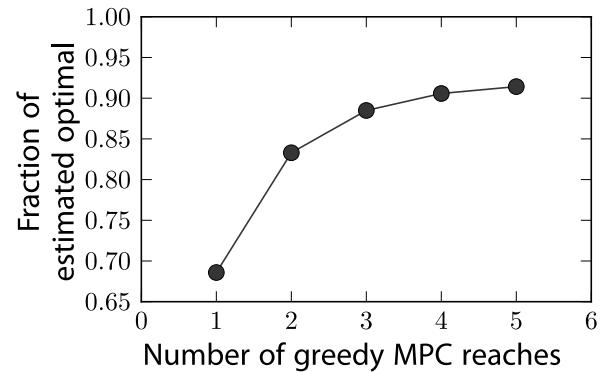


Fig. 15. Success rate of our model predictive controller with whole-arm tactile sensing as a fraction of the estimated optimal success rate with multiple retries.

which the arm reached to the goal multiple times, (6 levels of clutter \times 1 ratio of movable to fixed cylinders \times 150 clutter fields \times 8 goal locations). We allowed our greedy controller to pull out and restart from up to five different end effector positions equally spaced along a line in front of the clutter field rectangle. During a reach attempt, if the end effector of the arm did not move more than 1 mm for a period of 2 s, the high-level controller aborted the current reach and attempted to pull the end effector out of the clutter to a new starting location. If the end effector failed to reach this new starting location, we counted the trial as a failure.

Figure 15 shows that the success rate as a percentage of the estimated optimal success rate increased with additional reach attempts. With a single reach, the success rate of our model predictive controller with tactile sensing was 68.6% of the estimated optimal. With up to five reaches, the success rate was 91.4% of the estimated optimal.

6.5. Discussion of software simulation results

Our experiments with a software simulated robot arm reaching in clutter demonstrate that our controller can keep most contact forces below a desired threshold while using per-link force-torque sensing or whole-arm tactile sensing. For moderate clutter, whole-arm tactile sensing outperformed per-link force-torque sensing in terms of success rates and keeping contact forces low. The extent to which these results would generalize to real-world applications remains an open question. While many improvements were statistically significant, the implications of the effect size for real-world performance would depend on the specific application. For some applications, even small increases in the success rate and small reductions in the contact forces may be beneficial (e.g. medical and military applications). The effect sizes were large in high clutter, but modest in moderate clutter. For example, with 200 movable objects (high clutter), whole-arm tactile sensing resulted in a 55.8% better success rate than per-link force-torque sensing, and resulted in a 38.9% better success rate when 50% of the objects were fixed (see Figure 11). With 80 total objects (moderate clutter), whole-arm tactile sensing had a 3.9% better success rate with 100% movable objects and a 15.9% better success rate with 50% fixed objects.

We found these trends to be robust over parameter variations in our simulation. For example, for our single reach trials in simulation, if the end effector of the arm did not move more than 1 mm for a period of 10 s, the trial ended in failure. Changing this timeout to 2 s resulted in lower contact forces for per-link force-torque sensing, but also lower success rates. Likewise, we tested the performance of the controller with alternative methods of converting per-link force-torque sensing into estimated contact locations and forces, such as requiring the inferred contact force's location to be on the link's perimeter and only using the component of the force normal to the link's perimeter. Out of the methods we tested, the method we present in this paper had the best overall performance. In general, for the variations we tested, higher success rates came with higher contact forces, while lower contact forces came with lower success rates.

The 100% movable object conditions highlight that tactile sensing enables the robot to apply a large total load distributed across many contacts, and in doing so achieve a higher success rate while keeping contact forces low. The 100% fixed object conditions show that this is not the only source of performance gain. Most notably, with 100% fixed objects whole-arm tactile sensing resulted in much lower contact forces than per-link force-torque sensing, and this improvement was greater with higher clutter. With 80 fixed objects (moderate clutter), whole-arm tactile sensing also had a higher success rate that was statistically significant. In general, however, the differences in success rates for 100% fixed clutter were much less pronounced than the differences in contact forces. Interestingly, in moderate clutter composed of 50% movable objects, whole-arm tactile

sensing had higher relative performance than it did with moderate clutter composed of 100% movable or 100% fixed objects, which suggests that combining movable and fixed objects results in clutter with distinct properties.

Another consideration is how the clutter levels we used in simulation relate to real-world clutter. To help quantify this, we examined the number of contacts that the controller handled at each time-step in two of our experiments. For the real robot with a real forearm tactile sensor reaching in artificial foliage (described in Section 7.2.1), we found that there were a maximum of 21 contacts that occurred simultaneously (i.e. 21 forearm taxels above threshold). For the simulated robot in the highest 100% movable clutter we tested (200 objects), the maximum number of contacts that occurred simultaneously was 14 contacts across the entire arm. This illustrates that even the highest clutter conditions in our software simulation resulted in lower numbers of contacts than can occur during real-world manipulation. For example, moving in foliage or against a compliant surface can result in contact with a large number of taxels.

7. Experiments with a real robot

In this section, we present results from experiments with the real robot named Cody.

7.1. Hardware-in-the-loop tactile sensing simulation testbed

The experiments in this section use the hardware-in-the-loop testbed to simulate whole-arm tactile sensing.

7.1.1. Selective control of force applied to different regions of the environment: With this experiment, we illustrate that our model predictive controller can be used to selectively control the contact forces in different regions. We defined a ‘fragile’ cylindrical volume with respect to the world frame. If the location of a contact c_i in the world frame was within the ‘fragile’ volume, we set the *don't care force threshold*, $f_{c_i}^{\text{thresh}}$, to 2 N. For contact outside this volume, we set the *don't care force threshold* to 5 N.

Figure 16 shows histograms of the contact forces while reaching to the goal location. The contact forces within the ‘fragile’ region had a median of 1.76 N, while the contact forces outside of the ‘fragile’ region had a median of 4.54 N.

7.1.2. Test of our model predictive controller as part of a fully autonomous system: Using the hardware-in-the-loop testbed, we tested our model predictive controller as part of a fully autonomous system. The autonomous system had two high-level controllers. The first high-level controller (HC1) moved the robot’s mobile base to one of three pre-defined positions equally spaced along a line in front of the clutter field. The robot would then hold its arm in a pre-defined pose while slowly moving forward towards the clutter field until it detected contact with its arm.

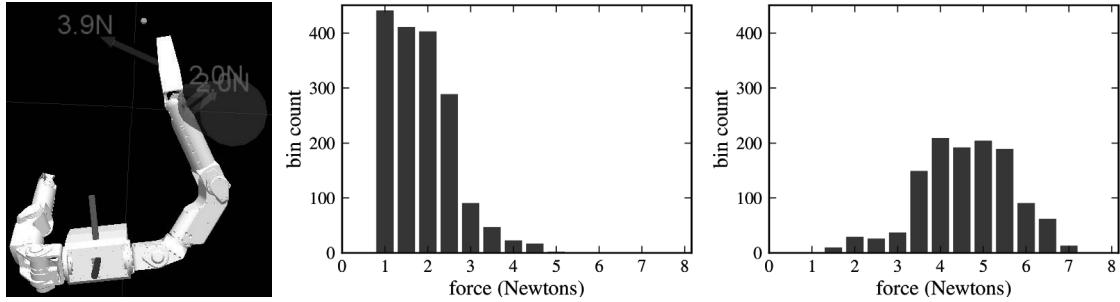


Fig. 16. (Left) Experiment to demonstrate selective control of contact force in different regions using our model predictive controller. Tactile sensing visualization when some contacts are within the ‘fragile’ region (red cylinder) and other contacts are outside the ‘fragile’ region. The cyan circle is the goal location that the robot successfully reached. **(Middle)** Histogram of contact forces within the ‘fragile’ region. **(Right)** Histogram of contact forces outside the ‘fragile’ region.

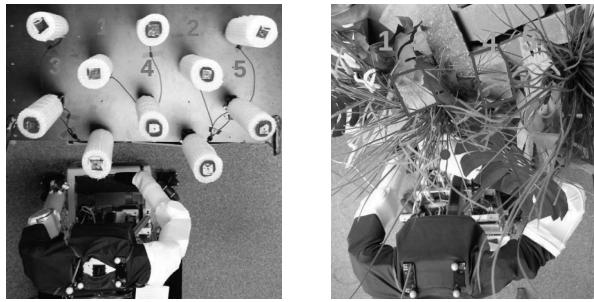


Fig. 17. (Left) Five different goal locations within the hardware-in-the-loop testbed that we used to test our model predictive controller as part of a fully autonomous system, as described in Section 7.1.2. **(Right)** Five different goal locations in artificial foliage that we used to compare the model predictive controller and the baseline controller, described in Section 7.2.2. The environment consists of compliant leaves and rigid blocks of wood (outlined in red). The red circle denotes the position of the end effector.

At this point, the second high-level controller (HC2) took over. HC2 attempted to reach the goal location using our model predictive controller. If HC2 detected that the end effector stopped making progress towards the goal location, it would attempt another reach or pull the arm out and return control to HC1. For each of up to four reach attempts, HC2 would perform some combination of pulling the arm out and moving the end effector to the left or right of a post. The details of this algorithm are beyond the scope of this paper. Extension 5 is a video of the full results of successfully reaching the five goal locations shown in Figure 17.

We also recorded the contact forces that occurred during the successful reach for each of the five trials. The average of the maximum force for each of these five reaches was 5.6 N. The average of the contact forces in these five reaches that exceeded the *don’t care force threshold*, $f_{ci}^{\text{thresh}} = 5$ N, was 5.5 N.

Additionally, for each of the five initial conditions (mobile base location and arm pose) from which HC2 succeeded, we attempted a single reach with the baseline controller. The baseline controller succeeded with three out of

five of these initial conditions, had an average maximum contact force of 17.7 N, and an average contact force above 5 N of 14.3 N. This indicates that two of the five successful reaches performed by the fully autonomous system benefitted from our model predictive controller, HC2, or both. Our controller also kept the contact forces lower than the baseline controller.

Our results with the baseline controller in this experiment and those we describe later in this section emphasize that compliance alone is insufficient to achieve success at reaching in clutter, and that even a highly compliant robot can produce high contact forces when reaching in clutter.

7.2. Real tactile sensing forearm

We also conducted experiments with the forearm tactile sensor described in Section 4.4 and Figure 8. Since the tactile sensor only covers the forearm of the robot, we designed the experiments to restrict contact to the forearm and the plastic cover on the wrist.

We performed experiments with clutter that simulates foliage and rubble. The artificial foliage consists of both compliant objects (plastic leaves) and fixed rigid objects (blocks of wood). The leaves can result in substantial occlusion (see Figure 1), and can be pushed aside with relatively low force. However, the blocks of wood are hidden, rigid, and effectively immovable.

The cinder block is a rigid, heavy, and fixed object, representative of some of the objects a robot would encounter in rubble. The diameter of the robot’s forearm is 10 cm. It is close to the width of the opening of the cinder block, which varies between 13.5 cm and 14.5 cm. Additionally, the edges are sharp and the surface is abrasive.

7.2.1. Illustrative examples – foliage and cinder block: We performed one trial each of the robot reaching to a goal location in foliage and reaching through the opening of a cinder block. Figure 19 shows two images and the histograms of the contact forces for these two trials. The average end effector velocities were 2.95 cm/s and 2.14 cm/s

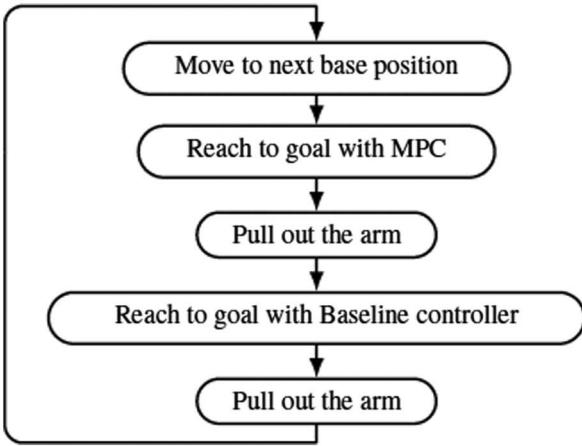


Fig. 18. Different steps that the robot performed for each of the five trials in our model predictive controller vs. baseline controller comparison in foliage, described in Section 7.2.2.

Table 2. Model predictive controller vs. baseline controller in foliage.

	MPC	Baseline controller
Success rate	3/5	1/5
Exceeded safety threshold (15 N)	0/20 attempts	19/20 attempts
Average max. contact force	5.5 N	14.5 N
Average contact force above f_{ci}^{thresh} (5 N)	5.2 N	9.2 N

respectively. Extensions 6 and 7 are videos of these two trials.

7.2.2. Model predictive controller vs. baseline controller in foliage: For a more thorough evaluation of our model predictive controller and the baseline controller in realistic conditions, we performed five trials with automatically generated goal locations that were equally spaced along a line within the artificial foliage, as shown in Figure 17.

We started each trial by positioning the robot at the same location in front of the clutter. The robot then autonomously moved its mobile base to four roughly equally spaced positions along a line, and attempted to reach to the goal location using both the model predictive controller and the baseline controller, as shown in Figure 18.

Table 2 shows the results from the twenty reach attempts (4 base positions \times 5 trials) for each controller. The model predictive controller successfully reached goal locations 1, 3 and 5, while the baseline controller was only successful for goal location 5. Further, our model predictive controller successfully kept the contact forces around the *don't care*

force threshold, $f_{ci}^{\text{thresh}} = 5 \text{ N}$. In contrast, the baseline controller exceeded the safety force threshold, $f_{ci}^{\text{safety}} = 15 \text{ N}$, 19 out of 20 times.

Extension 8 is a video of the complete experiment.

7.2.3. Contact stiffness parameter's effect on performance: In this section, we describe an experiment to investigate how the stiffness used by the controller to model each contact, K_{ci} , affects the controller's performance. For all other experiments in this paper, we set this parameter to a high stiffness value. For example, in the software simulation experiments and the other experiments involving the forearm tactile sensor, we set this parameter to be 1000 N/m. This is a conservative value that makes the robot think that the world is stiff. Consequently, the robot tends to move slowly while in contact, since it predicts that contact forces will rise quickly for small displacements.

As we show in this section, when the robot makes contact with a low-stiffness object, this high parameter value makes the robot move unnecessarily slowly. With a lower stiffness value that better matches the compliance of the object, the robot pushes more aggressively into the object to reach the goal location.

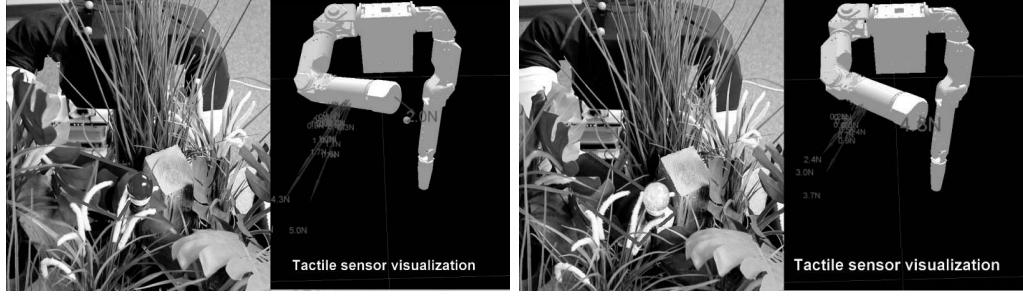
For the experiment, we fixed a cylindrical piece of styrofoam in front of the robot in an upright position, and selected a goal location such that the robot's arm could push and bend the cylinder to reach the goal (see Figure 20). We estimated the stiffness of the cylinder by hand and found it to be approximately between 100 and 300 N/m in the region where the robot pushed on the cylinder. Our measured stiffness values varied with the height of contact and how much the cylinder was displaced.

The robot used our model predictive controller to reach the same goal location five times for each of five contact stiffness values (200, 500, 1000, 3000 and 5000 N/m), giving a total of 25 trials. The robot successfully reached the goal location in all 25 trials.

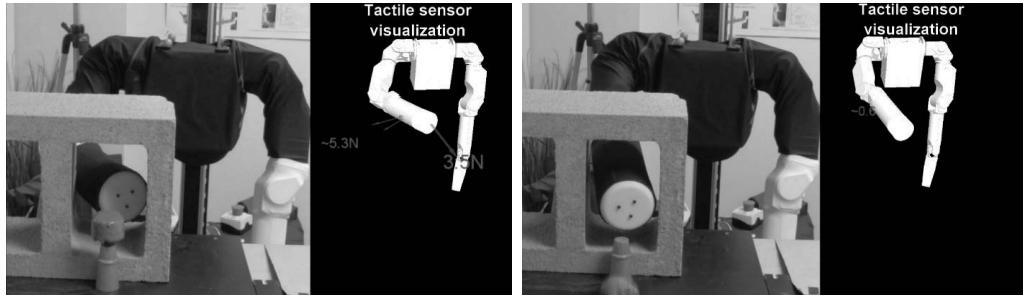
Table 3 shows the mean and standard deviation of the execution time and the maximum contact force for each trial, as sensed by the forearm tactile sensor. The results show that as the controller's contact stiffness estimate approached the actual stiffness, the execution time decreased while the forces continued to remain below the *don't care force threshold*, which we set to 5 N. This suggests that better estimates of contact stiffness could increase the controller's performance. Improved contact stiffness estimates might be gained via online estimation, or by using the statistics of stiffness in a particular type of environment.

8. Limitations

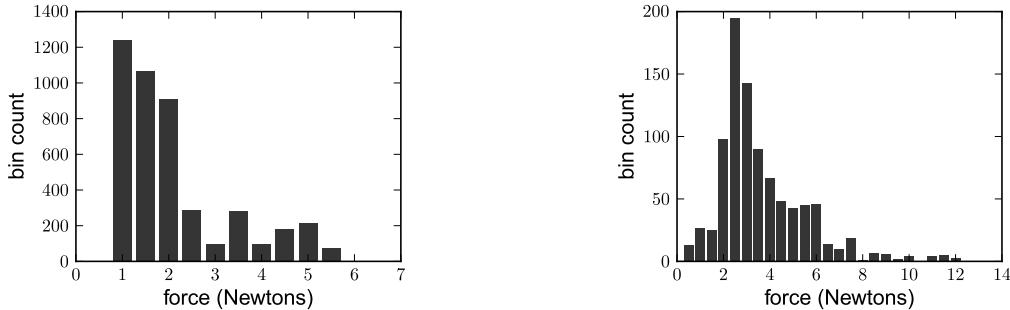
The stability of our controller is unproven. If our model predictive controller's quasi-static model were accurate, then the system would achieve static equilibrium in a single time-step. However, in addition to various linear approximations, the quasi-static model ignores dynamics and other



(a) Reaching to a goal location in foliage with multiple contacts along the arm. The forearm and 3D printed cover for the wrist are approximately outlined in red. The goal location is vertically below the blue bulb, and is the cyan circle in the visualization.



(b) Reaching to a goal location (green) through the opening of a cinder block.



(c) Histogram of contact forces while reaching to a goal location in foliage (left), and through the opening of the cinder block (right).

Fig. 19. Cody reaching to a goal location in realistic conditions using its forearm tactile sensor, described in Section 7.2.1.

Table 3. Mean (std) over five trials of the execution time and maximum force for different values of the modeled contact stiffness while pushing into a compliant object.

	Modeled contact stiffness (N/m)				
	200	500	1000	3000	5000
Mean (std) execution time	3.44 s (0.06 s)	3.49 s (0.05 s)	3.66 s (0.04 s)	4.49 s (0.19 s)	5.42 s (0.15 s)
Mean (std) max force	3.55 N (0.17 N)	3.87 N (0.15 N)	3.78 N (0.19 N)	3.65 N (0.17 N)	3.69 N (0.40 N)

real-world factors that influence stability. Empirically, we have found that the system behaves in a stable manner, but

formal analysis, potentially using the methods of Mayne et al. (2000), could be beneficial.

Our contact model consists of a linear spring, which is computationally favorable, but predicts adhesive forces when breaking contact.

Our controller places no penalty on a predicted contact force below f_{ci}^{thresh} , and has a hard inequality constraint that prevents higher predicted forces. It may be advantageous to soften this constraint.

Our controller also ignores dynamics. The resulting quasi-static model is suitable for slow motions, which is reasonable for haptic reaching since a collision could occur at any moment. Taking dynamics into account may result in better performance and higher speeds.

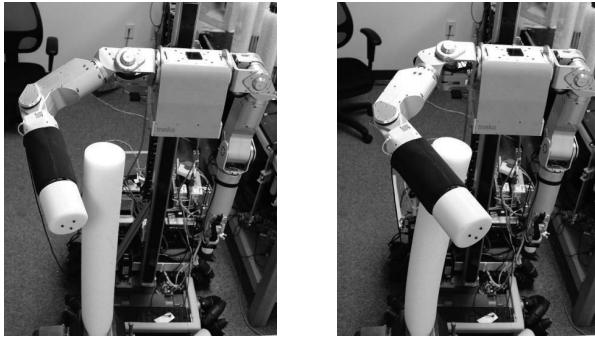


Fig. 20. Initial configuration of the arm (left) and a final configuration after it has reached the goal by bending the styrofoam cylinder (right).

The high-level control we used in this paper may not be appropriate for some types of clutter. The design of high-level controllers, and associated representations with memory, merits further inquiry. For example, surface following while exploring an environment may be beneficial.

We set the stiffnesses of the joints to constant low values. We did not investigate how performance changes with different values. A related open question is how to initialize and adapt the controller parameters such as the joint stiffnesses, stiffnesses of the springs in the contact model and starting configuration of the arm, given a robot, an environment, and a task.

9. Conclusion

We have presented our approach to manipulation, which from the outset emphasizes the benefits of making contact with the world. We have also presented a new controller and evaluated it with respect to the task of reaching in high clutter. In three distinct testbeds over numerous experiments, our controller with whole-arm tactile sensing outperformed other controllers. With real robots, it outperformed a baseline controller with the same low-stiffness actuation. In simulations with moderate clutter, it outperformed the same model predictive controller using no force sensing and using per-link force-torque sensing, with the relative benefits of whole-arm tactile sensing increasing as the clutter increased. Moreover, as the number of retries increased, the success rate quickly rose towards our estimate of the optimal success rate. This suggests that detailed models with long time horizon planning may not be necessary when reaching in clutter, which is similar in spirit to research by Byl and Tedrake (2008) on bipedal walking. An open question is the extent to which our controller can be used for other tasks, robots, and environments. We expect that controllers designed for whole-arm contact could eventually serve as general purpose controllers upon which other capabilities, such as avoiding contact, would be built.

Funding

This work was supported by DARPA Maximum Mobility and Manipulation (grant number W911NF-11-1-603).

Acknowledgements

This work benefitted from discussions with Magnus B Egerstedt, Harvey Lipkin, Mike Stilman, and James M Rehg. We thank Mark Cutkosky and the Stanford Biomimetics and Dexterous Manipulation Lab for their contributions to the forearm tactile sensor. We thank Jeff Weber, Andy Metzger, Benjamin Valenti, Pierre-Luc Bacon, Robert Kelbley, and John Ulmen for contributions to the forearm tactile sensor hardware and software. We also thank the anonymous reviewers for their valuable feedback and suggestions.

References

- Abbeel P, Coates A and Ng A (2010) Autonomous helicopter aerobatics through apprenticeship learning. *International Journal of Robotics Research* 29: 1608–1639.
- Albu-Schaffer A, Ott C, Frese U and Hirzinger G (2003) Cartesian impedance control of redundant robots: Recent results with the DLR-light-weight-arms. In: *IEEE international conference on robotics and automation (ICRA '03)*, 14–19 September 2003, vol. 3, pp. 3704–3709. Piscataway: IEEE Press.
- Alexander R (1990) Three uses for springs in legged locomotion. *International Journal of Robotics Research* 9: 53–61.
- Bellingham J, Richards A and How J (2002) Receding horizon control of autonomous aerial vehicles. In: *American control conference*, vol. 5, pp. 3741–3746. Piscataway: IEEE Press.
- Bianchi L (2007) Mechanotransduction: Touch and feel at the molecular level as modeled in *caenorhabditis elegans*. *Molecular Neurobiology* 36(3): 254–271.
- Bicchi A (1993) Force distribution in multiple whole-limb manipulation. In: *IEEE international conference on robotics and automation (ICRA '93)*, 2–6 May 1993, vol. 2, pp. 196–201. Piscataway: IEEE Press.
- Bicchi A and Kumar V (2000) Robotic grasping and contact: A review. In: *IEEE international conference on robotics and automation (ICRA '00)*. Vol. 1, pp. 238–353. Piscataway: IEEE Press.
- Bicchi A, Salisbury J and Brock D (1993) Contact sensing from force measurements. *International Journal of Robotics Research* 12: 249–262.
- Bierbaum A, Asfour T and Dillmann R (2009) Dynamic potential fields for dexterous tactile exploration. *Human Centered Robot Systems* 6: 23–31.
- Boyd S and Vandenberghe L (2004) *Convex Optimization*. Cambridge: Cambridge University Press.
- Buerger S and Hogan N (2007) Complementary stability and loop shaping for improved human–robot interaction. *IEEE Transactions on Robotics* 23(2): 232–244.
- Byl K and Tedrake R (2008) Approximate optimal control of the compass gait on rough terrain. In: *IEEE international conference on robotics and automation (ICRA '08)*. Pasadena, CA, 19–23 May 2008, p. 1258–1263. Piscataway: IEEE Press.
- Catania K (1999) A nose that looks like a hand and acts like an eye: the unusual mechanosensory system of the star-nosed mole. *Journal of Comparative Physiology A: Neuroethology, Sensory, Neural, and Behavioral Physiology* 185: 367–372.

- Chitta S, Sturm J, Piccoli M and Burgard W (2011) Tactile sensing for mobile manipulation. *IEEE Transactions on Robotics* 27(99): 1–11.
- Cottle R and Dantzig G (1968) Complementary pivot theory of mathematical programming. *Linear Algebra and its Applications* 1: 103–125.
- Cottle R, Pang J and Stone R (2009) *The Linear Complementarity Problem*. Philadelphia, PA: Society for Industrial & Applied Mathematics.
- Cutkosky MR and Ulmen JV (2012) Dynamic tactile sensing. In: Santos V and Balasubramanian R (eds) *The Human Hand: A Source of Inspiration for Robotic Hands*. Berlin: Springer, ch. 24.
- De Luca A, Albu-Schaffer A, Haddadin S and Hirzinger G (2006) Collision detection and safe reaction with the DLR-III lightweight manipulator arm. In: *IEEE/RSJ international conference on intelligent robots and systems (IROS '06)*, Beijing, China, October 2006, pp. 1623–1630. Piscataway: IEEE Press.
- De Luca A and Mattone R (2004) An adapt-and-detect actuator FDI scheme for robot manipulators. In: *IEEE international conference on robotics and automation (ICRA '04)*, 26 April–1 May, vol. 5, p. 4975–4980. Piscataway: IEEE Press.
- De Schutter J, Bruyninckx H, Dutré S, De Geeter J, Katupitiya J, Demey S et al. (1999) Estimating first-order geometric parameters and monitoring contact transitions during force-controlled compliant motion. *International Journal of Robotics Research* 18: 1161–1184.
- Diankov R and Kuffner J (2008) Openrave: A planning architecture for autonomous robotics. Report, Robotics Institute, USA.
- Dogar M, Hemrajani V, Leeds D, Kane B and Srinivasa S (2010) Proprioceptive localization for mobile manipulators. Report, Carnegie Mellon University, USA.
- Dogar M and Srinivasa S (2011) A framework for push-grasping in clutter. *Robotics: science and systems*. Cambridge, MA: The MIT Press.
- Dominy N (2004) Fruits, fingers, and fermentation: the sensory cues available to foraging primates. *Integrative and Comparative Biology* 44: 295–303.
- Duchaine V, Bouchard S and Gosselin C (2007) Computationally efficient predictive robot control. *IEEE/ASME Transactions on Mechatronics* 12(5): 570–578.
- Dupont P and Yamajako S (1994) Jamming and wedging in constrained rigid-body dynamics. In: *IEEE international conference on robotics and automation (ICRA '94)*, San Diego, CA, USA, 8–13 May, 1994, vol. 3, pp. 2349–2354. Piscataway: IEEE Press.
- Eberman B (1989) *Whole-arm manipulation: kinematics and control*. MSc Thesis, Massachusetts Institute of Technology, USA.
- Eberman B and Salisbury J (1990) Determination of manipulator contact information from joint torque measurements. In: *1st international symposium on experimental robotics*, vol. 139, pp. 463–473. New York: ACM Press.
- Edsinger A and Kemp CC (2007a) Human–robot interaction for cooperative manipulation: Handing objects to one another. In: *16th IEEE international symposium on robot and human interactive communication (RO-MAN '07)*, Jeju, Korea, 26–29 August, pp. 1167–1172. Piscataway: IEEE Press.
- Edsinger A and Kemp CC (2007b) Two arms are better than one: A behavior-based control system for assistive bimanual manipulation. In: *13th international conference on advanced robotics (ICAR '07)*, vol. 370, pp. 345–355.
- Erez T, Tassa Y and Todorov E (2011) Infinite-horizon model predictive control for periodic tasks with contacts. In: *Robotics: science and systems*. Cambridge, MA: The MIT Press, pp. 73–80.
- Escande A and Kheddar A (2009) Contact planning for acyclic motion with tasks constraints. In: *IEEE/RSJ international conference on intelligent robots and systems (IROS '09)*, St Louis, MO, USA, 10–15 October 2009, pp. 435–440. Piscataway: IEEE Press.
- Featherstone R and Orin DE (2008) Dynamics. In: Siciliano B and Khatib O (eds) *Handbook of Robotics*. New York: Springer, ch. 2.
- Frank B, Stachniss C, Abdo N and Burgard W (2011) Using Gaussian process regression for efficient motion planning in environments with deformable objects. Workshops at: *25th AAAI conference on artificial intelligence*, San Francisco, CA, USA, August 7–11, 2011.
- From P, Gravdahl J, Lillehagen T and Abbeel P (2011) Motion planning and control of robotic manipulators on seaborne platforms. *Control Engineering Practice* 19: 809–819.
- Ganesh G, Jarrasse N, Haddadin S, Albu-Schaeffer A and Burdet E (2012) A versatile biomimetic controller for contact tooling and haptic exploration. In: *IEEE international conference on robotics and automation (ICRA '12)*, Saint Paul, MN, USA, 14–18 May 2012, pp. 3329–3334. Piscataway: IEEE Press.
- Garcia C, Prett D and Morari M (1989) Model predictive control: Theory and practice—a survey. *Automatica* 25: 335–348.
- Garcia M, Chatterjee A, Ruina A and Coleman M (1998) The simplest walking model: Stability, complexity, and scaling. *Journal of Biomechanical Engineering* 120: 281–288.
- Grosso M, Quach R and Badler N (1989) Anthropometry for computer animated human figures. *State-of-the Art in Computer Animation*. pp. 83. Berlin: Springer.
- Gu X and Ballard D (2006) An equilibrium point based model unifying movement control in humanoids. In: *Robotics: science and systems*.
- Haddadin S, Albu-Schaffer A, De Luca A and Hirzinger G (2008) Collision detection and reaction: A contribution to safe physical human–robot interaction. In: *IEEE/RSJ international conference on intelligent robots and systems (IROS '08)*, Nice, Italy, 22–26 September 2008, pp. 3356–3363. Piscataway: IEEE Press.
- Haddadin S, Belder R and Albu-Schäffer A (2011) Dynamic motion planning for robots in partially unknown environments. In: *IFAC world congress*, Milan, Italy, 28 August–2 September, vol. 18, pp. 6842–6850.
- Hauser K, Bretl T and Latombe J (2005) Non-gaited humanoid locomotion planning. In: *IEEE conference on humanoid robots*, Tsukuba, Japan, December 2005, pp. 7–12. Piscataway: IEEE Press.
- Hersch M and Billard A (2006) A biologically-inspired controller for reaching movements. In: *IEEE/RAS-EMBS international conference on biomedical robotics and biomechatronics (BIOROB '06)*, Pisa, Italy, 20–22 February 2006, pp. 1067–1072. Piscataway: IEEE Press.
- Hogan N (1984) Adaptive control of mechanical impedance by coactivation of antagonist muscles. *IEEE Transactions on Automatic Control* 29: 681–690.
- Hogan N (1988) On the stability of manipulators performing contact tasks. *IEEE Journal of Robotics and Automation* 4(6): 677–686.

- Hogan N and Buerger S (2005). Impedance and interaction control. In: Siciliano B and Khatib O (eds) *Handbook of Robotics*. New York: Springer, ch. 19.
- Hsiao K, Chitta S, Ciocarlie M and Jones E (2010a) Contact-reactive grasping of objects with partial shape information. In: *IEEE/RSJ international conference on intelligent robots and systems (IROS '10)* Taipei, Taiwan, 18–22 October 2010, vol.10, pp. 1228–1235. Piscataway: IEEE Press.
- Hsiao K, Kaelbling L and Lozano-Pérez T (2010b) Task-driven tactile exploration. *Robotics: science and systems*.
- Hsiao K and Lozano-Pérez T (2006) Imitation learning of whole-body grasps. In: *IEEE/RSJ international conference on intelligent robots and systems (IROS '06)*, Beijing, China, 9–15 October 2006, pp. 5657–5662. Piscataway: IEEE Press.
- Ivaldi S, Fumagalli M, Nori F, Baglietto M, Metta G and Sandini G (2010) Approximate optimal control for reaching and trajectory planning in a humanoid robot. In: *IEEE/RSJ international conference on intelligent robots and systems (IROS '10)* Taipei, Taiwan, 18–22 October, pp. 1290–1296. Piscataway: IEEE Press.
- Iwaniuk A and Whishaw I (1999) How skilled are the skilled limb movements of the raccoon (*procyon lotor*)? *Behavioural Brain Research* 99(1): 35–44.
- Pratt JPD, Chew M and Pratt G (2001) Virtual model control: An intuitive approach for bipedal locomotion. *International Journal of Robotics Research* 20(2): 129–143.
- Jain A and Kemp CC (2009a) Behavior-based door opening with equilibrium point control. RSS Workshop Mobile Manipulation in Human Environments at: *Robotics: science and systems*.
- Jain A and Kemp CC (2009b) Pulling open novel doors and drawers with equilibrium point control. In: *IEEE conference on humanoid robots*, Paris, France, 7–10 December, pp. 498–505. Piscataway: IEEE Press.
- Jain A and Kemp CC (2010a) EL-E: An assistive mobile manipulator that autonomously fetches objects from flat surfaces. *Autonomous Robots* 28: 45–64.
- Jain A and Kemp CC (2010b) Pulling open doors and drawers: coordinating an omni-directional base and a compliant arm with equilibrium point control. In: *IEEE international conference on robotics and automation (ICRA '10)*, Anchorage, AK, USA, 3–7 May 2010, pp. 1807–1814. Piscataway: IEEE Press.
- Johnson KL (1985) *Contact Mechanics*. Cambridge: Cambridge University Press, ch. 4
- Kaneko M and Tanie K (1994) Contact point detection for grasping an unknown object using self-posture changeability. *IEEE Transactions on Robotics and Automation* 3: 355–367.
- Kao I, Lynch K and Burdick JW (2008). Contact modeling and manipulation. In: Siciliano B and Khatib O (eds) *Handbook of Robotics*. New York: Springer, ch. 27.
- Kavraki LE and LaValle SM (2008) Motion planning. In: Siciliano B and Khatib O (eds) *Handbook of Robotics*. New York: Springer, ch. 5.
- Khatib O (1987) A unified approach for motion and force control of robot manipulators: The operational space formulation. *IEEE Journal of Robotics and Automation* 3: 43–53.
- Killpack M, Deyle T, Anderson C and Kemp C (2010) Visual odometry and control for an omnidirectional mobile robot with a downward-facing camera. In: *IEEE/RSJ international conference on intelligent robots and systems (IROS '10)*, Taipei, Taiwan, 18–22 October 2010, pp. 139–146. Piscataway: IEEE Press.
- Kraft D, Bierbaum A, Kjaergaard M, Ratkevicius J, Kjaer-Nielsen A, Ryberg C et al. (2009) Tactile object exploration using cursor navigation sensors. In: *EuroHaptics symposium on haptic interfaces for virtual environment and teleoperator systems*, Salt Lake City, UT, USA, 18–20 March 2009, pp. 296–301. Piscataway: IEEE Press.
- Kulchenko P and Todorov E (2011) First-exit model predictive control of fast discontinuous dynamics: Application to ball bouncing. In: *IEEE international conference on robotics and automation (ICRA '11)* Shanghai, China, 9–13 May 2011, pp. 2144–2151. Piscataway: IEEE Press.
- LaValle S and Kuffner J (2001) Randomized kinodynamic planning. *International Journal of Robotics Research* 20(5): 378–400.
- Lederman SJ and Klatzky RL (2009) Haptic perception: A tutorial. *Attention, Perception and Psychophysics* (71): 1439–1459.
- Leeper A, Hsiao K, Ciocarlie M, Takayama L and Gossow D (2012) Strategies for human-in-the-loop robotic grasping. In: *ACM/IEEE international conference on human robot interaction*, New York, NY, USA, pp. 1–8. New York: ACM Press.
- Legagne S, Kheddar A and Yoshida E (2011) Generation of optimal dynamic multi-contact motions: Application to humanoid robots. *IEEE Transactions on Robotics*, submitted.
- Lozano-Pérez T (1987) A simple motion-planning algorithm for general robot manipulators. *IEEE Transactions on Robotics and Automation* 3: 224–238.
- Lumpkin EA, Marshall KL and Nelson AM (2010) The cell biology of touch. *Journal of Cell Biology* 191(2): 237–248.
- Maladen R, Ding Y, Umbanhowar P, Kamor A and Goldman D (2010) Biophysically inspired development of a sand-swimming robot. *Robotics: science and systems*.
- Manchester I, Mettin U, Iida F and Tedrake R (2011) Stable dynamic walking over uneven terrain. *International Journal of Robotics Research* 3: 265–279.
- Mason M (2001) *Mechanics of Robotic Manipulation*. Cambridge: MIT Press.
- Mason M, Rodriguez A, Srinivasa S and Vazquez A (2011) Autonomous manipulation with a general-purpose simple hand. *International Journal of Robotics Research* 31: 688–703.
- Mayne D, Rawlings J, Rao C and Scokaert P (2000) Constrained model predictive control: Stability and optimality. *Automatica* 36(6): 789–814.
- McClave J, Benson P and Sincich T (2008) *Statistics for Business and Economics*. New Jersey: Pearson Education.
- McKenna J, Anhalt D, Bronson F, Brown H, Schwerin M, Shammas E et al. (2008) Toroidal skin drive for snake robot locomotion. In: *IEEE international conference on robotics and automation (ICRA '08)*, Pasadena, CA, USA, 19–23 May 2008, pp. 1150–1155. Piscataway: IEEE Press.
- Metta G, Natale L, Nori F and Sandini G (2011) Force control and reaching movements on the ICUB humanoid robot. *International symposium on robotics research*.
- Migliore S (2009) *The role of passive joint stiffness and active knee control in robotic leg swinging: Applications to dynamic walking*. PhD Thesis, Georgia Institute of Technology, USA.
- Migliore S, Brown E and DeWeerth S (2005) Biologically inspired joint stiffness control. In: *IEEE international conference on robotics and automation (ICRA '05)*, 18–22 April 2005, pp. 4508–4513. Piscataway: IEEE Press.

- Morari M and Lee JH (1999) Model predictive control: past, present and future. *Computers & Chemical Engineering* 23(4–5): 667–682.
- Murray R, Li Z and Sastry S (1994) *A Mathematical Introduction to Robotic Manipulation*. New York: CRC Press.
- Natale L and Torres-Jara E (2006) A sensitive approach to grasping. In: *International workshop on epigenetic robotics*, Paris, France, 20–22 September 2006, pp. 87–94.
- Okamura A and Cutkosky M (2001) Feature-guided exploration with a robotic finger. In: *IEEE international conference on robotics and automation (ICRA '01)*, vol. 1, pp. 589–596. Piscataway: IEEE Press.
- Okamura A and Cutkosky M (1999) Haptic exploration of fine surface features. In: *IEEE international conference on robotics and automation (ICRA '99)*, Detroit, MI, USA, 10–15 May, vol. 4, pp. 2930–2936. Piscataway: IEEE Press.
- Park J and Khatib O (2008) Robot multiple contact control. *Robotica* 26(5): 667–677.
- Pastor P, Righetti L, Kalakrishnan M and Schaal S (2011) Online movement adaptation based on previous sensor experiences. In: *IEEE/RSJ international conference on intelligent robots and systems (IROS '11)* San Francisco, CA, 25–30 September 2011, pp. 365–371. Piscataway: IEEE Press.
- Patil S, van den Berg J and Alterovitz R (2011) Motion planning under uncertainty in highly deformable environments. In: *Robotics: science and systems*: 242–248. Cambridge, MA: The MIT Press.
- Petrovskaya A and Khatib O (2011) Global localization of objects via touch. *IEEE Transactions on Robotics* 27(3): 569–585.
- Petrovskaya A, Park J and Khatib O (2007) Probabilistic estimation of whole body contacts for multi-contact robot control. *IEEE international conference on robotics and automation (ICRA '07)*, Rome, Italy, 10–14 April 2007, pp. 568–573. Piscataway: IEEE Press.
- Pezzementi Z, Plaku E, Reyda C and Hager G (2011) Tactile-object recognition from appearance information. *IEEE Transactions on Robotics* 27(3): 473–487.
- Platt Jr R, Fagg A and Grupen R (2003) Extending fingertip grasping to whole body grasping. In: *IEEE international conference on robotics and automation (ICRA '03)*, 14–16 September 2003, vol. 2, pp. 2677–2682. Piscataway: IEEE Press.
- Platt Jr R, Permenter F and Pfeiffer J (2011) Using Bayesian filtering to interpret tactile data during flexible materials manipulation. *IEEE Transactions on Robotics* 27: 586–598.
- Pratt G (2002) Low impedance walking robots 1. *Integrative and Comparative Biology* 42(1): 174–181.
- Pratt G and Williamson M (1995) Series elastic actuators. In: *IEEE/RSJ international conference on intelligent robots and systems (IROS '95)*, Pittsburgh, PA, USA, 5–6 August 1995, vol. 1, pp. 399–406. Piscataway: IEEE Press.
- Prescott T, Pearson M, Fox C, Evans M, Mitchinson B, Anderson S et al. (2010) Towards biomimetic vibrissal tactile sensing for robot exploration, navigation, and object recognition in hazardous environments. In: *4th international workshop on robotics for risky interventions and environmental surveillance-maintenance (RISE '10)*.
- Quigley M, Gerkey B, Conley K, Faust J, Foote T, Leibs J et al. (2009) ROS: An open-source robot operating system. ICRA Open-Source Software workshop at: *IEEE international conference on robotics and automation (ICRA '09)*. Piscataway: IEEE Press.
- Raibert M, Blankespoor K, Nelson G, Playter R et al. (2008) Bigdog, the rough-terrain quadruped robot. In: *17th world congress. The International Federation of Automatic Control*, Seoul, South Korea, July 6–11, pp. 10822–10825.
- Raibert M and Craig J (1981) Hybrid position/force control of manipulators. *Journal of Dynamic Systems, Measurement, and Control* 102(127): 126–133.
- Rodriguez S, Lien J and Amato N (2006) Planning motion in completely deformable environments. In: *IEEE international conference on robotics and automation (ICRA '06)*, Orlando, FL, USA, 15–19 May 2006, pp. 1067–1079. Piscataway: IEEE Press.
- Romano J, Hsiao K, Niemeyer G, Chitta S and Kuchenbecker K (2011) Human-inspired robotic grasp control with tactile sensing. *IEEE Transactions on Robotics* 27: 1067–1079.
- Salisbury JJ (1984) Interpretation of contact geometries from force measurements. In: *IEEE international conference on robotics and automation (ICRA '84)*, March 1984, vol. 1, pp. 240–247. Piscataway: IEEE Press.
- Saranli U, Buehler M and Koditschek D (2001) Rhex: A simple and highly mobile hexapod robot. *International Journal of Robotics Research* 20: 616–631.
- Saxena A, Driemeyer J and Ng A (2008) Robotic Grasping of Novel Objects using Vision. *International Journal of Robotics Research* 27(2): 157–173.
- Sentis L and Khatib O (2005) Synthesis of whole-body behaviors through hierarchical control of behavioral primitives. *International Journal of Humanoid Robotics* 2: 505–518.
- Sentis L, Park J and Khatib O (2010) Compliant control of multicontact and center-of-mass behaviors in humanoid robots. *IEEE Transactions on Robotics* 26(3): 483–501.
- Shadmehr R (1993) Control of equilibrium position and stiffness through postural modules. *Journal of Motor Behavior* 25: 228–241.
- Srinivasa S, Ferguson C, Helfrich D, Berenson D, Collet A, Diankov R et al. (2009) Herb: A home exploring robotic butler. *Autonomous Robots* 28: 5–20.
- Stilman M, Nishiwaki K, Kagami S and Kuffner J (2007a) Planning and executing navigation among movable obstacles. *Advanced Robotics* 21(14): 1617–1634.
- Stilman M, Schamburek J, Kuffner J and Asfour T (2007b) Manipulation planning among movable obstacles. In: *IEEE international conference on robotics and automation (ICRA '07)*, Rome, Italy, 10–14 April 2007, pp. 3327–3332. Piscataway: IEEE Press.
- Stulp F, Kresse I, Maldonado A, Ruiz F, Fedrizzi A and Beetz M (2009) Compact models of human reaching motions for robotic control in everyday manipulation tasks. In: *IEEE international conference on development and learning*, Shanghai, China, 5–7 June, pp. 1–7. Piscataway: IEEE Press.
- Tanaka H, Kushihama K, Ueda N and Hirai S (2003) A vision-based haptic exploration. In: *IEEE international conference on robotics and automation (ICRA '03)*, 14–16 September, vol. 3, pp. 3441–3448. Piscataway: IEEE Press.
- Tilley A and Dreyfuss H (2001) *The Measure of Man and Woman: Human Factors in Design*. New York: Wiley.
- Whitney D (1982) Quasi-static assembly of compliantly supported rigid parts. *Journal of Dynamic Systems, Measurement, and Control* 104: 65–77.
- Whitney D and Nevins J (1979) What is the remote center compliance (RCC) and what can it do. In: *9th international symposium on industrial robots*, pp. 135–152.

- Wieber P (2006) Trajectory free linear model predictive control for stable walking in the presence of strong perturbations. In: *IEEE-RAS international conference on humanoid robots*, Genoa, Italy, 4–6 December, pp. 137–142. Piscataway: IEEE Press.
- Williamson M (1996) Postural primitives: Interactive behavior for a humanoid robot arm. In: *4th international conference on simulation of adaptive behavior*, pp. 124–131. Cambridge, MA: The MIT Press.
- Williamson M (1999) *Robot arm control exploiting natural dynamics*. PhD Thesis, Massachusetts Institute of Technology, USA.

Appendix: Index to Multimedia Extensions

The multimedia extension page is found at <http://www.ijrr.org>

Table of Multimedia Extensions

Extension	Type	Description
1	Video	“Simple” impedance controller and low stiffness at the joints for the robot Cody, described in Section 3.2.1.
2	Video	First example in software simulation where MPC with feedback from force-torque sensor fails but MPC with tactile sensing succeeds, described in Section 6.3.2.
3	Video	Second example in software simulation where MPC with feedback from force-torque sensor fails but MPC with tactile sensing succeeds, described in Section 6.3.2.
4	Video	Additional high-level control layer in a simplified environment in software simulation, described in Section 6.4.
5	Video	Test of MPC as part of a fully autonomous system within the hardware-in-the-loop testbed, described in Section 7.1.2.
6	Video	Illustrative example of reaching in foliage using the real forearm tactile sensor, described in Section 7.2.1.
7	Video	Illustrative example of reaching through the opening of a cinder block the real forearm tactile sensor, described in Section 7.2.1.
8	Video	MPC vs. baseline controller in foliage using forearm tactile sensor, described in Section 7.2.2.
9	Code	Code, instructions and data to reproduce results within the software simulation testbed (Section 6).