REPORT DOCUMENTATION PAGE	Form Approved OMB NO. 0704-0188

1. REPORT DATE (DD-MM-YYYY) 16-06-2014	2. REPORT TYPE Final Report	3. DATES COVERED (From - To) 11-Mar-2011 - 10-Mar-2014

4. TITLE AND SUBTITLE Multi-contact Variable-Compliance Manipulation in Extreme Clutter	5a. CONTRACT NUMBER W911NF-11-1-0063
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER 0620BK
6. AUTHORS Charles C. Kemp	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAMES AND ADDRESSES Georgia Tech Research Corporation 505 Tenth Street NW Atlanta, GA 30332 -0420	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS (ES) U.S. Army Research Office P.O. Box 12211 Research Triangle Park, NC 27709-2211	10. SPONSOR/MONITOR'S ACRONYM(S) ARO
	11. SPONSOR/MONITOR'S REPORT NUMBER(S) 59660-MS-DRP.10

12. DISTRIBUTION AVAILIBILITY STATEMENT

Approved for Public Release; Distribution Unlimited

13. SUPPLEMENTARY NOTES

The views, opinions and/or findings contained in this report are those of the author(s) and should not contrued as an official Department of the Army position, policy or decision, unless so designated by other documentation.

14. ABSTRACT

We have developed new foundational capabilities for robot manipulation that assume contact across the entire manipulator is inevitable and desirable. Our approach makes use of compliant actuation and full-body force-sensing skin. We developed force-sensitive skin, low-level control algorithms, mid-level control algorithms, and planners that enable robots to reach to locations in extreme clutter, such as foliage and rubble, while haptically generating 3D maps of their surroundings. We have performed experiments with software simulated robots with skin, a hardware-in-the-loop system that simulates skin for a real robot, and real robots with real force-sensing skin

15. SUBJECT TERMS

manipulation, haptics, skin, robotics, clutter

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	15. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON Charles Kemp
a. REPORT UU	b. ABSTRACT UU	c. THIS PAGE UU	UU		19b. TELEPHONE NUMBER 404-385-8192

Report Title

Multi-contact Variable-Compliance Manipulation in Extreme Clutter

ABSTRACT

We have developed new foundational capabilities for robot manipulation that assume contact across the entire manipulator is inevitable and desirable. Our approach makes use of compliant actuation and full-body force-sensing skin. We developed force-sensitive skin, low-level control algorithms, mid-level control algorithms, and planners that enable robots to reach to locations in extreme clutter, such as foliage and rubble, while haptically generating 3D maps of their surroundings. We have performed experiments with software simulated robots with skin, a hardware-in-the-loop system that simulates skin for a real robot, and real robots with real force-sensing skin covering their arms. In these tests, our novel control systems have enabled robots to perform qualitatively new tasks and outperformed baseline systems both in terms of success rate and keeping contact forces low. Our most recent control system also substantially outperforms our original control system in terms of time to complete (i.e. speed), success rate, and contact forces. Our project has resulted in a new and broadly applicable approach to robot manipulation that enables robots to achieve dramatically improved real-world manipulation performance. We have also produced open source code and open hardware implementations using open standards.

Enter List of papers submitted or published that acknowledge ARO support from the start of the project to the date of this printing. List the papers, including journal references, in the following categories:

(a) Papers published in peer-reviewed journals (N/A for none)

Received		Paper
03/14/2014	5.00	A. Jain, M. D. Killpack, A. Edsinger, C. C. Kemp. Reaching in clutter with whole-arm tactile sensing, The International Journal of Robotics Research, (03 2013): 0. doi: 10.1177/0278364912471865
03/14/2014	6.00	Advait Jain, Charles C. Kemp. Improving robot manipulation with data-driven object-centric models of everyday forces, Autonomous Robots, (06 2013): 0. doi: 10.1007/s10514-013-9344-1
TOTAL:		2

Number of Papers published in peer-reviewed journals:

(b) Papers published in non-peer-reviewed journals (N/A for none)

Received	Paper
TOTAL:	

Number of Papers published in non peer-reviewed journals:

(c) Presentations

Presented at International Symposium on Robotics Research 2013. Presentation is in the attachment, after Q4 2013 and before the final technical report.

Number of Presentations: 1.00

Non Peer-Reviewed Conference Proceeding publications (other than abstracts):

Received Paper

TOTAL:

Number of Non Peer-Reviewed Conference Proceeding publications (other than abstracts):

Peer-Reviewed Conference Proceeding publications (other than abstracts):

Received		Paper
03/14/2014	3.00	Phillip M. Grice, Marc D. Killpack, Advait Jain, Sarvagya Vaish, Jeffrey Hawke, Charles C. Kemp. Whole-arm tactile sensing for beneficial and acceptable contact during robotic assistance, 2013 IEEE 13th International Conference on Rehabilitation Robotics (ICORR 2013). 24-JUN-13, Seattle, WA, USA. : ,
03/14/2014	4.00	T. Bhattacharjee, A. Jain, S. Vaish, M. D. Killpack, C. C. Kemp. Tactile sensing over articulated joints with stretchable sensors, 2013 World Haptics Conference (WHC 2013). 14-APR-13, Daejeon. : ,
06/02/2014	8.00	Bhattacharjee, Tapomayukh, Kapusta, Ariel, Rehg, James M., Kemp, Charles C.. Rapid Categorization of Object Properties from Incidental Contact with a Tactile Sensing Robot Arm, Humanoids. 15-OCT-13, . : ,
06/02/2014	7.00	Killpack, Marc D., Kemp, Charles C.. Fast Reaching in Clutter While Regulating Forces Using Model Predictive Control, Humanoids. 15-OCT-13, . : ,
08/31/2012	1.00	Tapomayukh Bhattacharjee, James M. Rehg, Charles C. Kemp. Haptic Classification and Recognition of Objects Using a Tactile Sensing Forearm, IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). 07-OCT-12, . : ,
TOTAL:		**5**

Number of Peer-Reviewed Conference Proceeding publications (other than abstracts):

(d) Manuscripts

Received	Paper

TOTAL:

Number of Manuscripts:

Books

Received	Book

TOTAL:

Received	Book Chapter

TOTAL:

Patents Submitted

Patents Awarded

Awards

Graduate Students

NAME	PERCENT_SUPPORTED	Discipline
Ari Kapusta	1.00	
Tapomayukh Bhattacharjee	0.50	
Daehyung Park	1.00	
Marc Killpack	1.00	
FTE Equivalent:	**3.50**	
Total Number:	**4**	

Names of Post Doctorates

NAME	PERCENT_SUPPORTED
FTE Equivalent:	
Total Number:	

Names of Faculty Supported

NAME	PERCENT_SUPPORTED	National Academy Member
Charles C. Kemp	0.04	
FTE Equivalent:	**0.04**	
Total Number:	**1**	

Names of Under Graduate students supported

NAME	PERCENT_SUPPORTED
FTE Equivalent:	
Total Number:	

Student Metrics

This section only applies to graduating undergraduates supported by this agreement in this reporting period

The number of undergraduates funded by this agreement who graduated during this period: 0.00

The number of undergraduates funded by this agreement who graduated during this period with a degree in science, mathematics, engineering, or technology fields:...... 0.00

The number of undergraduates funded by your agreement who graduated during this period and will continue to pursue a graduate or Ph.D. degree in science, mathematics, engineering, or technology fields:...... 0.00

Number of graduating undergraduates who achieved a 3.5 GPA to 4.0 (4.0 max scale):...... 0.00

Number of graduating undergraduates funded by a DoD funded Center of Excellence grant for Education, Research and Engineering:...... 0.00

The number of undergraduates funded by your agreement who graduated during this period and intend to work for the Department of Defense 0.00

The number of undergraduates funded by your agreement who graduated during this period and will receive scholarships or fellowships for further studies in science, mathematics, engineering or technology fields:...... 0.00

Names of Personnel receiving masters degrees

NAME

Total Number:

Names of personnel receiving PHDs

NAME
Marc Killpack
Total Number: **1**

Names of other research staff

NAME	PERCENT_SUPPORTED

FTE Equivalent:
Total Number:

Sub Contractors (DD882)

Inventions (DD882)

Scientific Progress

Technology Transfer

Multi-contact Variable Compliance Manipulation in Extreme Clutter

DARPA Maximum Mobility and Manipulation (M3)
Final ARO Report
Covers July 01, 2013 to March 10, 2014
Submitted: June 16, 2014

PI: Charles C. Kemp (Georgia Tech)
Co-PI: Aaron Edsinger (Meka Robotics)

1 Abstract

We have developed new foundational capabilities for robot manipulation that assume contact across the entire manipulator is inevitable and desirable. Our approach makes use of compliant actuation and full-body force-sensing skin. We developed force-sensitive skin, low-level control algorithms, mid-level control algorithms, and planners that enable robots to reach to locations in extreme clutter, such as foliage and rubble, while haptically generating 3D maps of their surroundings. We have performed experiments with software simulated robots with skin, a hardware-in-the-loop system that simulates skin for a real robot, and real robots with real force-sensing skin covering their arms. In these tests, our novel control systems have enabled robots to perform qualitatively new tasks and outperformed baseline systems both in terms of success rate and keeping contact forces low. Our most recent control system also substantially outperforms our original control system in terms of time to complete (i.e. speed), success rate, and contact forces. Our project has resulted in a new and broadly applicable approach to robot manipulation that enables robots to achieve dramatically improved real-world manipulation performance. We have also produced open source code and open hardware implementations using open standards.

2 Contents*

** Each item's pages are individually numbered. Numbers stated here are from document start.*

Multi-contact Variable Compliance Manipulation in Extreme Clutter

DARPA Maximum Mobility and Manipulation (M3)
Final Technical Report
Covers January 01, 2014 to March 10, 2014
Submitted: June 16, 2014

PI: Charles C. Kemp (Georgia Tech)
Co-PI: Aaron Edsinger (Meka Robotics)

1 Overview

In the first quarter of 2014 we made the following progress:

- Completed development of integrated system for dynamic reaching in clutter.
  - Combines our work on this project
    - Dynamic MPC controller
    - Tactile-based sensing
    - Online 3D haptic mapping of objects based on categorization of object properties
    - Learned initial conditions
    - Cost-based planning over sparse maps
- Tested system on the robot DARCI.
  - Results:
    - 79.19% success rate in a complex, unmodeled, cluttered foliage environment.
    - Performs complex, multi-step reaching behaviours on the robot DARCI.
    - Reaching behavior uses fastest, simplest behaviors first.
    - System haptically maps environment during reaching.
    - Geometric planning over sparse haptic map used when greedy reaching fails.
    - Improved success in more diverse situation compared with individual components.

Contents

2 An Integrated Robotics System for Haptically Reaching in Clutter with Whole Arm Tactile Sensing.

2.1 Introduction

Humans and other animals readily reach into complex environments without visually observing the detailed contents. During the day-to-day manipulation tasks, humans frequently come into 'incidental contact' with objects in their environments as shown in Fig. 1. By incidental contact, we mean any contact that occurs unintentionally while performing goal-directed manipulation tasks. Being able to reach into various environments without the need of avoiding contact with objects, would be a generally useful capability for robots in a variety of application areas, including assistive robotics [1]. Within this tech report, we describe an integrated system for robotic control that enables a robot to reach locations in unmodeled, cluttered environments solely based on joint-angle, joint-torque, and tactile sensing (See. Fig. 2) from 'incidental contact'. The system builds on our previous research in a number of ways, including integrating a variety of system components, both published and unpublished. We designed our system to first use efficient, memory-free greedy

(a) (b) (c) (d)

Figure 1: Humans and animals frequently come into contact with the environment while reaching into clutter. (a) A raccoon reaches into a bird house to find eggs and young. (b) When noodling, people find catfish holes from which to pull fish out. (c)-(d) A person makes contact along his forearm while reaching for objects in a cluttered cabinet and refrigerator. (All images used with permission)

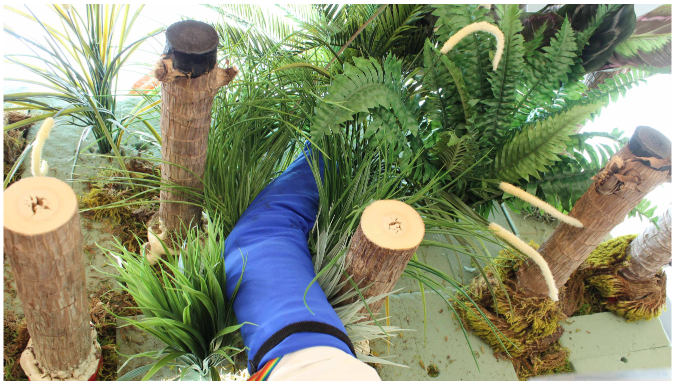


Figure 2: The DARCI Robot reaching through dense foliage using the integrated system described in this paper.

reaching followed, if necessary, by resource-intensive geometric planning using a map. A motivating intuition for this structure is the common human experience of reaching to a location without paying much attention, and then realizing that one needs to pay careful attention in order to succeed.

2.2 System Overview

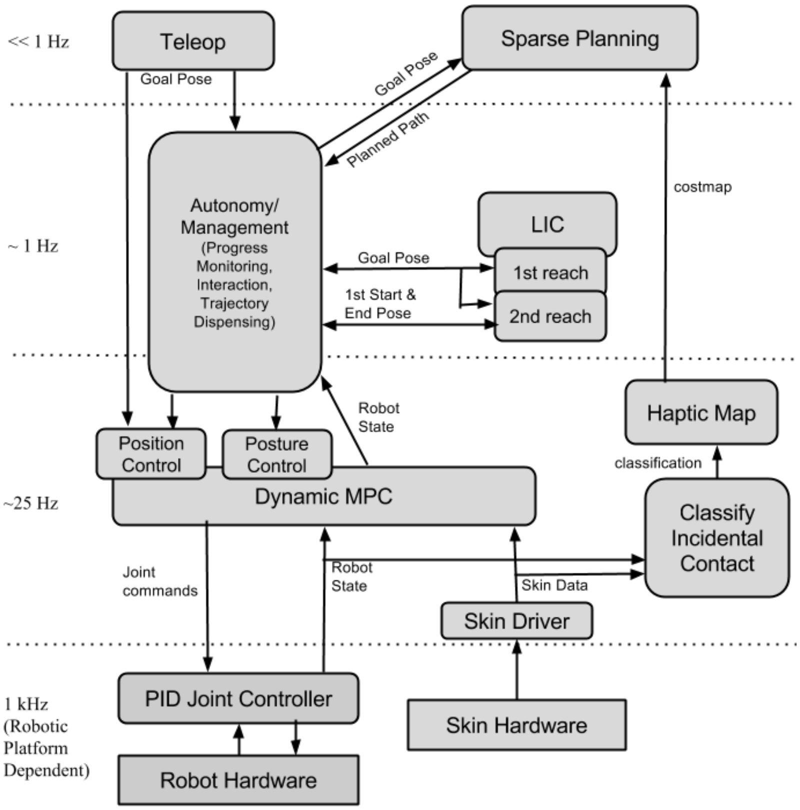


Figure 3: Block diagram showing the integrated system architecture. High update rate processes, such as low-level joint control, appear at the bottom of the diagram, while slower-updating processes are presented higher up. The teleoperation interface is only used to provide a single goal end-effector pose, after which the integrated system proceeds autonomously.

2.2.1 System Architecture

Figure 3 illustrates the architecture of our system. At all times, our system uses the newest version of our model predictive controller from [2] to control the robot at about 25 Hz. It attempts to reach either an end-effector pose or an arm configuration while keeping contact forces low. This model predictive controller runs on top of gravity compensation and an impedance controller that simulates low-stiffness visco-elastic springs at the robot's joints running at about 1 kHz.

When a desired end effector goal is received, the system first attempts to bring the arm to an initial configuration which has performed well in similar circumstances. The system then uses the model predictive controller to greedily reach to the goal location from this initial arm configuration. As we presented in [3], two greedy reaches from random locations can achieve over an 80% success rate, and we have found that using learned initial conditions (LIC) can result in a significantly higher success rate. Greedy reaching has the advantages of not requiring a map, relatively low computational requirements, and efficient use of redundant degrees of freedom. However, greedy reaching can become stuck in local minima and does not always succeed in finding a solution. For example, in [3] around 10% of the situations encountered were not reached after 5 greedy reaches from random initial arm configurations.

Algorithm 1 Integrated System Procedure.

```
Require: $GoalPose\ g$
    $HapticMap\ h \leftarrow blocked\_locations$
                                        ▷ Begin Contact Classification and Haptic Mapping
    $LIC1\_Pose\ lp \leftarrow$ LIC 1($g$)
    DYNAMIC MPC($g$)
    if at g then
        return
    else
        return $StuckPose\ s$
    end if
    DYNAMIC MPC($RetreatPose\ r$)
    LIC 2($g,\ lp,\ s$)
    DYNAMIC MPC($g$)
    if at g then
        return
    end if
    repeat
        $Path\ p \leftarrow$ PLAN($g, h$)
        DYNAMIC MPC($p$)
    until at g
```

In order to handle rare, but challenging, situations like these, our system makes use of geometric planning based on a map of locations that our tactile recognition system has estimated to be impassable. While computationally intensive, planning has the advantage of being able to eventually find solutions for situations requiring complex sequences of arm configurations. Our system plans based on a map it constructs during greedy reaching, and which it continues to update during planned reaches. If the robot becomes stuck while attempting to follow a planned sequence of arm configurations, the system replans using the current map. Using this method, the maps over which trajectories are planned are relatively sparsely populated with known obstacles, but initial work has shown this to be sufficient to produce useful behaviors from the controller. Besides having relatively large computational requirements, a disadvantage of the planning system is that, unlike greedy reaching, it does not reactively take advantage of the robot's redundant degrees of freedom, and instead needs to replan in the event of becoming stuck.

The pseudocode in Algorithm 1 provides an overview of the way in which the integrated system functions.

2.3 System Components

We now provide brief summaries of our system's components.

2.3.1 Dynamic MPC

Moving a robot arm in cluttered, unknown, and unmodeled workspaces can be difficult as interaction with obstacles can block paths and generate high contact forces. We use a model predictive control (MPC) controller that explicitly models the robot arm dynamics with tactile sensing to move the robot arm quickly and control contact forces as the arm moves towards its goal. We implemented an updated version of our dynamic model predictive control (MPC) controller from [2] that runs on our humanoid robot, DARCI, and that adds additional functionality to work with the various modules of our integrated system. Our dynamic MPC controller moves towards a designated goal position while keeping contact forces and worst-case, unexpected-impact forces low. We added an integral controller term to compensate for errors in the gravity compensation model of the robot. Gravity is not explicitly modeled in our dynamic model since we assumed that the low-level joint controllers were canceling it perfectly. To compensate for gravity, we introduced the following into the previous cost function:

$$\alpha \left\|\Delta \boldsymbol{x}_{des} - \boldsymbol{J}_{ee}(\boldsymbol{q}[t_0 + H_u + H_y] - \boldsymbol{q}[t_0]) - \boldsymbol{d}_{grav}\right\|^2 \tag{1}$$

where $\boldsymbol{d}_{grav}$ is a function that acts as an integral term in the controller.

$$\boldsymbol{d}_{grav} = \boldsymbol{f}(k_i \sum_{t=0}^{t_0}(\boldsymbol{x}_{des} - \boldsymbol{x}_{ee}[t]), \boldsymbol{x}_{des} - \boldsymbol{x}_{ee}[t_0]) \tag{2}$$

For the posture controller we use a slightly different version of the integral term to correct for error due to gravity. It is incorporated into the control system shown in Figure 4.

$$\boldsymbol{d}_{err} = \boldsymbol{f}(k_i \sum_{t=0}^{t_0}(\boldsymbol{q}_{goal} - \boldsymbol{q}[t]), \boldsymbol{q}_{goal} - \boldsymbol{q}[t_0]) \tag{3}$$

We introduced some straight forward anti-windup measures and saturation limits and made k_i very small. Furthermore, this term only becomes active when the end effector is within 8 cm of the desired goal location so as to avoid serious overshoot and high forces when we are stuck far from the goal. See the Q3 2013 Technical Report for additional details on changes to the controller, including details on the impulse-momentum model. Our MPC controller uses a control horizon of 3 and prediction horizon of 4 which gives the controller four time steps of control and predicts the arm output for 4 additional time steps over which it aims to minimize its cost function. We added the functionality to the controller to receive a joint configuration posture in addition to the option of a Cartesian end-effector goal location. To allow posture control we altered the cost function to use the difference from the desired joint configuration rather than end effector location from [2], as shown in Figure 4. In the posture-control version of our controller we removed the limit on the rate of change of contact forces to improve computational performance. Posture goals are the method by which goals are sent from the planner to the controller and the method by which the arm is extracted from clutter. Due to differences in the optimization between pose- and posture-control modules, and in order to keep each optimization as small as possible, two separate control modules are run in parallel throughout the demonstration, one for pose-control and another for posture-control. When one is active, the other is set to a waiting state, wherein it does not solve the optimization or send commands to the low-level joint controllers. This avoids conflicting commands to joint controllers and reduces the computational requirements of the control system.

$$\underset{\boldsymbol{\Delta q_{des}}}{\textbf{minimize}}$$

$$\alpha \left\| \Delta \boldsymbol{q}_{goal} - (\boldsymbol{q}[t_0+H+1] - \boldsymbol{q}[t_0]) - \boldsymbol{d}_{err} \right\|^2 \tag{4}$$

$$+\beta \sum_{t=t_0}^{t_0+H} \sum_{i=1}^{N} \max\left(\boldsymbol{n}_{c_i}^T \boldsymbol{K}_{c_i} \boldsymbol{J}_{c_i} (\boldsymbol{q}[t+1] - \boldsymbol{q}[t_0]) - (f_{threshold} - \left\| \boldsymbol{f}_i^{measured}[t_0] \right\|), \mathbf{0} \right) \tag{5}$$

$$+\kappa \sum_{t=t_0}^{t_0+H} \sum_{i=1}^{N} \max\left(abs(2\boldsymbol{M}(\boldsymbol{q})\dot{\boldsymbol{q}}[t+1]) - \boldsymbol{\tau}_{max} \Delta t_{impact}, \mathbf{0} \right) \tag{6}$$

$$+\mu \textstyle\sum_{t=t_0}^{t_0+H} \left\| \Delta \boldsymbol{q}_{des}[t] \right\|^2 \tag{7}$$

subject to : $(for\ t = t_0 \ldots t_0 + H)$

$$\begin{bmatrix} \dot{\boldsymbol{q}}[t+1] \\ \boldsymbol{q}[t+1] \end{bmatrix} = \boldsymbol{A}_d[t] \begin{bmatrix} \dot{\boldsymbol{q}}[t] \\ \boldsymbol{q}[t] \end{bmatrix} + \boldsymbol{B}_d[t] \begin{bmatrix} \boldsymbol{q}_{des}[t] \\ \sum_{i=1}^{N} \boldsymbol{J}_{c_i}^T \boldsymbol{f}_i^{measured}[t_0] \\ \boldsymbol{q}[t_0] \end{bmatrix} \tag{8}$$

$$\boldsymbol{q}_{des}[t+1] = \boldsymbol{q}_{des}[t] + \Delta \boldsymbol{q}_{des}[t] \tag{9}$$

$$\boldsymbol{q}[t+1] \leqq \boldsymbol{q}_{max} \tag{10}$$

$$\boldsymbol{q}[t+1] \geqq \boldsymbol{q}_{min} \tag{11}$$

$$abs(\Delta \boldsymbol{q}_{des}[t]) \leqq \Delta \boldsymbol{q}_{max,des} \tag{12}$$

Figure 4: The altered form of the controller used for joint configuration posture control

Nomenclature

$\alpha, \beta, \kappa, \mu$	Scalar weighting terms for the multi-objective cost function
t_0	Current time where state measurements are valid. Starting point of predictive model
H	Number of time steps in the prediction model
$\Delta \boldsymbol{q}_{goal}$	Desired change in joint configuration
$\boldsymbol{d}_{err}$	Error correcting integreal term Desired final joint configuratin
$f_{threshold}$	User-defined allowable contact force threshold
$\boldsymbol{n}_{c_i}$	Contact normal direction at contact i
$\boldsymbol{K}_{c_i}$	Cartesian stiffness matrix for contact i
$\boldsymbol{J}_{c_i}$	Geometric Jacobian at contact i
$\boldsymbol{q}_{min}$	Minimum joint angle limits
$\boldsymbol{q}_{max}$	Maximum joint angle limits
$\boldsymbol{\tau}_{max}$	Maximum allowable torque due to impact forces
Δt_{impact}	Time duration of an expected impact
$\boldsymbol{q}, \dot{\boldsymbol{q}}$	State variables of joint angle and velocity
$\boldsymbol{f}_i^{measured}$	Measured normal force for contact i
$\boldsymbol{q}_{des}$	Commanded joint angles that are sent to the joint impedance controller
$\Delta \boldsymbol{q}_{des}$	Change in commanded joint angles, this is the output of our MPC
$\boldsymbol{A}_d, \boldsymbol{B}_d$	Discrete time linear approximations of the system state space matrices
$\Delta \boldsymbol{q}_{max,des}$	Maximum allowable change in commanded joint angle

2.3.2 Learned Initial Conditions

In this section, we describe learning and prediction schemes for identifying good initial configurations during manipulation in clutter. We have shown that reaching a goal in clutter may require multiple attempts before succeeding [3]. However, if we can identify initial configurations which result in successful reaching, we can significantly decrease the number of required retries [4].

2.3.2.1 Learning initial conditions without detailed knowledge Prior to observing the environment in which manipulation is to take place, and without detailed knowledge of the environment, we define the problem of the selecting the best initial condition as

$$\begin{aligned} \underset{x_0}{\text{maximize}} \quad & P(x_\infty = g|x_0) \\ \text{subject to} \quad & x_0 \in \text{open space}, \end{aligned} \tag{13}$$

where $x_0 \in \Re^6$ is the initial pose of the end effector before beginning a greedy reaching behavior, $x_\infty \in \Re^3$ is the final stopping position, and $g \in \Re^3$ is the goal position. x_0 must satisfy joint constraints. In addition, we constrain x_0 to lie in open space outside the cluttered region of interest.

Given an environment v for which we only know the category c, the marginal probability density function of the selection problem is written as Eqn. 14. If the properties of v are similar to the environments V_c, which have been explored, we can approximate the marginal probability distribution as follows:

$$P(x_\infty = g|x_0) = \int_v P(x_\infty = g|x_0, v)dv \tag{14}$$

$$\approx \int_{V_c} P(x_\infty = g|x_0, v')P(v')dv', \tag{15}$$

where v' is a map in an experienced environment set V_c. Thus, given a goal from past trial experiences in the same or similar environments, we can predict the probability of the best condition. We will use 'LIC-1' (learning an initial condition for a first reach into a new cluttered environment) to denote the framework in Eqn. 15.

2.3.2.2 Learning initial condition with observations After one attempt, we have obtained observations o about the environment v, and we can adapt the initial condition to improve the probability of success. This problem can be written as

$$\begin{aligned} \underset{x_0}{\text{maximize}} \quad & P(x_\infty = g|x_0, o) \\ \text{subject to} \quad & x_0 \in \text{open space}, \end{aligned} \tag{16}$$

where x_0 denotes the restart condition and o denotes observed information from the previous trial. In this system, we define o as

$$o = \{x'_0, x'_\infty\}, \tag{17}$$

where x'_0 denote the previous initial condition and x'_∞ is the final position of the previous trial. Similar to LIC-1, we compute the marginal probability conditioned on the observation. We denote this second framework 'LIC-2.'

For the implementation in this paper, we trained the model using a large number of successful- and failed-trial samples in a simulation environment, shown as Fig. 5. This clutter includes 60 fixed-floating spheres, each with a 0.05 m radius, in a 0.5 m x 0.9 m x 0.6 m rectangular parallelepiped area in front of a simulated DARCI robot. The robot tries to reach to 15 grid-distributed goals of size 5 x 3 in 20 different clutters from 28 initial conditions. The goals were placed behind of a set of spheres on a vertical, rectangular plane 0.6 m wide and 0.3 m tall, at 0.15 m intervals. The initial positions were equally distributed on a vertical, rectangular plane 0.6 m wide and 0.3 m tall, at 0.1 m intervals. We ran 22,684 trials for the sampling of trials. Using simulated or real-world trials that more closely match the target environment would be likely to improve performance. Here, we used spheres in 3D as a generic notion of clutter.

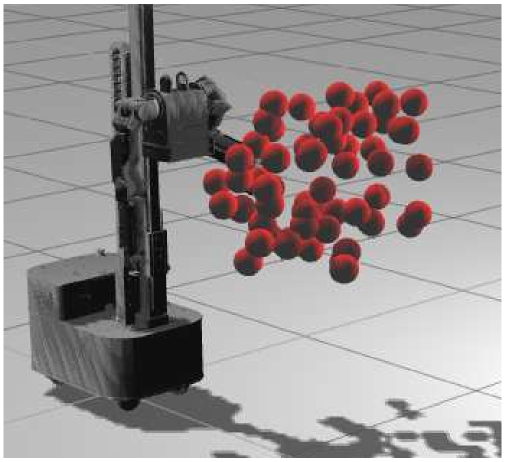


Figure 5: Training environment in GAZEBO. Training for LIC is performed in simulation prior to the real demonstration. We use 60 fixed-floating spheres with 0.05 m radius in a 0.5 m x 0.9 m x 0.6 m rectangular parallelepiped area in front of DARCI to simulate a densely cluttered environment.

During the demonstration, each module, trained in the environment, returns an initial configuration of the robot arm based upon the goal pose received (in the case of LIC-1) and also based on the initial and final (unsuccessful) pose of the first reach (in the case of LIC-2). The robot then moves to the indicated initial configuration before executing a greedy reaching behavior using the dynamic MPC controller.

2.3.3 Greedy Reaching

Once an initial configuration, suggested by LIC, has been assumed by the robot, the central interaction manager sends the goal pose to the dynamic MPC controller, which executes a greedy reaching behavior toward the goal, while maintaining low contact forces with the environment. Because the controller limits contact forces along the arm, it often moves along even rough obstacles without becoming stuck against them, enabling it to reach seemingly difficult-to-reach goals.

However, the controller can become stuck against relatively simple obstacles, such as artificial foliage, if it finds a local minimum such that greedily reducing the control error will not advance the end effector toward the goal. The controller is deemed to have failed or become stuck if it fails to reduce the distance from the goal by one tenth over a 4 second period. In these cases, the greedy nature of the controller prevents it from discovering alternative paths which might allow it to reach the goal successfully. In such cases, the complete system is able to compensate for this shortcoming by providing increasingly more-informed plans, both in the form of LIC-2 initial conditions, and through haptic mapping and geometric planning.

2.3.4 Extracting the Arm

After performing a greedy reach, the robot must extract its arm from the cluttered environment. To accomplish this, we have explored two methods. The first, more suitable for simpler environments, records the trajectory of the end effector as the greedy reach is performed, adding an additional point to the path once the end effector has traveled more than 1 *cm* from the previous point. Upon completing a greedy reach, the interaction module then uses the greedy dynamic MPC controller to bring the end effector to each pose in the recorded path, in the reverse order. The next goal is given once the end effector is within 5 *cm* of the currently-assigned goal. This method does not constrain the redundant degrees of freedom in the arm, but still preserves some of use of the clear path which was found by the dynamic MPC controller in reaching.

The second method, used in the demonstration, records the joint configurations of the robot as it performs a greedy reach, and then uses the posture-controlling dynamic MPC to return to each configuration in the reverse order. The module records a new configuration each time any joint in the arm moves more than 3 degrees from its position in the previous history entry. In order to better take advantage of the ability of the MPC controller to resolve constraints on the arm, a configuration along the return path is considered reached when the angles of the arm are all within 3 degrees of the desired configuration. This method is often less successful in simple environments, where the former method allows the MPC controller to avoid simple obstacles, but is more successful in complex configurations with complex obstacles, where it can more exactly trace the clear path that was found during reaching.

2.3.5 Haptic Mapping

During manipulation in cluttered environments, unintentional or ‘incidental’ contact with objects can be frequent. The information from these incidental contacts could be potentially used to infer properties of the environment. These inferred properties can in-turn help in intelligent manipulation planning strategies. However, rapid identification of haptic properties of objects in unknown environments during exploration or navigation is a difficult problem. In this section, we demonstrate that data-driven methods can be used to rapidly categorize objects encountered through incidental contact on a robot arm.

We use hidden Markov models (HMMs) to model the time-series contact force data from the fabric-based tactile sensor and use the models to classify the objects in the environment into the categories of ‘rigid’ and ‘soft.’ The elements which constitute an HMM are (1) N, the number of states in the model; (2) M, the number of distinct observation symbols per state; (3) A = $\{a_{ij}\}$, the state transition probability distribution; (4) B = $\{b_j\,(k)\}$, the observation symbol probability distribution; and (5) P = $\{\pi_i\}$, the initial state distribution [5–7]. It is represented as given in eq. (18), where the parameter λ describes the HMM model.

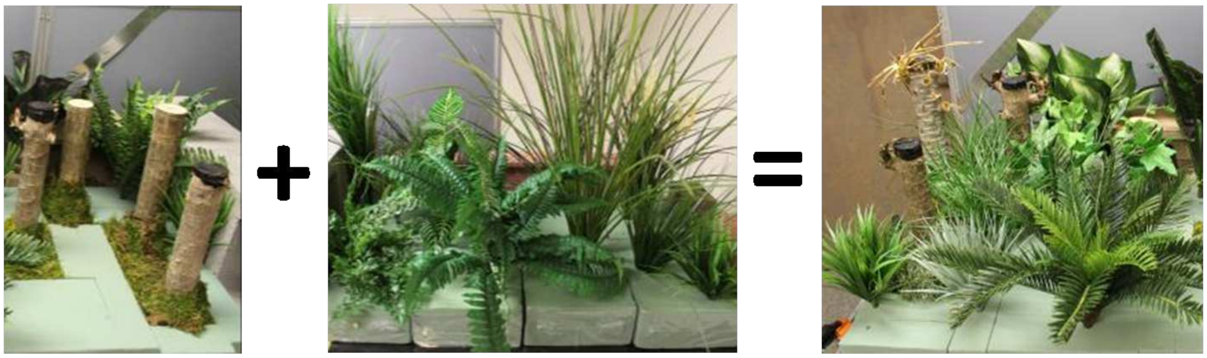


Figure 6: (Left) Trunk-only environment for training the HMM model for Trunk Category; (Middle) Leaf-only environment for training the HMM model for Leaf Category; (Right) Combined environment for testing.

$$\lambda = (A, B, \pi) \tag{18}$$

We trained the two HMM models (Rigid and Soft) using training data collected on the robot platform 'Cody' with an artificial skin on its forearm, on environments composed of small tree trunks (rigid objects) and artificial leaves (soft objects) [5] as shown in Fig. 6. We used the quasi-static MPC controller from [3] for manipulation in these cluttered environments. We had two HMM models which we trained on the leaf and trunk environments. We trained the HMMs by choosing the λ which locally maximizes $P\left(O|\lambda\right)$ iteratively using expectation-maximization (EM) techniques [6]. After we train the models λ_T for trunk and λ_F for leaf, we evaluate a new observation sequence $O = \{O_1, O_2, ...O_n\}$ according to eq. (19) which gives us the model which best matches the observation sequence. The third step in eq. (19) leads to the fourth step, if all the models are equally likely [5].

$$\begin{aligned} c^* &= \underset{c\in[T,F]}{\arg\max}\, P\left(\lambda_c|O\right) \\ &= \underset{c\in[T,F]}{\arg\max}\, \frac{P\left(O|\lambda_c\right) P\left(\lambda_c\right)}{P\left(O\right)} \\ &= \underset{c\in[T,F]}{\arg\max}\, P\left(\lambda_c|O\right) P\left(\lambda_c\right) \\ &= \underset{c\in[T,F]}{\arg\max}\, P\left(\lambda_c|O\right) \end{aligned} \tag{19}$$

During this demonstration for testing, we are using the dynamic MPC and the robot DARCI, with the flexible and stretchable fabric-based tactile sleeve, but still in an environment composed of trunks and leaves. The robot, DARCI, and the environment are shown in Fig. 2. We run the HMM models to classify, live and in real-time, the contact force data for every taxel on the tactile sleeve.

We classify the objects in the test environment into rigid and soft categories using the log-likelihood values of the two HMM models. We create a haptic map in Rviz visualization software

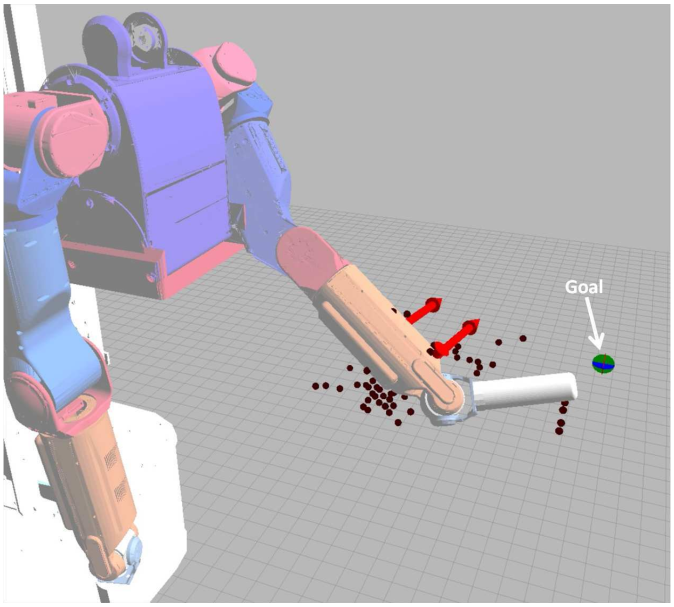


Figure 7: Haptic Map of detected rigid contacts.

by mapping all the rigid taxels at every time-instant. For visualizing the haptic map, we use point cloud/voxels for every taxel that is categorized as rigid. Each taxel with rigid contact is mapped using a dark brown sphere as shown in Fig. 7. This information is provided to the planner described in Sec. 2.3.6 so that it can avoid these areas of rigid contacts and come up with an intelligent planning strategy.

2.3.6 Planning with Contact

In this section, we describe a global search-based planner with a traverversability map constructed by the haptic classifier described in Sec. 2.3.5.

2.3.6.1 Traversability Map To use a planner in a cluttered environment, we first construct a 3D traversability map. We represent the workspace of the robot as a 3D voxel grid with 0.01 m x 0.01 m x 0.01 m voxel size in Cartesian space. Each voxel includes a traversability metric that shows the manipulation cost in that location. We define the traversability value as a scalar value between 0 to 100. The higher value a voxel has, the more difficult it is for the arm to pass through the voxel's location. In this demonstration, the robot knows what kind of object it is colliding with based on the haptic classifier of Sec. 2.3.5. This allows for updating the traversability map online during reaching. For this demonstration, we assign manipulation costs of 0, 50, and 100 into empty area, movable or soft object area, and fixed-rigid object area, respectively.

The area of map is defined as a rectangular box, 0.6 m x 0.7 m x 0.6 m in front of the robot. It is initially populated with zeros, assuming that the unknown environment is empty and that there is little cost associated with manipulating the arm in that area. The map records the contact information using Point Cloud Library's (PCL) Voxel Grid [8].

2.3.6.2 Traversability Planner The traversability planner has two main steps: goal posture selection and trajectory planning. The goal posture is randomly selected from a list of valid

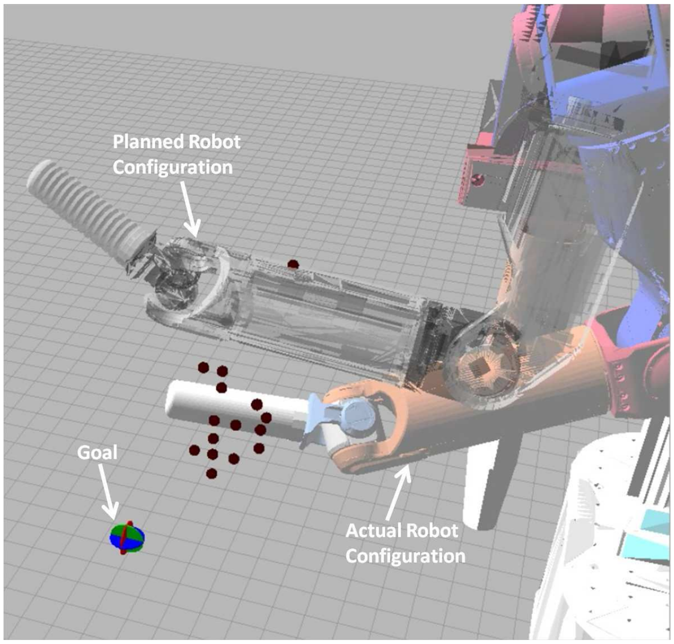


Figure 8: Planned Robot Configuration with Haptic Map.

arm postures. Valid arm postures are joint configurations such that the end-effector reaches a Cartesian goal, and the entire arm is placed in low-cost area. In detail, to create the list of the initial posture, 72 uniformly distributed orientations are sampled using the sampleSO3 function from OpenRAVE [9]. To check the cost of a path, we construct a traversability checker that computes the traversability of each vertex location from the arm collision meshes at each joint state, and rejects the state when the vertices are located inside of fixed-rigid object area of the map.

For trajectory planning, we use a global search-based planner, RRT-Connect [10] from OMPL [11]. It plans a path over the traversability map in joint space. Any arm posture in a high-cost configuration is rejected by the traversability checker. In this demonstration, we assume all other area is traversable except the rigid-fixed contact area. One example of a robot configuration returned by the planner using haptic map is shown in Fig. 8

2.3.7 Implementation

We now describe our software and hardware implementation of the system.

2.3.7.1 Tactile Sensor For tactile sensing, we use the fabric-based tactile-sensing sleeve we described in [12]. The sleeve is made of five layers of stretchable fabric. The inner and outer layers are electrically insulating, and isolate the inner layers from the robot and external world, and provide protection from abrasion. The middle of the skin contains two layers of electrically conductive fabric (a silver-plated Nylon/elastic fiber) separated by an electrically resistive fabric (a conductive-polymer coated Nylon/elastic fiber). The inner conductive layer consists of 25 individual patches of conductive fabric, each of which forms a sensing region, or ‘taxel’ for ‘tactile pixel.’ Each patch is supplied with 5V via a pull-up resister and an Arduino board. The outer conductive layer is a single sheet covering the entire sleeve, and is connected to the ground of the Arduino. As the central resistive fabric is compressed, the conductivity across the compressed portion of the fabric increases, and a drop in voltage can be detected by the Arduino in the circuit of the underlying taxel. This process is nonlinear, and depends upon both the force applied and the area over which contact occurs. However, we have found in practice that good performance can be obtained in our various systems by operating on sensor measurements directly.

2.3.7.2 Robot Platform The robot used in this work is the humanoid robot DARCI, an M1 Mobile Manipulation Platform from Meka Robotics, which includes a mobile base, a torso on a linear actuator, and two 7-Degree of Freedom arms. For the demonstration described here, the mobile base was not moved while the robot was performing the reaching task, and the torso remained fixed at its maximum height. We perform all demonstrations using the tactile sensing sleeve on the left arm of the robot, which is extended with a 3D-printed cylindrical extension of ABS plastic. The arms of the robot use a series elastic actuators at the joints, and are controlled to provide gravity compensation and an impedance controller that simulates low-stiffness visco-elastic springs at the robot's joints.

2.3.7.3 Software The software for this demonstration consists primarily of Python code, with some portions being written in C++. The system is coordinated using the Robot Operating System (ROS) [13] for communication between the various modules, as well as for communication with the low-level controllers on the robot arm. The modules described above (Sec. 2.3) are typically each contained in a single process, or 'node,' in the ROS framework. Individual modules make heavy use of various software libraries related to their specific functions, as noted above. In particular, the Model Predictive Controller uses the CVXGEN [14] library for solving a convex optimization in determining the control inputs to the low level controller at each time-step. The state of the system is observed using the ROS Rviz visualization engine to visualize the state of the robot, the location and sensor readings of contacts on the tactile sensor, the active goal location, and the current state of the haptic map. Rviz also allows goals to be identified using the 'interactive marker' interface. This interface is used extensively in development and testing of this demonstration. During the demonstration itself, a goal location is first identified by manually bringing the end effector of the disengaged robot to a desired goal location, and a Python script stores the location of the end effector based on the robot's kinematics. This script later sends this goal position to the system.

2.4 Results

The combined system was evaluated in the trunk-and-foliage environment described in Sec. 2.3.5. We identified seven goal locations distributed across the environment, attempting to identify locations of varying difficulty, with both trunks and foliage between the robot's setup location and the goal. The goal locations can be seen in Fig. 9. After each reach, the foliage was replaced to approximately its original position, so that repeated physical interaction by the robot did not significantly alter the environment. The system attempted to reach each goal location three times, for a total of 21 attempts. 16/21 (76.19%) of the reaches succeeded. Each location was reached at least once, and 6/7 locations were reached in less than 20s on at least one occasion. The fastest successful reach to goal #5 required 70 seconds.

The average time to reach each goal when successful was $39.74 \pm 46.00s(mean \pm std)$. 9/21 (42.86%) attempts were successful on the first reach using the dynamic MPC controller from a

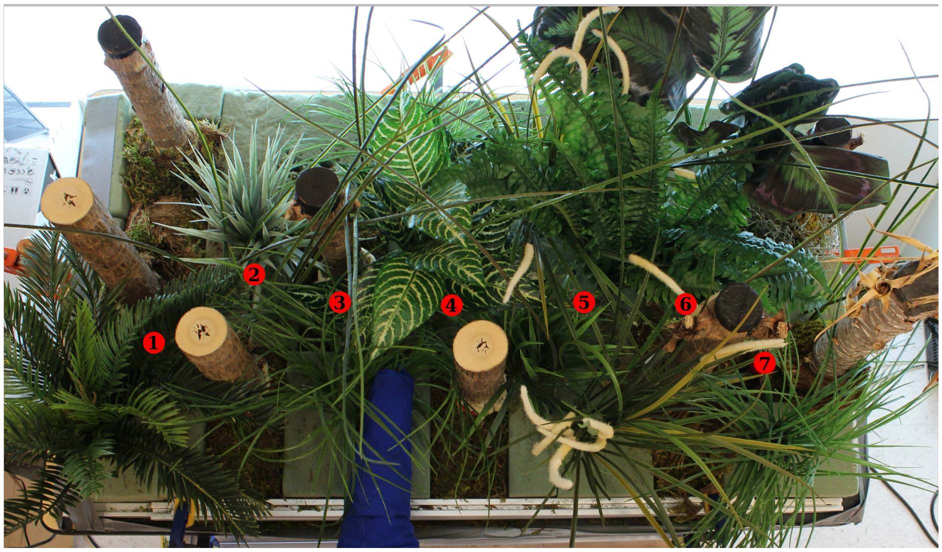


Figure 9: Trunk-and-foliage test environment. Seven goal locations identified by red dots.

Learned Initial Condition. These trials succeeded in $10.97 \pm 3.87s(mean \pm std)$. 1/21 (4.76%) attempts was successful on the second reach, starting from the second Learned Initial Condition. This trial succeeded in $21.30s$. 5/21 (23.81%) attempts were successful on the first reach using a planned path based on the haptic map. These trials succeeded in $67.59 \pm 26.77s(mean \pm std)$. 1/21 (4.76%) attempts was successful on the direct reach using the dynamic MPC controller, starting from the failure point of a third planned trajectory. This trial succeeded in $177.93s$. 1/21 (4.76%) attempts failed when attempting to pull back after the first reach failed. 2/21 (9.52%) attempts failed when attempting to reach the second Learned Initial Condition setup configuration. 2/21 (9.52%) attempts failed when a plan could not be found either after the second LIC reach, or after pulling back to the LIC-2 setup configuration.

It is interesting to note that goal #5 required the longest time to reach, and was only successful twice, and that goal #6 was only reached once successfully. These two goal locations are relatively near the robot, and largely obstructed by ostensibly movable foliage, rather than the rigid trunks, but were still the most difficult for the system to reach.

2.5 Discussion

When successful using the Learned Initial Conditions and dynamic MPC controller only, the system is able to complete a reach in $12.00 \pm 4.90s(mean \pm std)$. This is significantly faster than when geometric planning is required. In the cases where the system was successful, but planning was required, the reach was completed in $85.98 \pm 46.91(mean \pm std)$. See Fig. 10. The success of the dynamic MPC controller in combination with Learned Initial Conditions in quickly reaching a variety of goal locations in dense clutter emphasizes the capability of these relatively simple modules for coping effectively with extreme clutter. This is enabled by the use of whole-arm tactile sensing. Such tactile sensing provides data of limited size and scope that is immediately relevant to the control algorithm attempting to reach a goal and maintain low contact forces. Unlike traditional vision-based geometric planning for manipulation in clutter, where an model of the environment is produced in advance of manipulation, and significant data which is not immediately relevant to the manipulation task may be collected and processed, the MPC controller and tactile

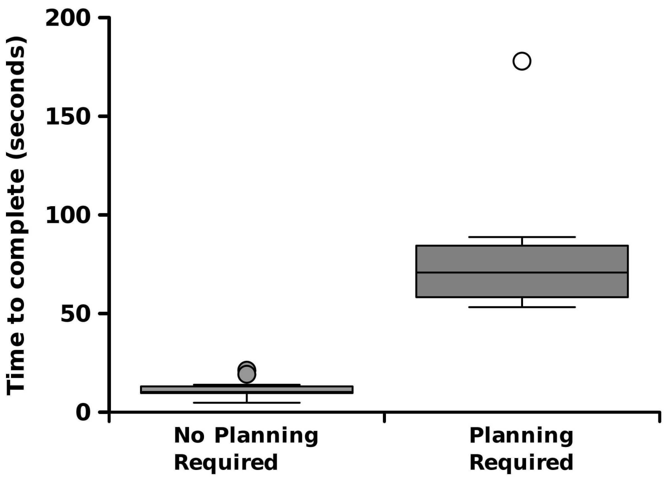


Figure 10: Completion time in cases where planning is or is not required. More difficult cases, when greedy reaching does not succeed, require significantly longer to solve. However, the planning component of the system is often able to reach the goal eventually.

sensor use significantly less data with a more direct physical relationship to the task, requiring less computation, and in turn enabling real-time feedback control. In addition, because the data is collected during reaching, rather than requiring a map to be developed in advance of manipulation, the delay of sensing and planning before acting can be removed from the traditional sense-plan-act model.

However, it is clear from the results that greedy-reaching behaviors are not always sufficient for reaching through dense clutter. For example, Figure 12 shows an the end-effector stuck against foliage which has become intertwined between two plants, and which the controller cannot push through. In such cases, the haptic mapping and geometric planning components are able to improve the success of the overall system. However, even in these cases, the maps used in planning are typically quite sparse in comparison with those produced via traditional 3D sensing such as stereo vision. This enables faster planning as fewer obstacles are present. The trade-off is that additional re-plans may be required as previously undiscovered obstacles are contacted and mapped when executing the planned trajectory. In five of the cases presented here, the planning system was able to arrive at the goal location using the first plan. In these cases, the end effector was typically near the goal location, but had become stuck against some obstacle, and only small alterations from the greedy approach were able to extract the arm and reach the goal.

The data used by the planner has the advantage of being less dense than traditional geometric maps, such as that produced by 3D vision. In addition, the haptic map is of greater relevance to the manipulation task, as it is able to represent the mechanical properties of the environment, rather than only the visual properties. In the environment presented, it is unlikely that traditional planning methods which avoid contact with the environment, using a map of the environment developed from at visual sensor at the 'head' of the robot, would be able to find any feasible path to any of the goal locations. However, our system was able to reach all of the goal locations, often quite quickly. The heavy foliage in the environment obscures the environment visually, but does relatively little to obstruct the physical progress of the end effector. The haptic mapping, in addition, is able to classify those objects encountered in the environment which do impede the progress of the end-effector, so that the planning module can appropriately plan around those

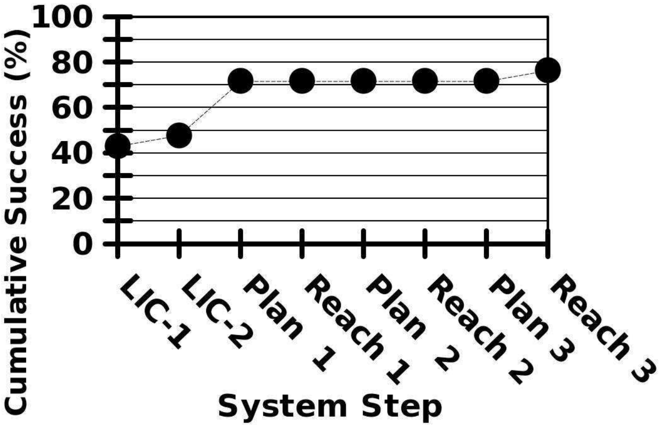


Figure 11: Cumulative success percentage as the system progresses through the process in order. The initial attempts using Learned Initial Conditions produce an almost 50% success rate, and the additional planning capabilities increase this to over 75% success after the process has completed.

obstacles, while allowing for plans that pass through light foliage and other non-rigid objects.

Finally, sequentially trying differing techniques for reaching goal locations enables the combined system to reach goals quickly when mechanically clear paths are available, while still finding less direct paths to goals which are harder to reach. Figure 11 shows the cumulative success percentage as the system progresses through the defined sequence of actions. This attempts to take advantage of the complementary capabilities of the system components, and derives inspiration from biology, where our intuition suggests that animals, and humans in particular, may attempt a task quickly and immediately to try to achieve rapid success and to gain more information about the task at hand should their initial attempts fail. If initial attempts are met with failure, a more deliberative approach is then applied, which may require more careful examination of the situation at hand, and more careful evaluation of potential courses of action. Just so, we present a similar pattern in our combined system for manipulation in extreme clutter.

However, despite these many benefits, the system is not infallible, as evidenced by the 5/21 failed trials. In one case, the system became stuck when attempting to extract its arm after making its first greedy reach. The current extraction behavior attempts to pull the end effector out along the same path which was taken to enter the environment, while maintaining low forces along the arm. In these cases, it is possible for the joints of the arm to assume different configurations than when entering the foliage and become stuck. However, this was deemed favorable to the case of extracting the arm by reversing the entire arm configuration, as this is more likely to become stuck, and does not leave the redundant degrees of freedom to maintaining low contact forces. Alternating between these two behaviors can provide some benefits of each, but is not infallible. Further work remains in the area of identifying sound strategies for extracting robotic arms from dense clutter.

Twice the system failed when it was unable to reach the setup position of the second Learned Initial Condition. These failures were somewhat unexpected, and resulted from two different causes. One failure (when attempting to reach goal #6) resulted from the setup configuration bringing the arm of the robot against the robots torso, at which point the desired configuration could not be reached. The second case was caused by the robot making contact with one of the obstacles while attempting to reach the setup position and becoming stuck. Only a simple arm-configuration path

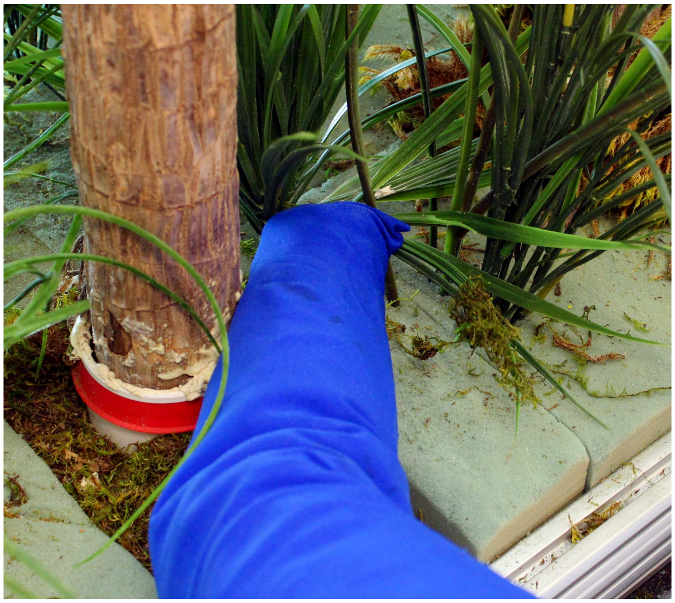


Figure 12: End-effector stuck against foliage and trunk during greedy reach toward goal #5. The leaves of two fake plants have intertwined and created a barrier which the controller will not push through due to force constraints.

is used in the setup, and failure is declared if the arm cannot continue advancing along this path at any point, making it possible to fail with only limited contact restricting the arm's motion, such as contact at the tip of the end-effector. The system currently does not advance to using planning in the case of a failed extraction or setup during the first greedy reaching phase, though this is certainly a reasonable possibility. This was not implemented in order to maintain a consistent order of operations for comparison across trials. It may also be advantageous to use the mobility of the robot's base to back away from the cluttered environment when repositioning before reaching attempts, and to move closer to the environment to enable the robot to reach further into the environment. This additional capability was not included in the current system to isolate the capabilities of the various components for controlling the motion of the arm. Another two failures were the result of the planner being unable to identify a clear path within the three minute timeout. In these cases, the maps were relatively dense, and the planning algorithm was able to identify a clear path neither from within the environment (after having performed a greedy reach), nor from outside the environment after pulling back to the LIC-2 setup position. In both cases it appeared from the live visualization that a clear path did exist, at least from outside of the environment. However, the planner, which is RRT-based, was unable to identify a clear path quickly enough. The planning algorithm has not been optimized for speed, and it is possible that other methods, such as trajectory-optimization based planning, may be able to provide valid paths more quickly and consistently.

2.6 Conclusion

We have presented an integrated robotic system capable of haptically reaching locations in cluttered environments. The system does not require detailed information about the environment in advance. When provided with a goal location, it moves to an arm configuration that it has learned from offline simulation works well in similar circumstances and then greedily attempts to reach the goal. If this fails, after extracting its arm from the environment it moves to another arm configuration that it has learned works well based on the nature of failure in the first reach. It then greedily reaches to the goal again. While the system is operating, it uses tactile recognition to detect impassable locations based on incidental contact and continually updates a map of the environment with this information. If the system does not reach the goal via these two greedy reaches, it plans and re-plans paths to the goal based on this constantly updating map, withdrawing the arm from the clutter if the planner is unable to find a path in a reasonable amount of time. In our demonstration, the robot successfully reached goals using greedy reaching and planning. Further testing and debugging of system components and integration challenges is ongoing, and careful evaluation of our final systems performance using the robot DARCI will be performed, with detailed results being provided in a subsequent report.

References

[1] P. Grice, M. Killpack, A. Jain, S. Vaish, J. Hawke, and C. Kemp, "Whole-arm tactile sensing for beneficial and acceptable contact during robotic assistance," in *ICORR, 2013 IEEE*, Seattle, WA, USA, Jun. 2013.

[2] M. D. Killpack and C. C. Kemp, "Fast reaching in clutter while regulating forces using model predictive control," in *Humanoid Robots, IEEE-RAS International Conference on*, Atlanta, GA, USA, Oct. 2013.

[3] A. Jain, M. D. Killpack, A. Edsinger, and C. C. Kemp, "Reaching in clutter with whole-arm tactile sensing," *International Journal of Robotics Research (IJRR)*, vol. 32, no. 4, pp. 458–482, April 2013.

[4] D. Park, Y. K. Kim, J. Rehg, and C. C. Kemp, "Learning to reach into the unknown: Selecting an initial condition when reaching in clutter," May 2014, submitted for publication.

[5] T. Bhattacharjee, A. Kapusta, J. M. Rehg, and C. C. Kemp, "Rapid categorization of object properties from incidental contact with a tactile sensing robot arm," in *Humanoid Robots, IEEE-RAS International Conference on*, Atlanta, GA, USA, Oct. 2013.

[6] L. R. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," in *Readings in Speech Recognition*, A. Waibel and K. F. Lee, Eds., Kaufmann, San Mateo, CA, 1990, pp. 267–296.

[7] W. Chai and B. Vercoe, "Folk music classification using hidden markov models," in *Proceedings of International Conference on Artificial Intelligence*, vol. 6, no. 6.4. Citeseer, 2001.

[8] R. B. Rusu and S. Cousins, "3D is here: Point Cloud Library (PCL)," in *IEEE International Conference on Robotics and Automation (ICRA)*, Shanghai, China, May 9-13 2011.

[9] R. Diankov and J. Kuffner, "Openrave: A planning architecture for autonomous robotics," Tech. Rep., 2008.

[10] J. Kuffner and S. LaValle, "Rrt-connect: An efficient approach to single-query path planning," in *Robotics and Automation, 2000. Proceedings. ICRA '00. IEEE International Conference on*, vol. 2, 2000, pp. 995–1001 vol.2.

[11] I. A. Şucan, M. Moll, and L. E. Kavraki, "The Open Motion Planning Library," *IEEE Robotics & Automation Magazine*, vol. 19, no. 4, pp. 72–82, December 2012, http://ompl.kavrakilab.org.

[12] A. Bhattacharjee, T. Jain, S. Vaish, M. D. Killpack, and C. C. Kemp, "Tactile sensing over articulated joints with stretchable sensors," in *IEEE World Haptics Conference (WHC), The 5th Joint EuroHaptics Conference and IEEE Haptics Symposium*, April 2013.

[13] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "Ros: an open-source robot operating system," in *ICRA workshop on open source software*, vol. 3, no. 3.2, 2009.

[14] J. Mattingley and S. Boyd, "Cvxgen: a code generator for embedded convex optimization," *Optimization and Engineering*, vol. 13, no. 1, pp. 1–27, 2012. [Online]. Available: http://dx.doi.org/10.1007/s11081-011-9176-9

Multi-contact Variable Compliance Manipulation in Extreme Clutter

DARPA Maximum Mobility and Manipulation (M3)
Quarterly Technical Report #6
Covers Jul 01, 2013 to Sep 30, 2013
Submitted: December 06, 2013

PI: Charles C. Kemp (Georgia Tech)
Co-PI: Aaron Edsinger (Meka Robotics)

1 Overview

In the third quarter of 2013 we made the following progress:

- Improvements in Learning and Prediction Schemes for Identifying the Best Initial Configurations during Manipulation in Clutter
  - Extended to reselection problems from multiple observations.
  - Success rate up to 48.1% better than conventional selection methods; random and cost-metric methods.
  - Demonstrated the scheme for a reaching-in-clutter experiment in a foliage-aperture-clutter field using a real robot, PR2.
- Performed Various Global Tasks of Reaching in Clutter Using the Dynamic Model Predictive Controller with a Real Robot, DARCI.
  - Adapted the new controller for DARCI using only the manufacturer's specified dynamic parameters.
  - Showed the running controller in Python at only about 25 Hz still allows us to have reasonable force control and success rates.
  - Showed the possibility of reaching at faster rates into cluttered environments while controlling velocities, forces, and mitigating effects of unexpected impact than quasi-static model predictive controllers.

2 Improvements in Learning and Prediction Schemes for Identifying Best Initial Configuration during Manipulation in Clutter

2.1 Learning initial condition with multiple observations

We have evaluated our statistical estimation and learning and prediction schemes to rationally decide initial position condition and reduce retrials using one observation. Now, we refine our approach and extend into a reselection problem from multiple observations.

After a trial, we have obtained observations o about the environment v, and we can adapt the initial condition to improve the probability of success. This problem can be written as

$$\begin{aligned} \underset{x_0}{\text{maximize}} \quad & P(x_\infty = g | x_0, o) \\ \text{subject to} \quad & q_{i,min} \leq IK(x_0) \leq q_{i,max},\ i \in [1, m] \\ & x_0 \in \text{open space}, \end{aligned} \tag{1}$$

where $x_0 \in \Re^n$ is the restart pose of an end effector as the initial condition, x_∞ is the final stop position, and g is the goal position. x_0 satisfies joint constraints, where q_{min} and q_{max} are minimum and maximum joint limits, and IK is the inverse kinematics of the end effector. In addition, we constrain x_0 to the open space outside the clutter. o denotes observed information from the previous interrupted trial. Here, we define o as

$$o = \{x'_0, f'_1, ..., f'_n\}, \tag{2}$$

where x'_0 denote the previous initial condition and f is a category-dependant feature vector that can include the final position x'_∞ or last moving direction $\dot{\hat{x}}'_\infty$ averaged from the directions of last 200 steps.

Similar to our previous development, we compute the marginal probability condition on the observation,

$$\begin{aligned} P(x_\infty = g | x_0, o) &= \int_v P(x_\infty = g | x_0, o, v) dv \\ &\approx \int_{V_c} P(x_\infty = g | x_0, o, v') P(v') dv'. \end{aligned} \tag{3}$$

We train the model using a number of successful- and failed-trial samples. The estimation can be extended to a reselection problem from multiple observations, if the observations are conditionally independent given a goal and start,

$$P(x_\infty = g | x_0, o_1, ..., o_n) = \prod_{i=1}^{n} P(x_\infty = g | x_0, o_i) \tag{4}$$

where n is the number of past trials. We will use LIC-1 to denote the first selection framework. We will use LIC-2 and LIC-N to denote the framework in (3) and (4), respectively.

2.2 Evaluation of the selection methods with investigation of the effectiveness of different machine learning techniques

2.2.1 Prediction techniques and evaluation strategies

We investigate the effectiveness of several different machine learning (ML) techniques for the density estimation of (3):

- SE-SVR: We use a statistical estimation (SE) method with radial basis kernel interpolation for LIC-1. Then, if it fails, we use support vector regression (SVR) for LIC-2. To reduce LIC-2's training and tuning time, we sample a fixed amount of relevant data from the training set by k-nearest neighbor (K-NN) algorithm.

- GP: We use gaussian process (GP) for both trials. To reduce LIC-2's training and tuning time, we also use the same technique as above.

- K-NN: We use k-nearest neighbor (K-NN) with weighted average algorithm for both modules.

To find the maximum probability of conditions, we use a bound constrained minimization algorithm, L-BFGS-B [1].

We test our LIC framework on several categories of clutters in the 2D and 3D testbeds. We also compared the performance of the each modules with random and cost-metric methods. The following subsection describes the evaluation strategies. To evaluate the modules, we test five strategies:

- RND-RND: This strategy randomly selects the first initial condition. When the first trial fails to reach a goal, this strategy randomly selects the second condition. We use a uniform random function to sample the conditions.

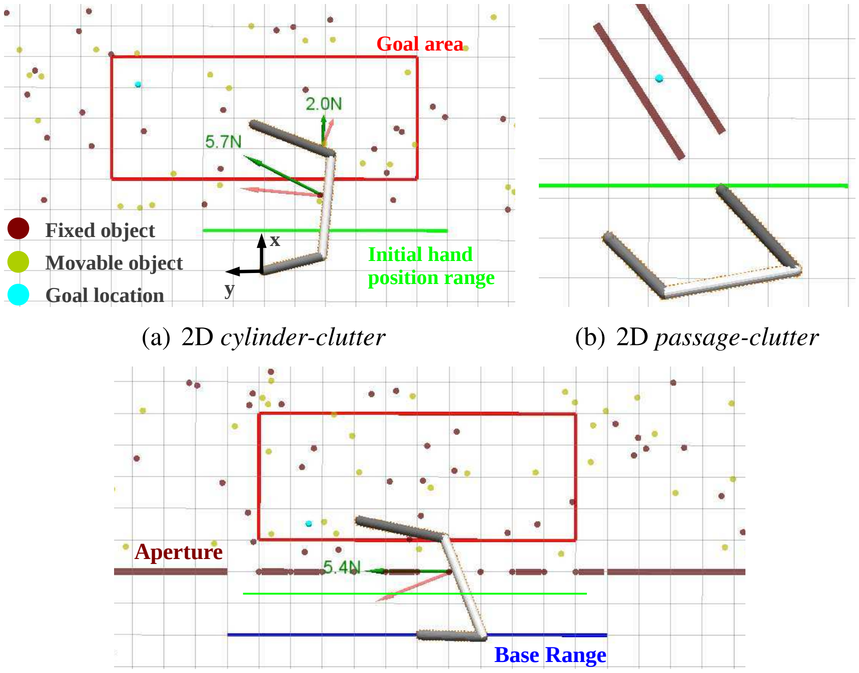
- RND-LIC2: This strategy reuses the first initial condition of RND-RND. When the first trial fails to reach a goal, this strategy selects the second condition from the LIC-2 module.

- COST-COST: This strategy selects the first initial condition from a cost-metric function that estimates an initial condition positioned closer to and orientated more directly toward a goal than other conditions. When the first trial fails to reach a goal, this strategy selects the second best condition from the cost-metric function excluding the first condition's position and neighborhood.

- COST-LIC2: This strategy reuses the first initial condition of COST-COST. When the first trail fails to reach a goal, this strategy selects the second condition from the LIC-2 module.

- LIC1-LIC2: This strategy selects the first initial condition from the LIC-1 module. When the first trial fails to reach a goal, this strategy selects the second condition from the LIC-2 module.

2.2.2 Evaluation setup description in 2D



(a) 2D *cylinder-clutter*

(b) 2D *passage-clutter*

(c) 2D *cylinder-aperture-clutter* with a mobile robot

Figure 1: An example of three different clutters with a three-link planar arm (grey). The arm starts from a condition on an initial hand range (green) to a goal (cyan) in a goal area (red). The green and red arrows represent the contact force and its normal assigned on the arm surface.

We use randomly generated three different categories of clutters in the 2D testbed. Each clutter included objects that are all planar and rigid with fixed sizes, masses, and friction coefficients.

- *Cylinder-clutter*: From a uniform distribution, we randomly placed 40 movable and 40 fixed circular objects, each with a 0.01 m radius, in a 0.65 m x 2.4 m rectangular area, as shown in Fig. 1(a). A three-link planar arm attempted to reach from 21 initial conditions to 45 grid-distributed goals of size 5 x 9 in 20 different clutters. The goals were placed on a horizontal, rectangular plane of 0.4 m long, 0.8 m wide, and 0.1 m intervals in a clutter. The initial positions were on a segment, 0.8 m long, between the robot and the clutter, equally distributed at 0.1 m intervals. The initial orientations were equally distributed at 30° intervals. For sampling, we ran 18,900 trials with 20 different clutter settings, 45 goal locations, and 21 initial conditions. For LIC-2, we used $o = \{x'_0, x'_\infty\}$.

- *Passage-clutter*: We placed a fixed narrow passage consisting of a 0.1 m gap between two walls 0.4 m long and 0.02 m width, as shown in Fig. 1(c). The center position of the passages were randomly selected on a horizontal segment 0.4 m long. We placed 12 fixed apertures, each with 0.1 m width, that are randomly blocked in evaluation test. The other objects were the same as the *cylinder-clutter*. The arm attempted to reach from 14 initial conditions to a middle point of a passage. We distributed the arm's initial positions as above. The initial orientations were equally distributed at 45° intervals. For sampling, we ran 1,680 trials of different clutter settings. For LIC-2, we used $o = \{x'_0, \dot{x}'_\infty\}$.

- *Cylinder-aperture-clutter*: First, a clutter field was generated in the same way as the *cylinder-clutter* category. Then, fixed-width openings (apertures) were randomly placed in front of the clutter field (see Fig. 1(c)). The robot was given exactly one initial position for each aperture and initial orientations equally distributed at 45deg intervals. We ran 7,200 trials with 480 different clutter settings. For LIC-2, we used $o = \{x'_0\}$.

2.2.3 Evaluation setup description in 3D

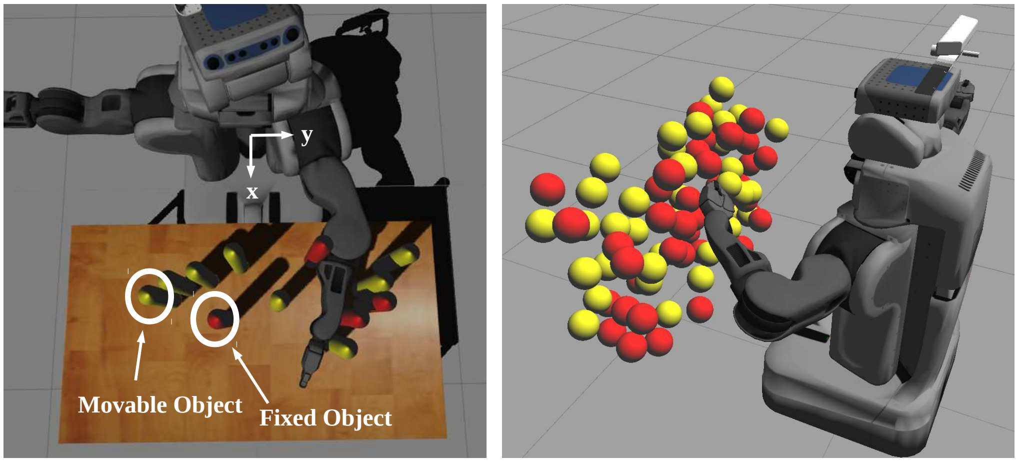


Figure 2: Visualization of a PR2 with two different 3D clutters. **Left:** a 3D *cylinder-clutter* **Right:** a 3D *sphere-clutter*. Red and yellow colors represent fixed and movable properties, respectively.

In the 3D testbed, we also used randomly generated clutters in three different categories.

- *Cylinder-clutter*: The objects were all rigid and upright cylinders with fixed sizes, masses, and friction coefficients. From a uniform distribution, we randomly placed eight movable and eight fixed objects into a 0.45 m x 1.0 m rectangular area on a desk. Each object had a 0.03 m radius, a 0.4 m length, and a 0.1 kg weight. To prevent the objects from falling over, we used sufficiently biased inertia. A PR2 robot tried to reach from 81 initial conditions to 55 grid-distributed goals of size 5 x 11 in 20 different clutters. The goals were placed on a horizontal, rectangular plane of 0.45 m long, 1.0 m wide, and 0.1 m intervals on a desk.

The initial positions were equally distributed on a vertical, rectangular plane of 0.5 m long, 0.3 m tall, and 0.1 m intervals. We used a subset of 72 equally distributed orientations that satisfied kinematic constraints and lay within the 78.5° of the PR2 torso x-axis. With these settings, we ran 89,100 trials for the sampling. For LIC-2, we used $o = \{x'_0, x'_\infty\}$.

- *Sphere-clutter*: This clutter included 40 fixed and 40 movable floating spheres in a 0.29 m x 0.4 m x 0.7 m rectangular parallelepiped area in front of the PR2. Each object had a 0.05 m radius and same properties as above. A PR2 robot tried to reach to 12 grid-distributed goals of size 4 x 3 in 40 different clutters from 20 initial conditions. The goals were placed on a vertical, rectangular plane of 0.6 m wide, 0.4 m tall, and 0.2 m intervals behind of a set of spheres. The initial positions were equally distributed on a vertical, rectangular plane of 0.8 m wide, 0.6 m tall, and 0.2 m intervals. To reduce the number of the initial orientations, we selected one from available configurations by a cost function C,

$$C = 1/(\|x_{ee} - x_{wrist}\| + \|x_{ee} - x_{elbow}\|), \tag{5}$$

  where x_{ee}, x_{wrist}, and x_{elbow} are positions of end-effector, wrist, and elbow, respectively. With these settings, we ran 9,025 trials for the sampling. For LIC-2, we used $o = \{x'_0, x'_\infty\}$.

- *Sphere-aperture-clutter*: For the real experiment, we constructed a *sphere-aperture-clutter* that consists of 40 movable spheres and 20 square apertures with 0.2 m width, as shown in Fig. 6 (Left). The spheres tended to be the foliage where the apertures were not blocked and fixed on the ground, as shown in Fig. 5. Any other settings were the same as the 3D *sphere-clutter*. The PR2 did not change its torso and base positions while training; it recorded all training data with respect to a torso frame. It is sufficient for the experiment, since any goal-robot pose pair can be translated into the torso frame with a torso offset,

$$\begin{aligned} &\{x_{goal/world}, x_{ee/world}\} \\ &\Rightarrow \{x_{goal/torso}, x_{ee/torso}, x_{torso/world}\}. \end{aligned} \tag{6}$$

  Likewise, the observation feature vector o can be translated into the torso frame. Thus, by varying the base, torso, and arm conditions, LIC can estimate the best initial condition. For LIC-2, we used $o = \{x'_0\}$.

2.2.4 Evaluation result

We compared the performance of each LIC modules to random and standard pre-computation techniques. Our results show that LIC exhibits better performance than existing methods over thousands of scenarios in various categories of cluttered environments.

Tables 1 and 2 show that LIC successfully solved the reaching-in-clutter problems more effectively than the random and cost-metric selection methods in 2D spaces. To train our framework, we use GP with K-NN for achieving stable prediction with relevant data classification. Each column shows the success rate by strategies and each row shows the success rate of consecutive trials. The fraction in parenthesis presents the number of successful trials of the number of total trials. As shown in the *cylinder-clutter* category, LIC1-LIC2 statistically increased the total success rate to 89.40% while the RND-RND and COST-COST strategies show 69.20% and 69.20%. In the *passage-clutter* category, although its shape is comparably complex than the *cylinder-clutter* category, LIC1-LIC2 increased the success rate by 28.8%. From the success rate of the first trials in both categories, we can confirm that the LIC-1 module selected reasonable initial conditions even without specific knowledge in random environments. The reason is that our training data included the kinematic factors that link to collision probabilities upon the initial conditions in a specific category of clutters. From the second trial, the LIC-2 shows largely increased success rate by 43.55% from the first random or cost-metric trials. Since LIC-2 uses observation features from the first trials as well as the kinematic factors like LIC-1, this module can retrieve another best initial conditions from similar past situations. Sometimes, the success rates of the second trials were lower than in other strategies, because the trials included some unreachable goals and the LIC-1 module had already solved many of the reachable goals.

Table 1: Success rate of two consecutive trials in 2D *cylinder-clutter* fields. We use 200 random environments with 10 different random goal locations.

	RND-RND	**RND-LIC2**	**COST-COST**	**COST-LIC2**	**LIC1-LIC2**
1st trial	Random		Cost metric		LIC1
	61.0% (1220/2000)		68.65% (1373/2000)		77.8% (1556/2000)
2nd trial	Random	LIC2	Cost metric	LIC2	LIC2
	33.71% (263/780)	52.44% (409/780)	30.30% (190/627)	41.95% (263/627)	29.96% (133/444)
Total	74.15% (1483/2000)	81.45% (1629/2000)	78.15% (1563/2000)	81.80% (1636/2000)	84.45% (1689/2000)

Table 2: Success rate of two consecutive trials in 2D *passage-clutter* fields. We use 1000 random environments with a goal in its passage.

	RND-RND	**RND-LIC2**	**COST-COST**	**COST-LIC2**	**LIC1-LIC2**
1st trial	Random		Cost metric		LIC1
	38.7% (387/1000)		34.1% (341/1000)		66.9% (669/1000)
2nd trial	Random	LIC2	Cost metric	LIC2	LIC2
	26.43% (162/613)	69.98% (429/613)	40.67% (268/659)	71.47% (471/659)	50.76% (168/331)
Total	54.9% (549/1000)	81.6% (816/1000)	60.9% (609/1000)	81.2% (812/1000)	83.7% (837/1000)

We performed multiple-reselection problems to evaluate cumulative performance in the same settings as above. Here, the random and cost-metric methods selected its next conditions from remained initial conditions excluding the past conditions' positions and neighborhood. Our LIC method selected Nth trial's condition using LIC-N when N is over than three. Fig. 3 shows the cumulated success rate as a percentage with five additional reach attempts through random, cost-metric, and LIC methods. With a single reach attempt, the success rate of LIC was a maximum of 32.8% higher than other methods. Then, it converged to a success rate, 89% and 86.3%, with only two or three attempts in both categories. On the other hand, cost-metric selections shows comparably better performance than the random method. However, both require at least five attempts to reach the success rate shown by LIC. Since the cost-metric method utilizes goal and robot configurations only, its performance can be worse than other methods if obstacles are placed in the middle of the path or the shape is not easily avoidable without a high-level path planner, like the first attempt of *passage-clutter*.

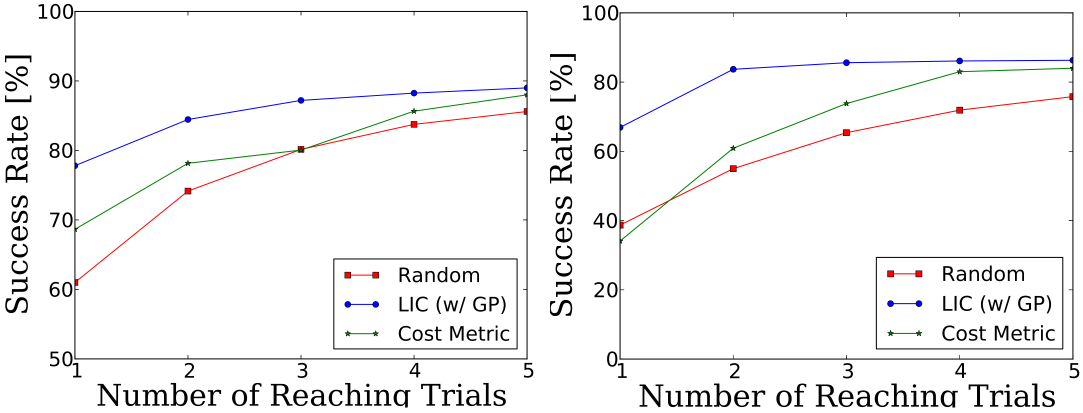


Figure 3: **Left:** Cumulative success rate of multiple retries in 2D *cylinder-clutter* field. **Right:** Cumulative success rate of multiple retries in 2D *passage-clutter* field.

Additionally, we addressed a more realistic problem with a high degree of freedom robot and random apertures, as depicted in Fig. 1(c). Table 3 shows LIC framework largely improves the success over random and cost-metric choice of initial configurations, while the mobile base and apertures dramatically increase the available configurations and reaching difficulty, respectively. This test is extended a 3D real experiment that will be mentioned later. We also tested three different ML techniques; SVR, GP, and K-NN. Fig. 4 shows all ML techniques are successful methods for the reaching-in-clutter problem, though SVR is slightly worse than others.

We extended our framework to 3D simulation and a real experiment using a PR2. Table 4 and 5 show the short 3D simulation results similar to that of 2D. The results demonstrate that LIC gives robust reaching-in-clutter performance even in 3D environment with two different categories of clutters.

Table 3: Success rate of two consecutive trials in 2D *cylinder-aperture-clutter* fields with a mobile bases. LIC framework uses a K-NN method

	RND-RND	**COST-COST**	**LIC1-LIC2**
1st trial	Random	Cost metric	LIC1
	17.2% (172/1000)	65.2% (652/1000)	67.5% (675/1000)
2nd trial	Random	Cost metric	LIC2
	15.1% (125/828)	16.4% (57/348)	31.7% (103/325)
Total	29.7% (297/1000)	70.9% (709/1000)	77.8% (778/1000)

Table 4: Success rate of two consecutive trials in 3D *cylinder-clutter* fields. We use 200 random environments with 10 different random goal locations.

	RND-RND	**RND-LIC1**	**LIC1-LIC2**
1st trial	Random		LIC1
	60.55% (1211/2000)		73.65% (1473/2000)
2nd trial	Random	LIC2	LIC2
	35.49% (280/789)	54.88% (433/789)	45.16% (238/527)
Total	74.55% (1491/2000)	82.2% (1644/2000)	85.55% (1711/2000)

Table 5: Success rate of two consecutive trials in 3D *sphere-clutter* fields. We use 200 random environments with 10 different random goal locations.

	RND-RND	**COST-COST**	**LIC1-LIC2**
1st trial	Random	Cost metric	LIC1
	54.3% (1086/2000)	60.3% (1206/2000)	67.8% (1356/2000)
2nd trial	Random	Cost metric	LIC2
	34.46% (315/914)	23.93% (190/794)	24.69% (159/644)
Total	70.05% (1401/2000)	69.8% (1396/2000)	75.75% (1515/2000)

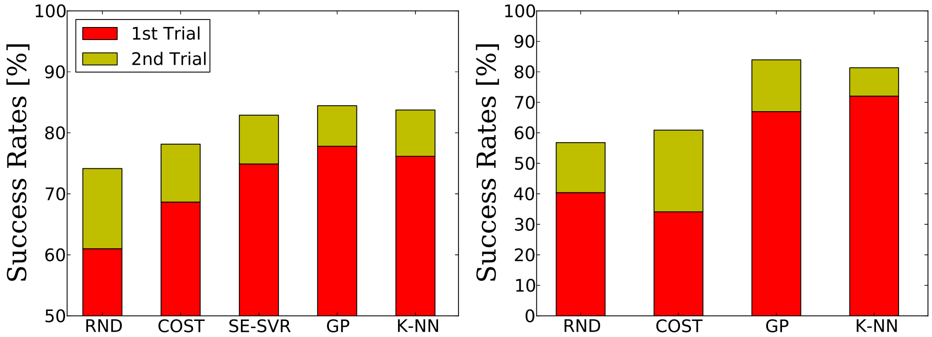


Figure 4: Success rate comparison between machine learning techniques. **Left:** Comparison in 2D *cylinder-clutter* field. **Right:** Comparison in 2D *passage-clutter* field.

2.3 Demonstration of a reaching-in-clutter in a foliage-aperture-clutter field

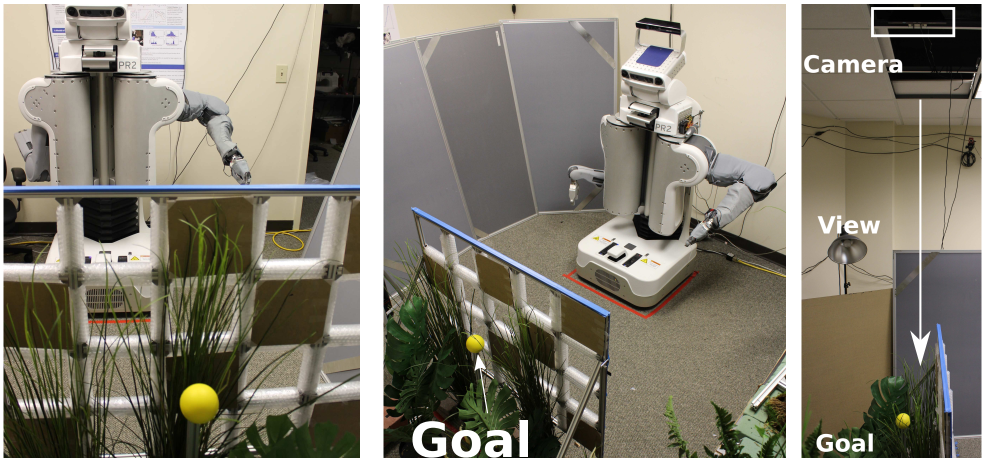


Figure 5: View of a reaching-in-clutter experiment with a PR2 in a *foliage-aperture-clutter*

For proof-of-concept demonstration, we designed a real *foliage-aperture-clutter*, as shown in Fig. 5. 20 square apertures, 0.2 m width, are randomly blocked and detected by a camera mounted on the PR2 head. A goal was also randomly placed and detected by an external camera. The objective is to reach the goal location by passing the unobservable *foliage-clutter* behind the observable apertures. We selected the best initial condition to achieve this goal. The PR2 used the MPC and fabric-based tactile sensors. Fig. 6 shows an experiment overview. From two cameras and PR2, the goal, apertures, and current robot pose were sent to LIC as the inputs g and x_0. Our LIC framework estimated the reachable probabilities of the available base, torso, and arm configurations with respect to the goal and aperture locations. After selecting the best initial condition, the PR2 transitioned to the location and performed a combined motion, forwarding and reaching, to the goal.

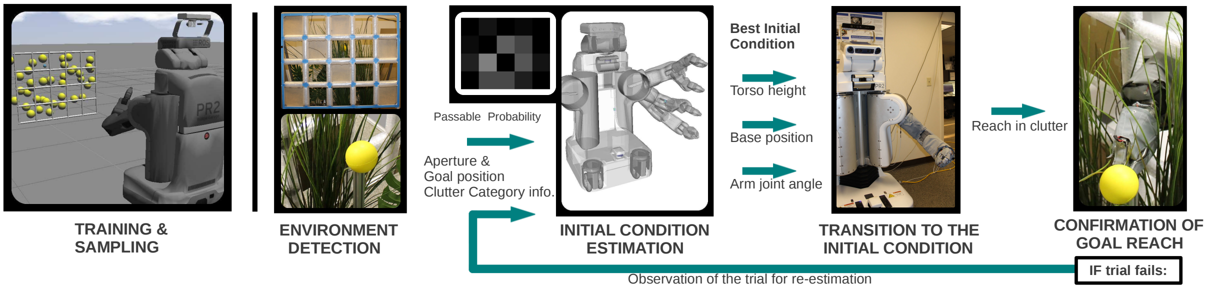

TRAINING & SAMPLING
ENVIRONMENT DETECTION
Passable Probability
Aperture & Goal position Clutter Category info.
INITIAL CONDITION ESTIMATION
Best Initial Condition
Torso height
Base position
Arm joint angle
TRANSITION TO THE INITIAL CONDITION
Reach in clutter
CONFIRMATION OF GOAL REACH
IF trial fails:
Observation of the trial for re-estimation

Figure 6: Overview of a reaching-in-clutter experiment in a *foliage-aperture-clutter* field. Training is performed in similar simulation setup prior to the real experiment. The goal and aperture locations are detected by external and head-mounted cameras, respectively. Then, LIC modules select the best initial condition from available base, torso, and arm configurations. The sample passable-probability map represents the best aperture to reach a goal as white color.

Fig. 7 shows the real experiment results that PR2 successfully estimated its best initial condition and aperture to pass into the *foliage-aperture-clutter*. The PR2 reached a goal after adjusting its base, torso, and arm configurations in order.

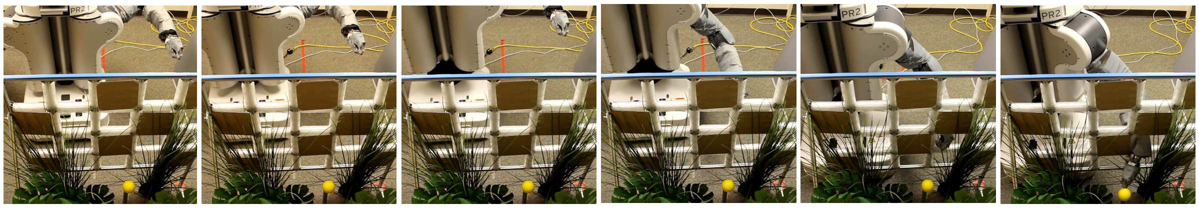
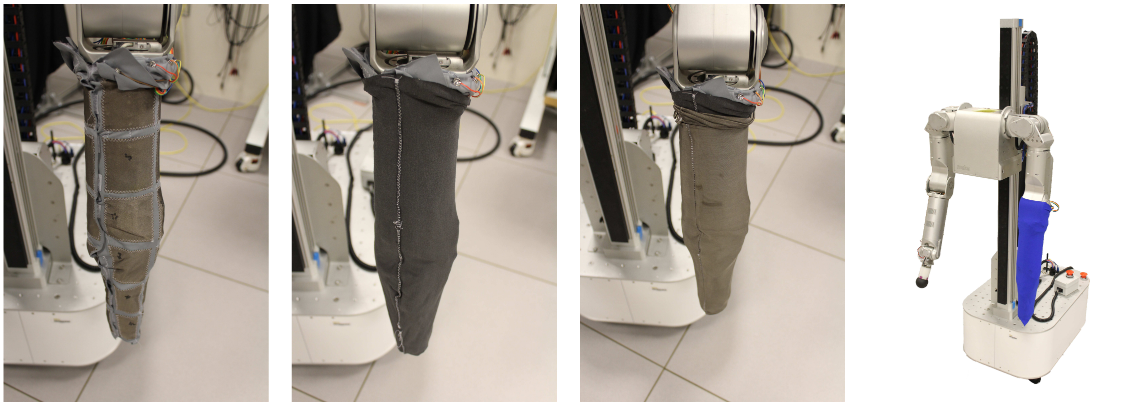


Figure 7: Snapshots of a reaching-in-clutter experiment in *foliage-aperture-clutter* field. A PR2 automatically adjusted its base, torso, and arm configurations to the best initial condition in order. Then, it successfully reached a goal location (a yellow ball).

3 Global Tasks of Reaching in Clutter Using Dynamic Model Predictive Controller with a Real Robot, DARCI

We present results using our dynamic model predictive controller on the real robot DARCI with a seven degree of freedom arm. Although the foliage-like environment we use is not identical to that presented in the previous report where we tested Cody, we believe that it has enough similarities to draw some conclusions in comparing the two. In particular, we show the ability of the dynamic controller to reach randomly generated goal locations in the clutter. We also show that according to the tactile sensor measurements, the arm is able to control its forces. Finally, we can also show that the end effector was reaching up to 5.6 times faster for the dynamic controller on DARCI than for the quasi-static controller on Cody.

The robot platform DARCI is described in [2]. Unlike Cody, we are able to use simple joint impedance control for all joints, including the wrist. The joint stiffnesses that we use for the trials in this section are 43, 43, 43, 43, 2.6, 3.4, and 3.4 Nm/rad. We also use the fabric-based tactile sensor that is an order of magnitude lower resolution (25 taxels vs 384 taxels) than the capacitive tactile sensor used in the previous section with Cody. The installation of the sensor is straightforward and is shown in Figure 8. We compressed multiple taxels with an approximately 1 cm^2 area probe and measured the ADC output along with real force measurements from a force-torque sensor. The real force on each taxel is a function of the force and area of the contact. However, for these trials, we fit an exponential curve to the ADC and real force values to calibrate the skin.



(a) Conductive layer of individual taxels wired into an Arduino board for ADC conversion.

(b) Resistive layer of fabric placed over the first layer.

(c) Last layer of conductive material that is grounded and completes the voltage divider circuit.

(d) Full sleeve on DARCI with a layer of protective blue fabric.

Figure 8: Sequencing for attaching the resistive fabric-based tactile sensor to DARCI. Another blue sleeve is also placed over the top for protection of the circuit .

Table 6: These are controller parameters identified by our simulated annealing optimization and some local search.

Parameters	Value	Description
α	239 (1/m^2)	Weight on distance to goal cost
β	255 (1/N)	Weight on contact forces above threshold
ζ	0.743 (1/N)	Weight on change in contact force above desired rate
μ	15 (1/rad^2)	Weight on change in control input
$\Delta t_{impulse}$ Offset (s)	0.825	Offset for linear function describing $\Delta t_{impulse}$
$\Delta t_{impulse}$ Slope (s/N)	-0.025	Slope for linear function describing $\Delta t_{impulse}$
$f_{rate,i}$	35 N	Desired maximum rate for change in contact force per time step
$waypoint$	0.015 m	Magnitude of waypoint that defines intermediate goal

3.1 Adaptation of the dynamic model predictive controller

Adapting the controller to run on DARCI involved a few changes. We used the same parameters identified in Table 6 except for $\Delta t_{impulse}$ and the waypoint magnitude size. These we tuned empirically to get reasonable speed and response of the arm moving in free space and in contact. Although DARCI has different kinematics and mass than our simulated three link arm, the weights gave adequate performance on the real robot. Local tuning of these parameters on a real robot is however an open research question as most research for tuning weights for MPC systems is focused on model identification or the single trade-off between performance and robustness as opposed to cost function identification and multi-objective trade-offs (see for example [3–5]). Our problem is particularly difficult as we have multiple objectives for which we do not know the appropriate weighting to achieve “good” performance which is different than a priori assuming we can define what “good” performance is such as in the following [6, 7]. This is one of the reasons that we used simulated annealing to generate Pareto fronts and explicitly see the trade-off between our cost terms.

We added an integral controller term to compensate for errors in the gravity compensation model of the robot. Gravity is not explicitly modeled in our dynamic model since we assumed that the low-level joint controllers were canceling it perfectly. This is obviously a naive assumption and in trying to reach goal locations in clutter and free space with the arm outstretched we had constant offsets of a few centimeters due to gravity acting on the robot arm. This was especially problematic for autonomous testing and termination criteria of determining when we had reached a specified goal location or extracted the arm from the clutter. To compensate for gravity, we introduced the following into the previous cost function:

$$\alpha \left\|\Delta \boldsymbol{x}_{des} - \boldsymbol{J}_{ee}(\boldsymbol{q}[t_0 + H_u + H_y] - \boldsymbol{q}[t_0]) - \boldsymbol{d}_{grav}\right\|^2 \tag{7}$$

where

$$\boldsymbol{d}_{grav} = k_i \sum_{t=0}^{t_0} (\boldsymbol{x}_{des} - \boldsymbol{x}_{ee}[t]) \tag{8}$$

We introduced some straight forward anti-windup measures and saturation limits and made k_i very small. Furthermore, this term only becomes active when the end effector is within 8 cm of the desired goal location so as to avoid serious overshoot and high forces when we are stuck far from the goal. This works in our applications since we assume that the goal location is known a priori from either a high level planner or a human operator. A more general approach that adaptively learns disturbance models is an important avenue for future work as MPC becomes more common on real robot systems.

For this controller, we also had the control horizon $H_u = 2$ and prediction horizon $H_y = 3$ which gives three time steps overall for control and then predicts the output for another four steps. These choices were mostly a function of computationally complexity in formulating the controllers and size limitations for problems generated on CVXGEN [8]. The solver generated using this configuration was able to solve reliably between 4 and 10 ms. Which means we should have been able to run between 50 and 100 Hz. However, communication and operating system issues using ROS and Python required that we run instead at 25 Hz with 40 ms time steps in order to have fairly regular intervals. Amazingly, although our force control should be worse given that our reaction time to contacts will be slower, the stability and free space motion of the arm seemed unaffected by this slow rate. It is important to remember that as long as the dynamic model does not go unstable because of the size of the time step, our discretization incorporates the time step into the model implicitly. Finally, the constraint that describes our impulse-momentum model was moved into the cost function similar to our cost on forces over the threshold. This change was due to noise on our joint velocity signal which caused the optimization to go infeasible when any joint was operating near its constrained joint velocity value. Having better filtering and dynamic models would likely allow us to move this cost back to a constraint.

3.2 Results in terms of success rates, force control and speed

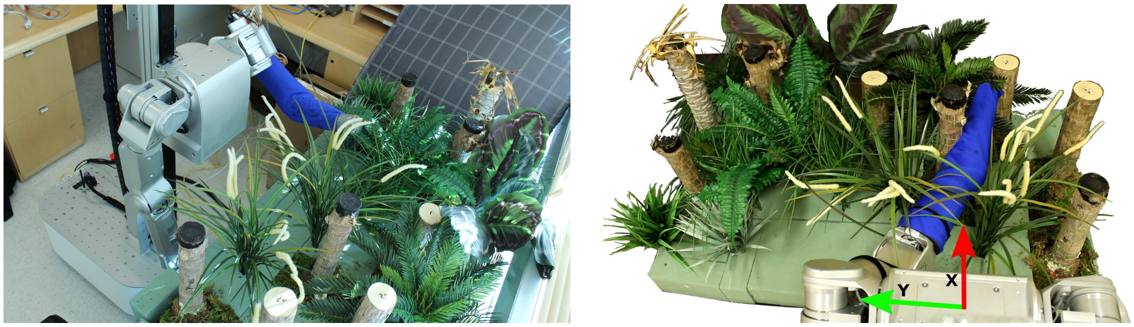


Figure 9: DARCI reaching into foliage-like environment. Coordinate frame for DARCI is also shown.

The environment that we used for testing reaching in clutter was developed by Tapomaykh

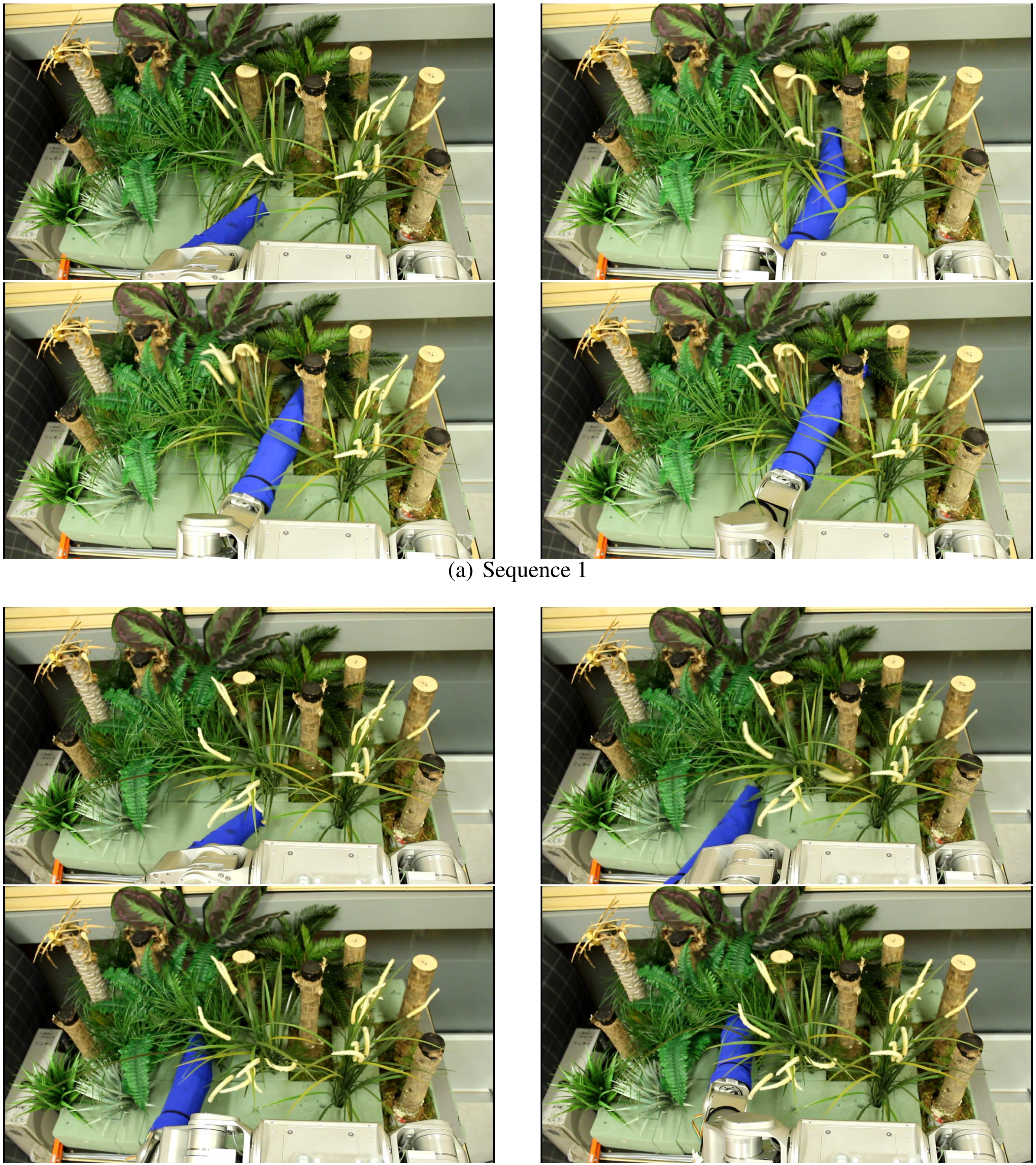
Bhattacharjee (see [9]). It is very similar to the environment presented in the previous report that we used with Cody except there are more but smaller rigid objects. The environment consists of plastic leaf-like vegetation and solid wooden trunks as can be seen in Figure 9. The leaf-like vegetation can still be difficult if not impossible to push through in certain sections of the workspace given a certain force threshold. This is due to the spacing of the plants and the fact that the base of them is more solid than the top which is different than the environment we used for Cody. Figure 9 also shows the coordinate frame used which is centered between the two robot arms and oriented as shown in the figure.

To generate data for reaching in the simulated foliage, we first estimated the workspace of the arm in the foliage. We extended the arm as far as possible to the extent of the foliage while moving through the workspace from left to right and recording the end effector positions. This essentially traced out a semi-circle that covered most of the foliage. We then uniformly randomly sampled from this approximate semi-circle in the x-y plane while randomly sampling between 5-30 cm above the ground for the foliage in the z-direction. While testing we reached into the clutter over 150 times, but only recorded data for 105 reaches. Figure 10 shows DARCI reaching into the foliage to two different locations using the dynamic MPC. The arm attempted to reach the goal for only twenty seconds before classifying the attempt as success or failure. After each reach, we again used the dynamic model predictive controller to reach to a goal 25 cm directly above its current end effector position in order to extract the arm from the clutter. This then allowed us to run the same set of commanded joint trajectories for all trials to get the arm back to its initial position and run another trial, making the trials almost completely autonomous.

The success rate for reaching the goal was 85% (89 out of 105 trials). We set the termination criterion for the minimum distance to the goal before we modified the gravity disturbance term described above. Before we introduced Eqn. 7 to the controller, we had on average 3-4 cm of error due to configuration dependent gravity disturbances when we had come to steady state since the robot was often fully outstretched. For this reason, we set 4 cm as the stopping criterion for reaching the goal at which point the controller moved on to the next trial. This is only a limitation of the success rates that we report for these trials. However, many of the trials “reached” the goal within 4 cm between 2-5 seconds, well before the maximum time allowed which was 20 seconds. We therefore expect that even with a much smaller required minimum distance to reach the goal, the success rates would have been comparable to those that we report. Using our controller to extract the arm from clutter, before resetting the arm for the next trial, was successful 104 out of 105 trials. This is a 99% success rate for extracting the arm from clutter. Figure 11 shows the distribution of the goals and the end-effector starting location.

One important aspect of these tests to remember is that the arm has very low resolution skin. There are only 4 taxels across the circumference of the forearm and wrist at any point along with a single taxel on the tip of the end effector. This also affects the performance of the controller in terms of being able to maneuver around specific contact locations.

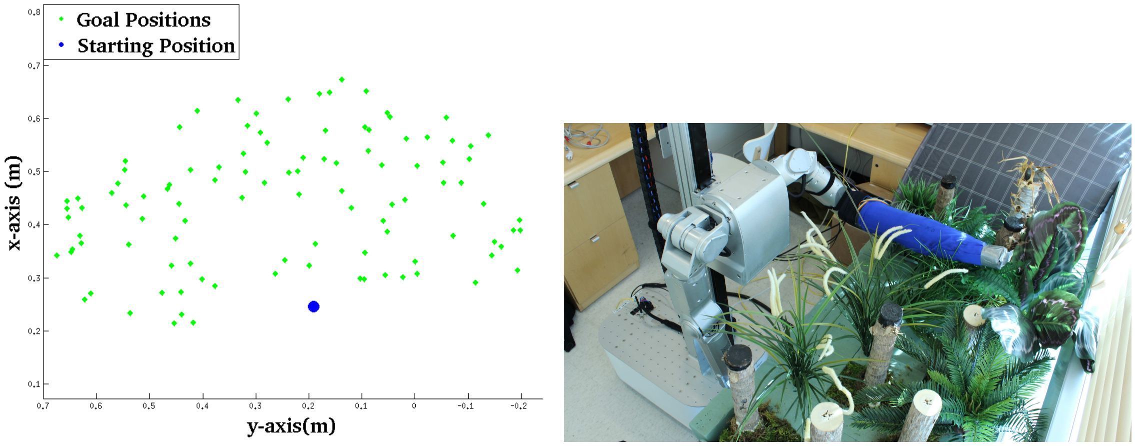
The average time to complete a trial for all of the successful trials was 4.4 seconds (noting that the end effector traveled an average distance of 0.32 cm across these same trials). The average velocity at the end effector for all trials was 12.5 cm/s, and a histogram of the velocities at the end effector estimated at 100 Hz for all trials is in Figure 12. The large number of low velocities in



(a) Sequence 1

(b) Sequence 2

Figure 10: Two sequences of images out of 105 trials where DARCI reached into simulated foliage while controlling estimated contact forces using the dynamic MPC.



(a) Goal Positions

Figure 11: **Left:** Distribution of goals (green) for the 105 reaching trials as well as the starting end effector position for all trials (blue). **Right:** Image showing DARCI with its arm partially extended. The farthest goals (shown in green) in the figure on the left are when the arm is almost fully extended.

the first bin are due in part to the failed trials where the end effector moves very little for around 15 seconds in each failed trial.

Finally, in terms of force control, the average force for all measured contact forces above a 0.2 N noise threshold was 1.6 N, while the average force for all forces above the threshold ($f_{thresh} = 5$) was 8.1 N. The average maximum contact force for all trials was 0.9 N (since many trials had very low maximum forces) and for contact forces above the threshold was 8.8 N. A histogram for all contact forces is presented in Figure 13. There is a sharp drop in the number of forces above the threshold in this histogram. However, there are still a number of forces up to the absolute maximum force sensed which was 13.9 N. The question of what causes these high forces and when they occur is addressed in the next Chapter using force-torque sensor data and more local tests. In general, we would expect that our max forces when moving faster would be slightly higher. In this case we are moving on average up to 5.6 times faster for successful trials than velocities reported for the quasi-static MPC on Cody. The trend and effect of moving faster with this given controller is also addressed in the next Chapter.

To summarize our results, Table 7 contains the values of metrics that are relevant to our task of reaching in clutter with the dynamic model predictive controller. We have shown that our dynamic model predictive controller has at least comparable success in reaching the goal as the quasi-static MPC results summarized above. The environments are not exactly the same and so direct comparison is not possible. However, it is interesting to note that although the force control is slightly worse, the speed of our dynamic controller, even in real environments appears to be faster than the quasi-static controller. For the quasi-static results on Cody reported above, the average speed for a single successful trial was reported as 2.95 cm/s. For our dynamic controller on DARCI, we had an average end effector velocity of 16.5 cm/s across all successful

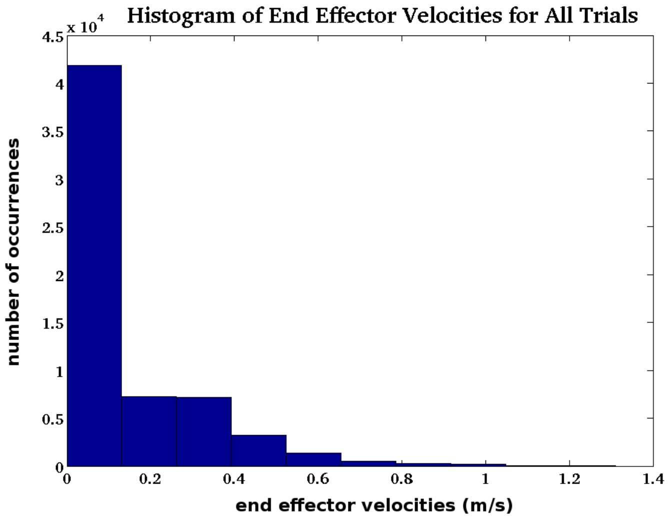


Figure 12: Histogram of end effector velocities across all trials including failed trials.

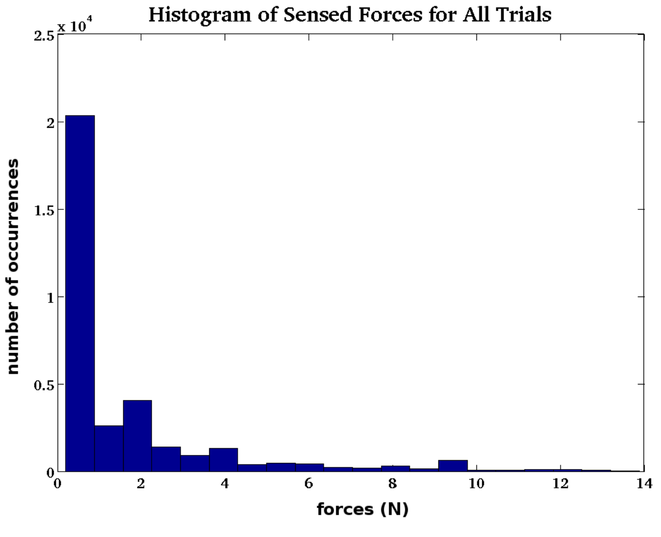


Figure 13: Histogram of measured contact forces across all trials using calibrated tactile sensing skin.

Table 7: Results for the dynamic model predictive controller reaching in foliage.

	Dynamic MPC
Success rate	85% (89/105)
Exceeded safety threshold (15N)	0 times
Avg. max. of all contact forces	0.9 N
Avg. max. of contact forces over f_{thresh}	8.8 N
Avg. of all contact forces	1.6 N
Avg. of contact forces over f_{thresh}	8.1 N
Avg. time to complete all trials	5.9 s
Avg. time to complete successful trials	4.4 s

trials. Again, the platforms are also different which makes a conclusive comparison difficult, but it does show that there is promise for increased speed while still controlling forces using our dynamic model predictive controller. Another important and final result to reiterate is that we now have a well-defined way of trading off maximum forces and end-effector speed. It is clear from the results in the previous report, that we were able to maintain the forces to be almost exclusively at or below the force threshold if the value for the joint velocities is stringent enough despite serious model error in the mass and control loop rate.

3.3 Performance of Dynamic Model Predictive Controller

In this section we present results with the real robot DARCI that emphasize and confirm some of the trends that we saw in the simulation results. We start with a set of six canonical trials that we formulated from the contact histograms. The second set of tests we ran explores the results from our MATLAB simulation in the previous report. In all cases we used the resistive fabric-based skin to control DARCI with the dynamic MPC only. We used the physical joint limits as constraints in the MPC for all joints except the wrist. Because the fabric-based sensor can have false readings of contact when the wrist joints deflect too much, we limited these to only deflect by a maximum of 30° in any direction.

We also used between one to three force-torque sensors to record the ground truth forces that the robot exerted on the environment. We recorded the data from the force-torque sensors at 100 Hz and used a solid state disk to write the data and attempt to mitigate skipping time steps in our force measurements. Figure 14 shows a typical setup for the robot and the force-torque sensors for this trial. Unlike previous trials with our robot Cody (see [10, 11]) we removed the bubble wrap from the aluminum 80/20 rods attached to the force-torque sensor. This meant that the contact was less compliant in general. There is still a small, hard foam covering on the aluminum rod. However, it appeared that the compliance at the fixture location for the force torque sensor using laser cut acrylic tended to be more significant than the foam (see Figures 15 through 25 below for examples of the rods deflecting at the base as the arm moves). Throughout the results

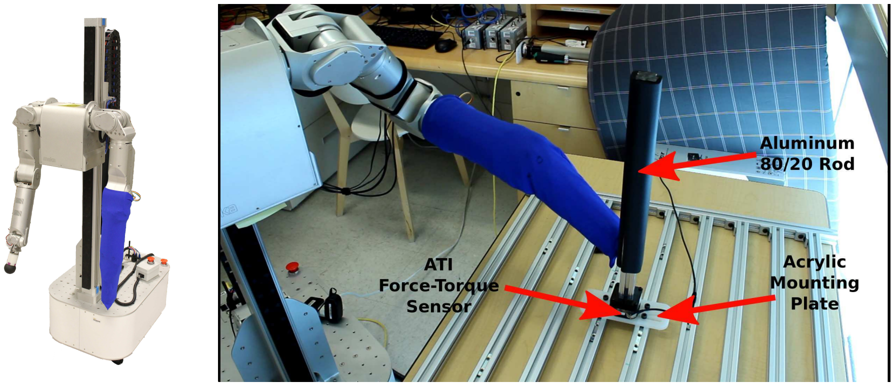


Figure 14: **Left:** Front view of DARCI with resistive tactile sleeve. **Right:** Force-torque setup that we used for gathering ground truth contact force data.

below, as we refer to “left” or “right”, this in reference to the robot’s frame of reference as seen in the images representing the canonical trials.

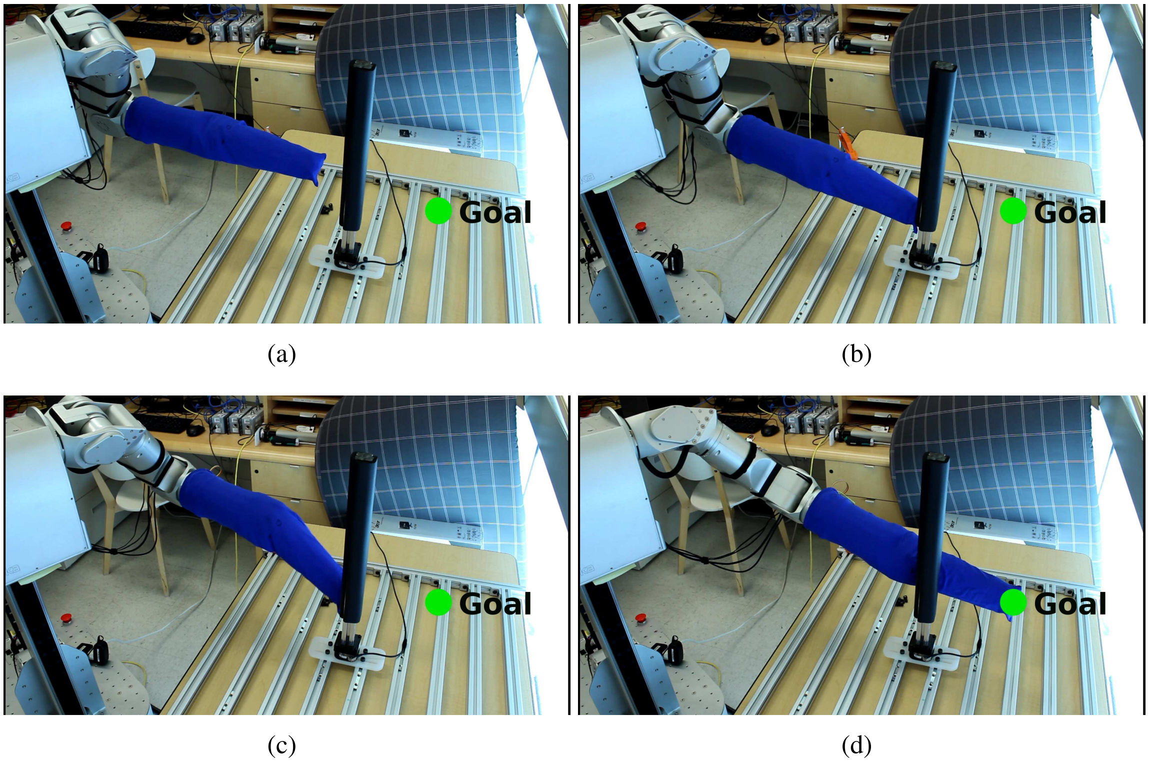
3.3.1 Control of High Forces While in Contact

Using the multi-contact distributions and general distributions for all contact forces along the simulated planar three link arm, we developed a set of canonical trials. Specifically, we examined the histograms of contact configurations on the three link simulated arm and identified contact configurations that occurred frequently or resulted in high force. We then manually confirmed that the canonical trials we had identified were representative of the histograms by randomly sampling and visualizing the actual contact configurations from the trials represented in the histograms. These tests are thus subjectively defined, but based on objective data for thousands of tests in simulation. For each set of trials in this section we first present a Figure showing the typical arm trajectories that we saw for that trial. We then present the force and velocity data if relevant for each trial. For the majority of the trials we used $f_{threshold} = 5N$ and $\Delta t_{impulse} = 0.4$. We also ran 20 trials for each canonical task unless otherwise noted. The termination criterion for each trial was either reaching the goal or a timeout of 15 seconds. Finally, it was common for two taxels to make contact with the force-torque sensor simultaneously (especially at the wrist joint). When this happens, the force torque value may be up to double the threshold and our controller would still be successfully controlling with regards to the skin sensor.

For the first trial, we had the arm move straight forward from its starting Cartesian position and make contact at the tip of the arm with the force-torque sensor, see Figure 15. This is a simple test but one that we expect to be a common occurrence given the amount of contact that we saw on the end effector in simulation.

For this trial, we also varied the value for $\Delta t_{impulse}$ between 2 and 4 while keeping f_{thresh} =5 N. In this case for each setting we only ran 10 trials. The resulting force histograms from these trials and settings are in Figure 16.

From Figure 16 we can see that as the impulse parameter increased so did the variance of the



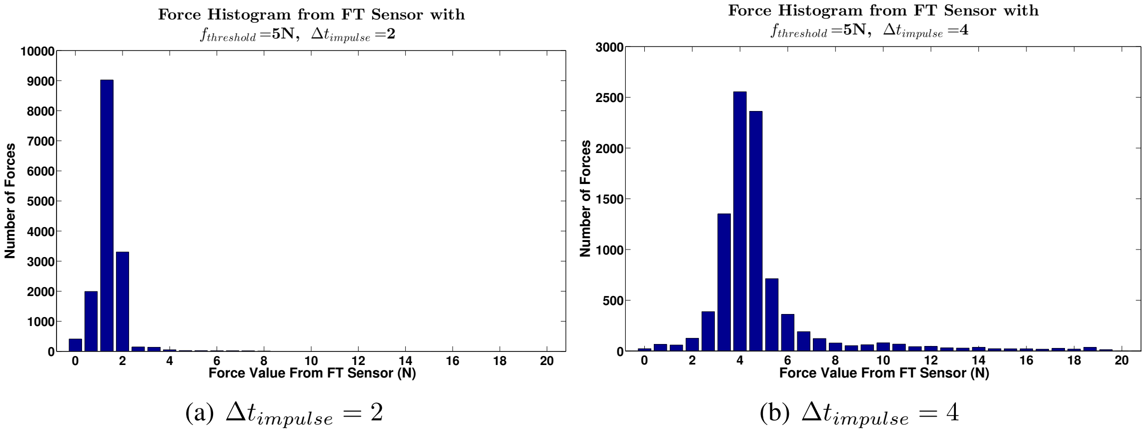
(a)

(b)

(c)

(d)

Figure 15: Canonical Test 1: DARCI reaching to a goal directly behind the force torque sensor.



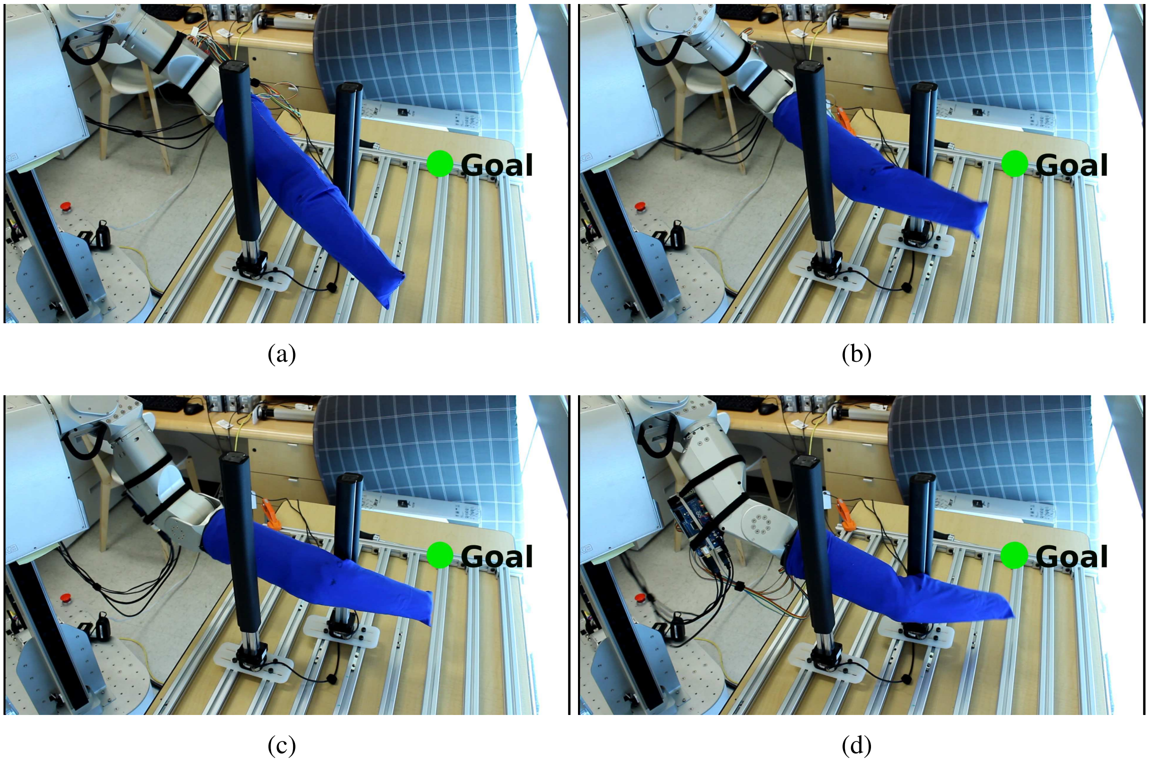
(a) $\Delta t_{impulse} = 2$

(b) $\Delta t_{impulse} = 4$

Figure 16: Canonical Test 1: Force Histogram results for DARCI as we vary the $\Delta t_{impulse}$ parameter.

distribution of contact forces as the distribution also shifted it to the right. To quantify the shift we report that the maximum contact forces as we varied $\Delta t_{impulse}$ from 2 seconds to 4 seconds, were 8.3, 19.6 N. While the 99^{th} percentile forces for $\Delta t_{impulse}$ equal to 2 and 4 seconds were 3.9 N and 17.2 N. This shows that the impulse time parameter has the effect of increasing maximum forces experienced by increasing the maximum allowed joint velocity as expected. We perform similar tests and vary $\Delta t_{impulse}$ over a wider range in Section 3.3.2. In general, we can see that the majority of the forces in both cases were regulated to be around the controller force threshold of 5 N.

In the next trial, DARCI's left arm starts between two posts positioned around the middle of the forearm before reaching to the left as seen in Figure 17.



(a)

(b)

(c)

(d)

Figure 17: Canonical Test 2: DARCI reaching to the left through two posts.

The force results for these 20 reaches are in Figure 18

In the force results for both this trial and canonical test 4 we have slightly higher forces than expected. There are also two observable modes in the force distribution in Figure 18. Upon reviewing video of the trials we can clearly see that the arm makes contact with the force-torque sensor that is closer to the robot torso and in a place on the arm where we have no taxels. This explains to some extent the high contact forces that we measured (between 20 and 40 N).

In canonical test 3 the robot arm starts in contact at the tip on the left side and in the middle of the forearm on the right hand side. In the second image in the sequence of Figure 19, the arm first pushes against both rods as it tries to move to the goal. In subsequent motions the arm uses out of plane motion to around the post and still get to the goal. This is one of the trials that we expected to have large forces due to what we saw in the simulation. However, the 3D motion and extra degrees of freedom made it straight forward for our controller to reach the goal and control the forces at the same time.

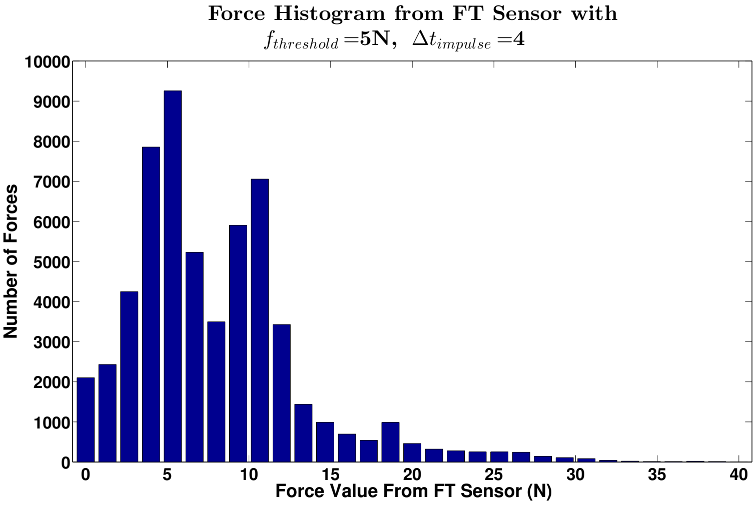
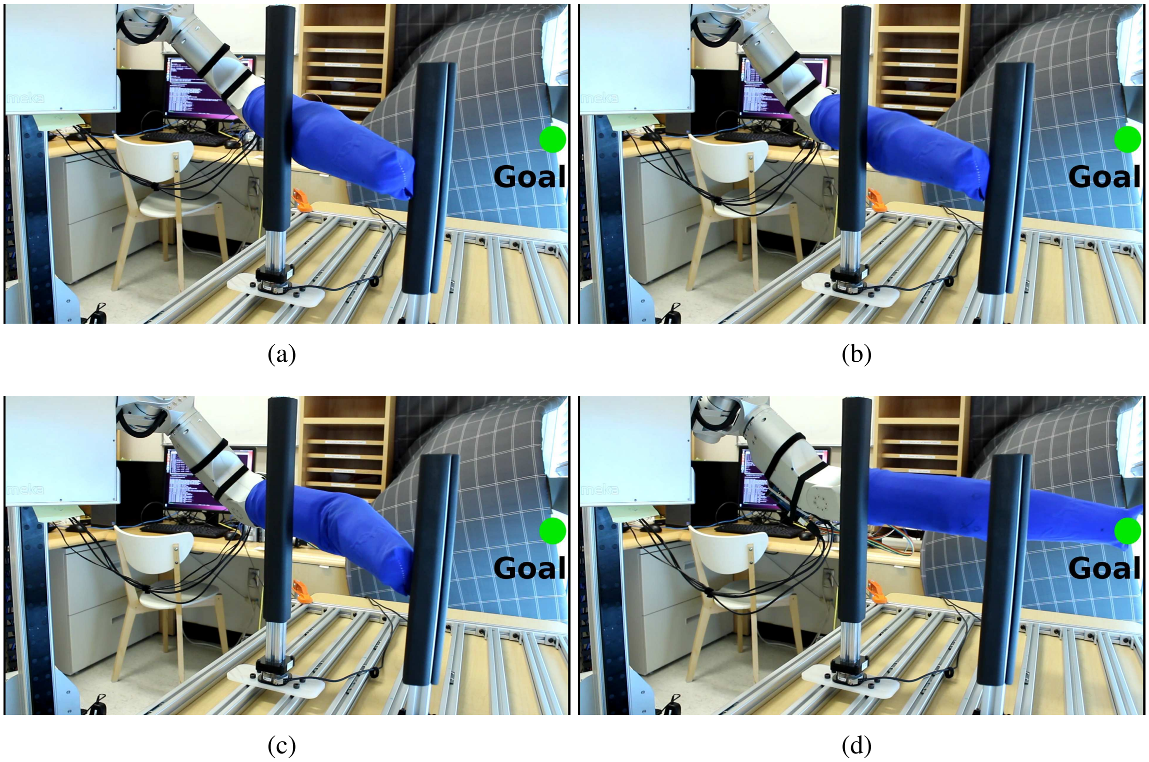


Figure 18: Canonical Test 2: Results for reaching through two posts to the left.



(a) (b) (c) (d)

Figure 19: Canonical Test 3: DARCI reaching to the left while starting in contact at the end effector.

The results for this trial are in Figure 20. These results have forces that agree more with our expectation of force control where almost all of the forces are below 10 N and the majority are even below 5 N.

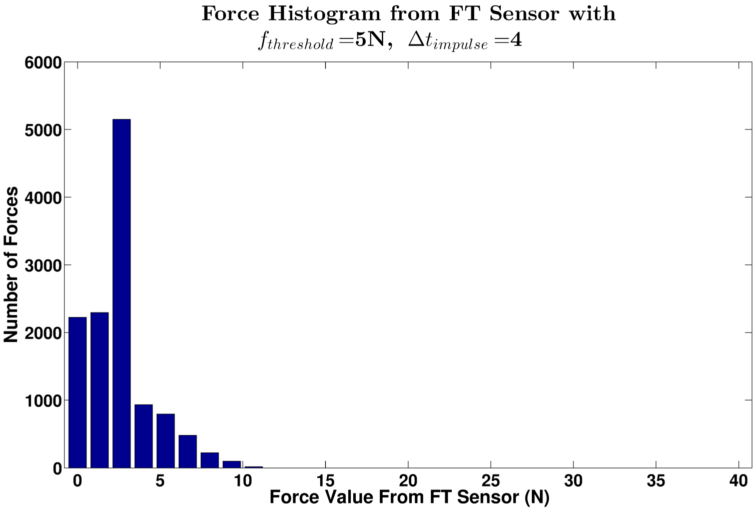


Figure 20: Canonical Test 3: Force results for reaching to the left.

Canonical test 4 is similar to test 2 except that the arm is now reaching to the right and the arm starts in contact with both posts. Additionally, the space between the two posts is much smaller in this trial. Figure 21 shows the start and progression of the trial. In both test 2 and 4, the arm never reached the goal. Reaching the goal was not the purpose of this set of tests and motion to the goal would have required non-greedy planning because of the starting configurations.

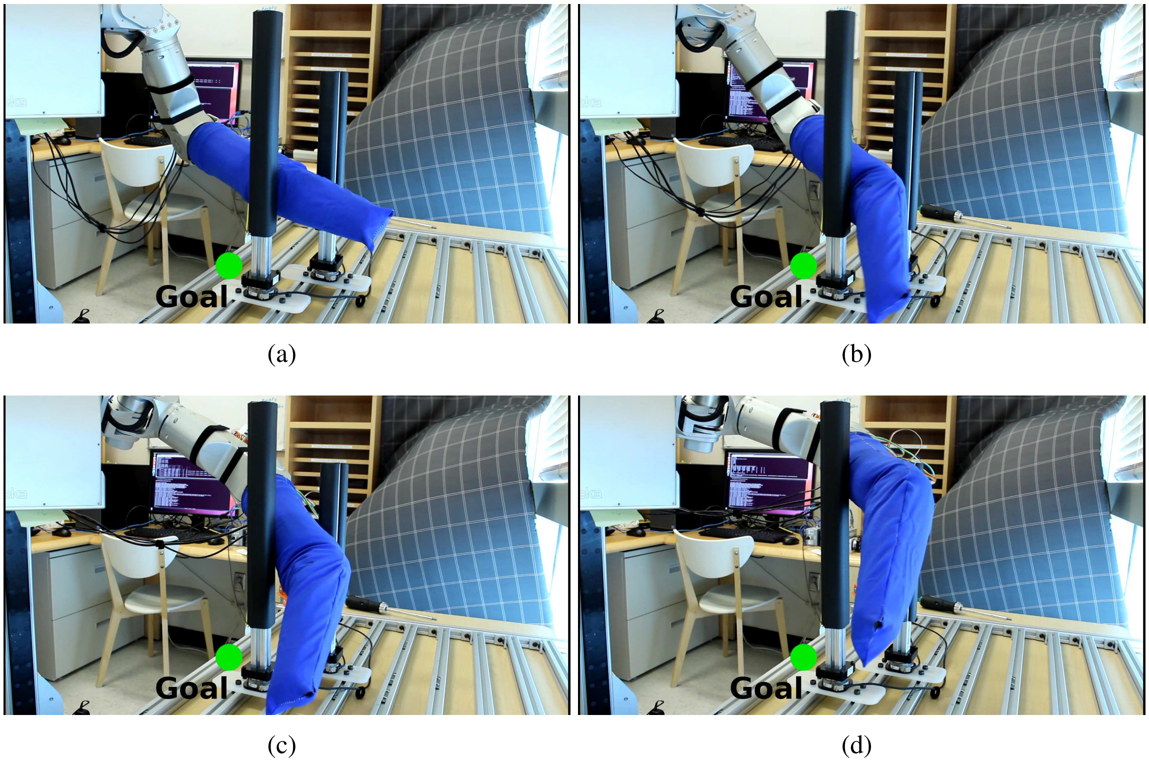
As previously mentioned, the force results for test 4 (as seen in Figure 22) were higher probably due to contact with the force-torque sensor closest to the torso on a place on the arm with no tactile sensing. However, it should be noted that this configuration was also very difficult as the posts were placed closer together and we expected to have high forces due to jamming in this situation. In either case, most forces were still limited to be below 15 N.

For canonical test 5, the arm started in contact near the wrist and was given a goal such that it kept contact with the first force-torque sensor while trying to wedge between the next two force-torque sensors to reach the goal shown in Figure 23. The goal was not attainable due to the small gap between the two distal force-torque sensors. It is possible that with the full range of motion of the wrist joints (plus or minus 60° instead of 30°) it could have at least gotten closer. However, the purpose of these tests was not to reach the goal so much as to test force control.

The contact forces for this trial are in Figure 24. The majority of the forces are again around 5 N and below 10 N with a maximum force around 25 N likely due to the wedging at the end effector.

In canonical test 6, the arm starts almost in contact with a force-torque sensor near the middle of its forearm and while reaching for a goal forward and to the right of the start position makes contact with a second force-torque sensor and then pivots about that sensor as seen in Figure 25. This was to simulate any two contacts that could occur on the same side of the arm.

The forces for contact in trial 6 are very comparable to the forces for trials 1 and 3 with the majority of contact forces being below 5 N and the max force being around 10 N.



(a) (b)

(c) (d)

Figure 21: Canonical Test 4: DARCI in contact with two posts in the middle of its forearm and reaching to the right.

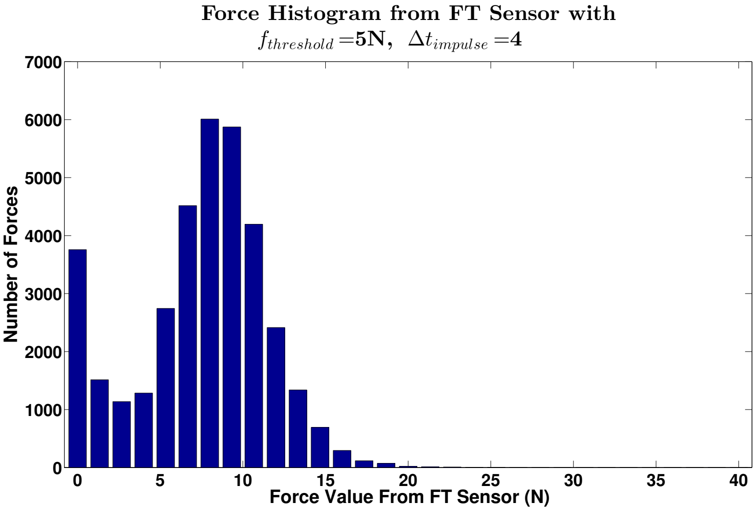
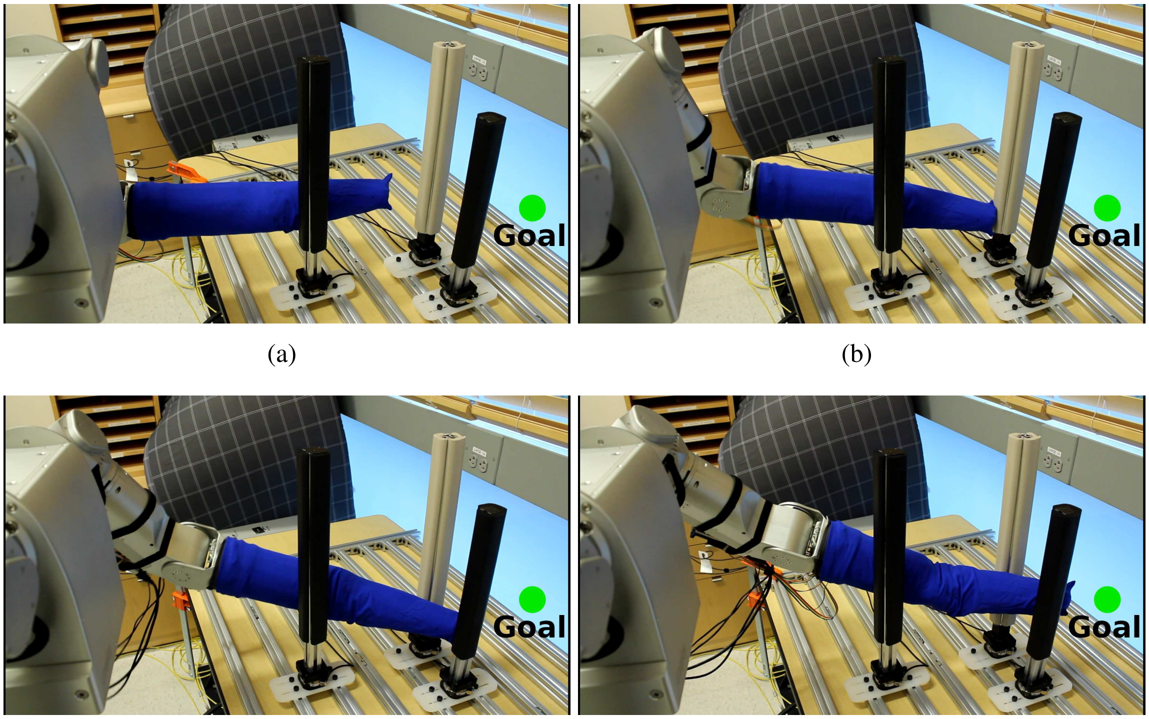


Figure 22: Canonical Test 4: Results for reaching to the right while starting in contact at the middle of the forearm



(a)

(b)

(c)

(d)

Figure 23: Canonical Test 5: DARCI reaching to a goal between two posts with a gap between them smaller than the diameter of the arm given the angle from which the arm is constrained to approach.

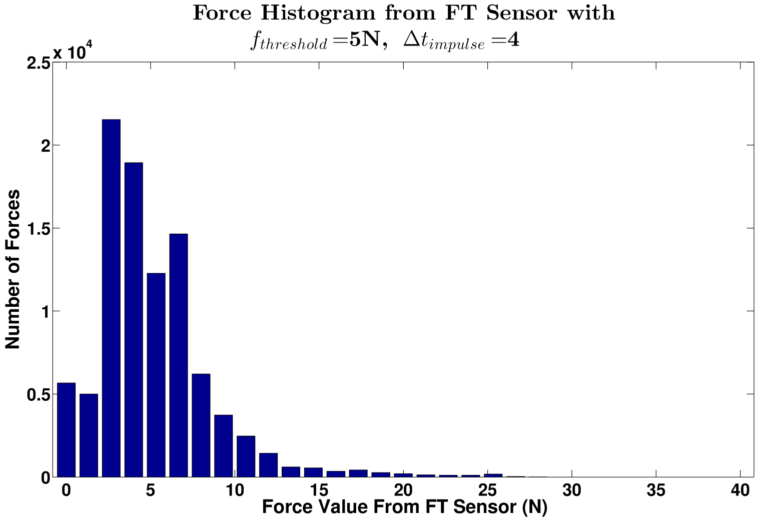
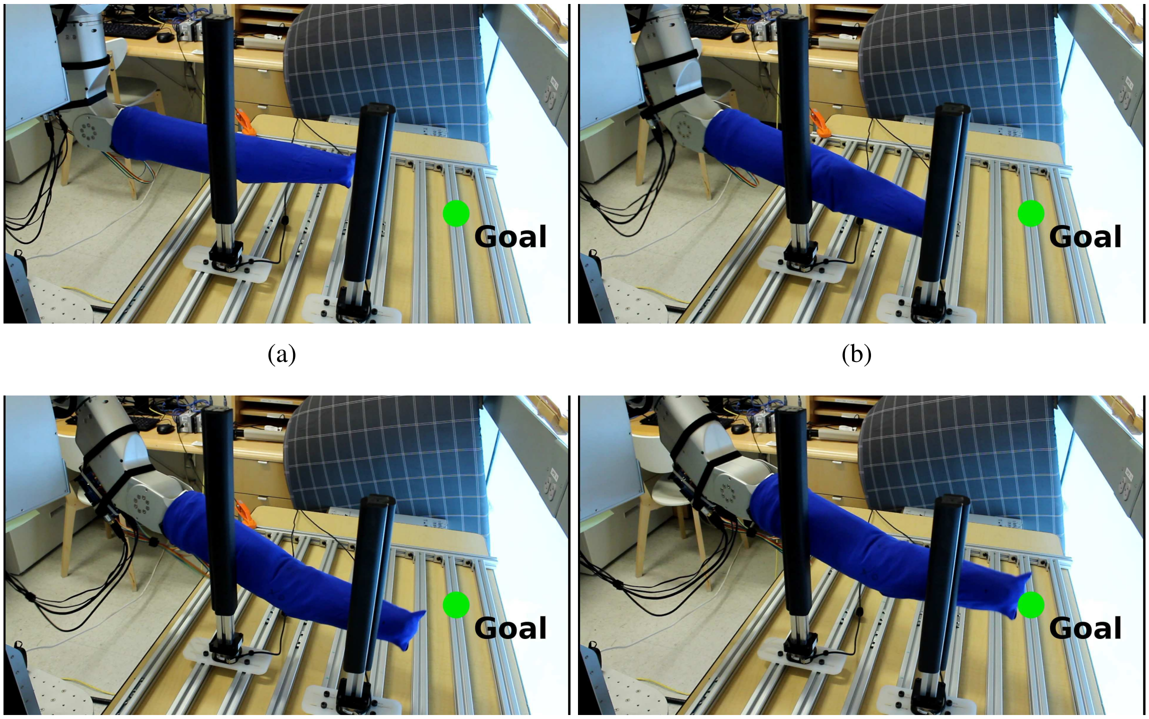


Figure 24: Canonical Test 5: Results for for reaching straight ahead while wedging between two posts and making contact with a third.



(a) (b)

(c) (d)

Figure 25: Canonical Test 6: DARCI reaching with two posts on the same side of the arm.

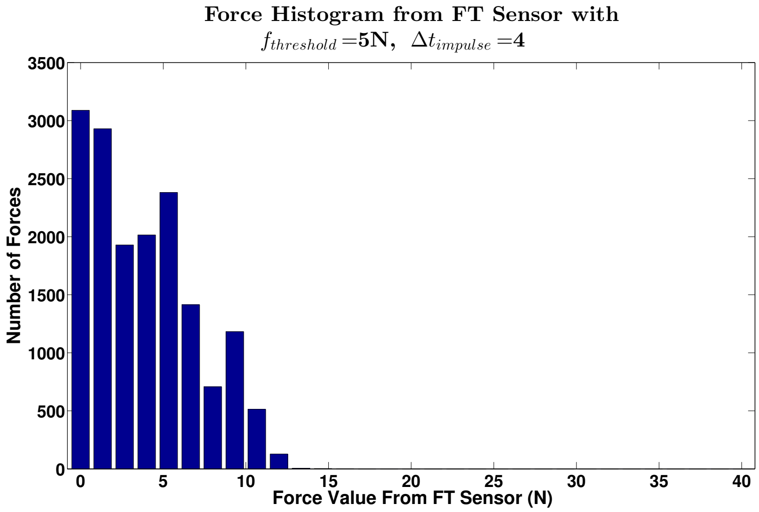


Figure 26: Canonical Test 6: Results for reaching while making contact with both posts on the same side of the arm.

Table 8 contains summary statistics of the forces for all of the canonical trials. These statistics include the mean of the maximum forces as well as the 99^{th} percentile of all contact forces and

Table 8: These are numerical values that summarize the data presented in the force histograms for the canonical test cases in this section.

Canonical Test #	99^{th} Percentile Force	Mean of All Forces	Mean of Max Forces	Max of Max Forces
1 ($\Delta t_{impulse} = 2$)	3.9 N	1.4 N	7.2 N	8.3 N
1 ($\Delta t_{impulse} = 4$)	17.2 N	4.9 N	19.1 N	19.6 N
2	26.7 N	7.9 N	23.5 N	39.3 N
3	8.6 N	2.5 N	8.1 N	10.0 N
4	15.9 N	7.5 N	13.8 N	25.3 N
5	18.3 N	5.1 N	13.3 N	28.1 N
6	11.2 N	3.9 N	8.7 N	15.7 N

the mean of all contact forces for each canonical test.

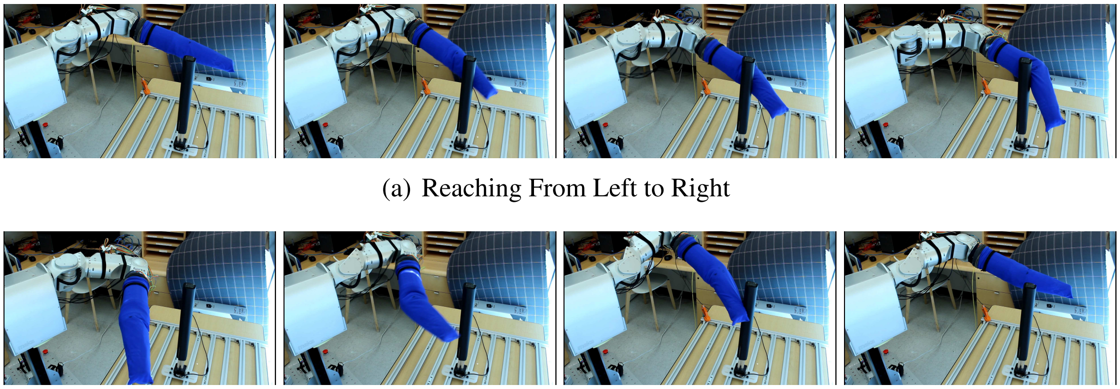
From previous work that we performed with Cody (see [11]), we know that this type of robot arm can easily apply upwards of 40 N on this type of force-torque sensor. Additionally, for previous trials using just compliance with the arm, every trial out of a total of six failed due to exerting forces over 40 N. While the quasi-static controller was successful at regulating forces and reaching the goal all 6 times. In our case, it is clear that our dynamic model predictive controller is also limiting contact forces to be within the same ranges we saw with the quasi-static controller in [11]. In many of the trials on DARCI in this chapter, there was no way to reach the goal, yet our dynamic controller regulated the forces to be generally low despite the fact that as it is noted in [11] "... the relationship between contact forces and taxel output is complex."

3.3.2 Control of High Velocity Impact Forces

In addition to the canonical tests that we developed for the previous section, we also performed a test where we varied the force threshold $f_{threshold}$ and the impulse-momentum parameter $\Delta t_{impulse}$ to explore the effect of these parameters on force control for unexpected impact and velocity at the end effector. These tests are related to the simulation tests we performed in the previous report.

In these tests we started the arm on either the left or right side of a single force-torque sensor post (the same as we used in the previous section). For each trial, the robot would reach to a goal on the opposite side of the post while making initial contact somewhere along the forearm, wrist or end effector. After making initial contact, the arm would continue to try and reach the goal using the dynamic MPC. In general, the arm was successful at reaching from right to left, but from left to right would get stuck in a local minimum. Figure 27 shows how this trial was executed.

For these trials we varied the force threshold ($f_{threshold}$) between 5, 10 and 15 N while varying

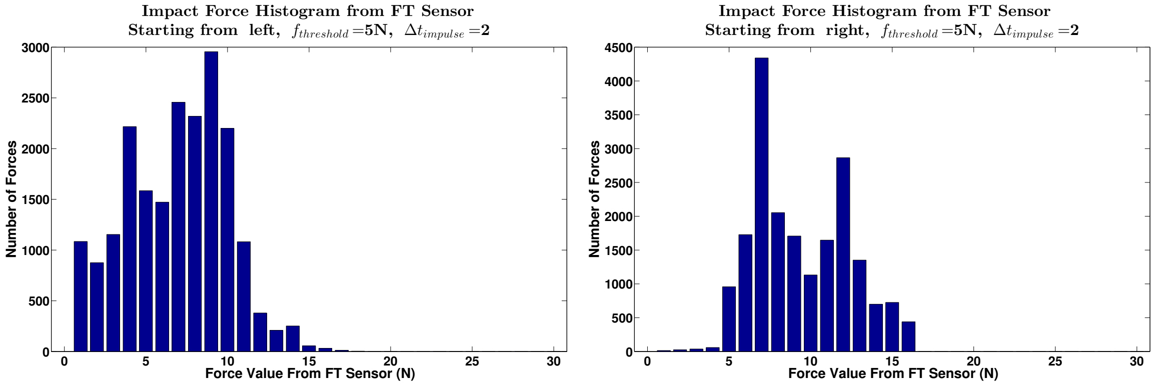


(a) Reaching From Left to Right

(b) Reaching From Right to Left

Figure 27: Sequence of images shows the arm reaching from left to right on top, and right to left on bottom while making contact with the post.

the impulse parameter ($\Delta t_{impulse}$) between 2, 4, 16 and 48. We ran 10 reaches for each direction (left to right and right to left) at each setting resulting in a total of 20 trials at each setting. We first look at the difference in force histograms for reaching from the right as opposed to reaching from the left. Figure 28 shows the force histograms for $f_{threshold} = 5$ N and $\Delta t_{impulse} = 2$.



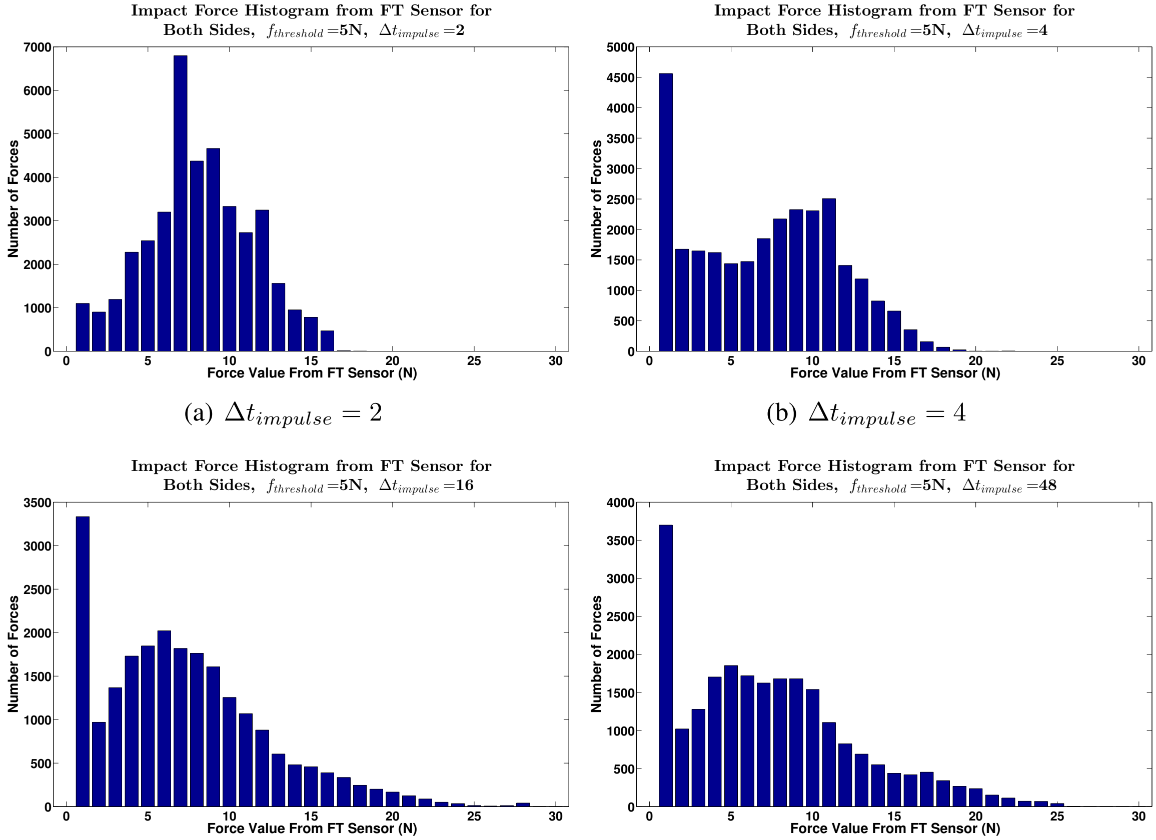
(a) Force Results for Reaching From Left to Right

(b) Force Results for Reaching From Right to Left

Figure 28: Force histograms for reaching from two different directions while making impact in about the same place in the workspace.

As can be seen in Figure 28 the force distributions are very different when reaching from one direction versus the other. We did not quantify why this was the case but expect that the kinematics and torque limits when moving more towards or away from the torso (meaning more or less torque being used for gravity compensation) may be part of the reason for this. In terms of practical use, this is an issue for further exploration in terms of robot design for manipulating in clutter. For the rest of the force histograms presented in this section we combine the forces from both the left and right reaches.

We next report the force and velocity results for holding $f_{threshold}$ fixed at 5 N while varying $\Delta t_{impulse}$ and for holding $\Delta t_{impulse}$ fixed at 4 while varying $f_{threshold}$. Figure 29 shows force histograms for keeping the controller force threshold fixed and varying $\Delta t_{impulse}$.



(a) $\Delta t_{impulse} = 2$

(b) $\Delta t_{impulse} = 4$

(c) $\Delta t_{impulse} = 16$

(d) $\Delta t_{impulse} = 48$

Figure 29: Force histograms for impacts while keeping the force threshold fixed and varying $\Delta t_{impulse}$.

For $\Delta t_{impulse}$ values of 2, 4 and 16 seconds the force distribution continues to shift to the right (with higher overall forces). It is important to note that we do not expect the time duration of the impact to physically last 2, 4 or 16 seconds. This parameter is multiplied by the expected moment arm of the impact force which we fixed to be equal to 2 cm from the joint. This means that in some cases $\Delta t_{impulse}$ loses some physical significance although it is still positively correlated with the maximum allowable joint velocity which was the purpose of this constraint. In addition, the tail relating to high forces gets longer which makes sense as the max force due to impact increases while for the duration of the rest of the same contact the force is regulated by the controller to be closer to the threshold. From 16 to 48 the difference is almost negligible. This is likely due to the fact that at $\Delta t_{impulse} = 16$, we are reaching known software limits on the joint velocity implemented by the company Meka who produced DARCI.

The trends that result from varying $\Delta t_{impulse}$ can be represented more succinctly by plotting

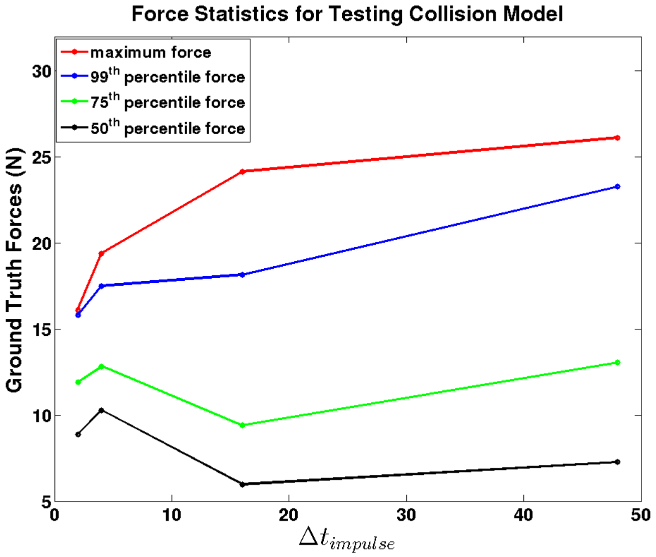
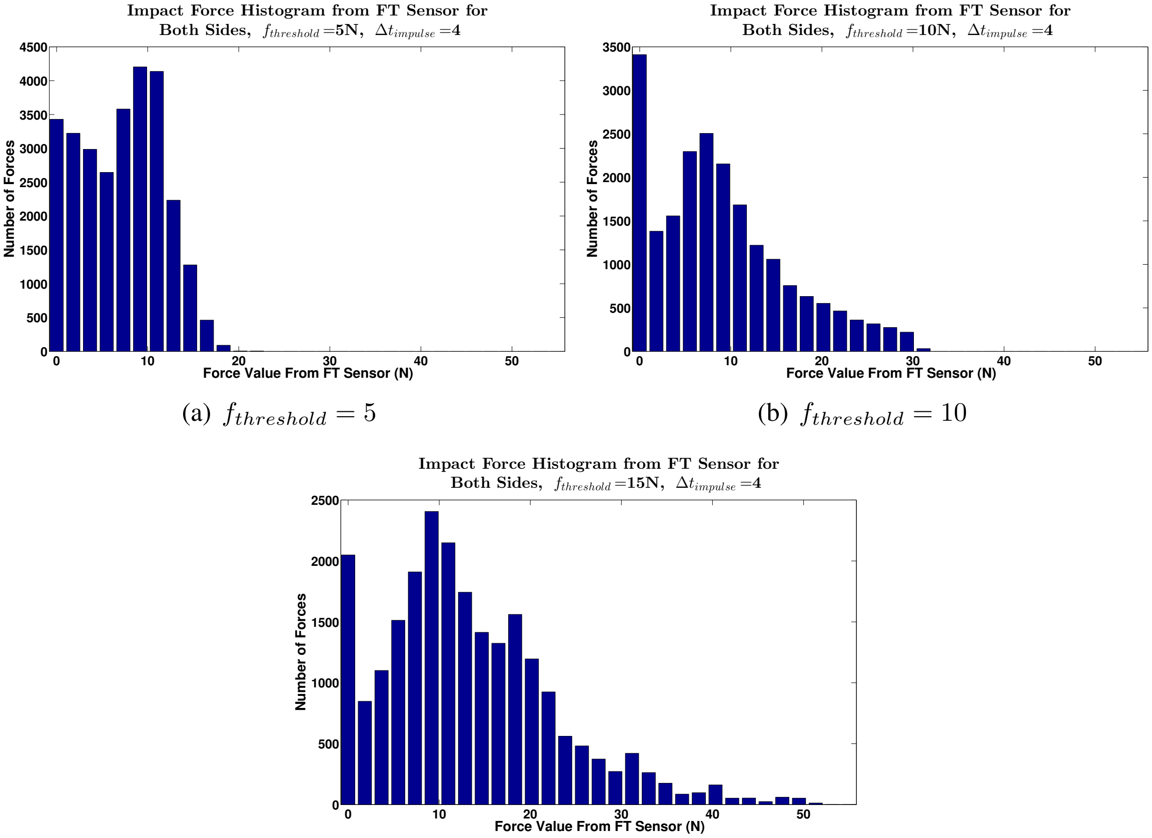


Figure 30: Maximum forces, 99^{th}, 75^{th}, and 50^{th} percentile forces as $\Delta t_{impulse}$ increases. Each data point represents 20 trials worth of data measuring the ground truth forces with a force-torque sensor at 100 Hz.

summary statistics for each set of 20 trials. The statistics that we plotted include the maximum contact force, and the 99^{th}, 75^{th}, and 50^{th} percentile contact force for each value of $\Delta t_{impulse}$. The results can be found in Figure 30. As expected, we can see that the maximum force increases rapidly before reaching an apparent asymptote that is caused by the low-level software limits that limited the maximum joint velocity for these tests. However, almost equally interesting is the fact that for both the 75^{th}, and 50^{th} percentile forces, the forces varying only a small amount around a specific force value as measured by the force-torque sensor instead of increasing as $\Delta t_{impulse}$ increased. This gives evidence to the fact that our controller is still able to control the majority of the forces to be near the threshold even when the maximum forces due to impact increase because of moving in clutter at higher speeds.

In addition to varying $\Delta t_{impulse}$ we wanted to verify that varying the force threshold ($f_{threshold}$) also had the expected effect on the real robot. Figure 31 shows force histograms for keeping $\Delta t_{impulse}$ fixed and varying the controller force threshold.

The max and mean forces of the force distributions shown in Figure 31 increase as $f_{threshold}$ increases. We expect that at higher forces, the calibration of our skin sensor will be more prone to error. In addition in many of these cases with higher force, the arm was making contact on the wrist with two different taxels making some of the these total max forces somewhat more reasonable with respect to the specified controller force threshold. Although the tactile sensor can be improved upon in terms of both sensor density and calibration with real forces, we can see that $f_{threshold}$ has the exact effect that we would expect and that we saw in simulation. The effect is that increasing the force threshold, $f_{threshold}$, in the the controller is positively correlated with the maximum and 99^{th} percentile forces that we measured while reaching in clutter. Figure



(a) $f_{threshold} = 5$

(b) $f_{threshold} = 10$

(c) $f_{threshold} = 15$

Figure 31: Force histograms for impacts while keeping $\Delta t_{impulse}$ fixed and varying the force threshold.

32 shows the summary statistics for each set of 20 trials. The statistics that we plotted include the maximum contact force, and the 99^{th}, and 50^{th} percentile contact force for each value of $f_{threshold}$. The correlation coefficient between the force threshold value and the 99^{th} percentile forces is 0.99975.

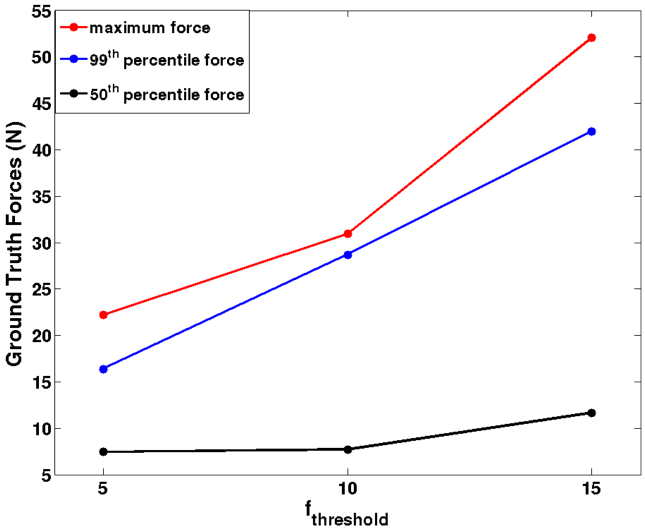


Figure 32: Maximum forces, 99^{th}, and 50^{th} percentile forces as $f_{threshold}$ increases. Each data point represents 20 trials worth of data measuring the ground truth forces with a force-torque sensor at 100 Hz.

References

[1] C. Zhu, R. H. Byrd, P. Lu, and J. Nocedal, "Algorithm 778: L-bfgs-b: Fortran subroutines for large-scale bound-constrained optimization," *ACM Trans. Math. Softw.*, vol. 23, no. 4, pp. 550–560, Dec. 1997. [Online]. Available: http://doi.acm.org/10.1145/279232.279236

[2] M. D. Killpack, "Model predictive control with haptic feedback for robot manipulation in cluttered scenarios," Ph.D. dissertation, 2013.

[3] J. L. Garriga and M. Soroush, "Model predictive control tuning methods: A review," *Industrial & Engineering Chemistry Research*, vol. 49, no. 8, pp. 3505–3515, 2010.

[4] G. Bunin, F. Fraire, G. François, and D. Bonvin, "Run-to-run mpc tuning via gradient descent," in *22nd European Symposium on Computer Aided Process Engineering (London)*, 2012, pp. 927–931.

[5] J. Lee and Z. Yu, "Tuning of model predictive controllers for robust performance," *Computers & chemical engineering*, vol. 18, no. 1, pp. 15–37, 1994.

[6] A. Al-Ghazzawi, E. Ali, A. Nouh, and E. Zafiriou, "On-line tuning strategy for model predictive controllers," *Journal of Process Control*, vol. 11, no. 3, pp. 265–284, 2001.

[7] J. Fan, G. E. Stewart, *et al.*, "Automatic tuning method for multivariable model predictive controllers," Aug. 18 2009, uS Patent 7,577,483.

[8] J. Mattingley and S. Boyd, “Cvxgen: a code generator for embedded convex optimization,” *Optimization and Engineering*, pp. 1–27, 2012.

[9] T. Bhattacharjee, A. Kapusta, J. Rehg, and C. C. Kemp, “Rapid categorization of object properties from incidental contact with a tactile sensing robot arm,” 2013.

[10] A. Jain, M. D. Killpack, A. Edsinger, and C. Kemp, “Reaching in clutter with whole-arm tactile sensing,” *The International Journal of Robotics Research*, 2013.

[11] T. Bhattacharjee, A. Jain, S. Vaish, M. D. Killpack, and C. C. Kemp, “Tactile sensing over articulated joints with stretchable sensors,” 2013.

Multi-contact Variable Compliance Manipulation in Extreme Clutter

DARPA Maximum Mobility and Manipulation (M3)
Quarterly Technical Report #7
Covers Oct 01, 2013 to Dec 31, 2013
Submitted: February 06, 2013

PI: Charles C. Kemp (Georgia Tech)
Co-PI: Aaron Edsinger (Meka Robotics)

1 Overview

In the fourth quarter of 2013 we made the following progress:

- Developed integrated system for dynamic reaching in clutter
  - Combines our work on this project
    - Dynamic MPC controller
    - Tactile-based sensing
    - Online 3D haptic mapping of objects based on categorization of object properties
    - Learned initial conditions
    - Cost-based planning over sparse maps
  - Preliminary results:
    - Performs complex, multi-step reaching behaviours on the robot DARCI.
    - Reaching behavior uses fastest, simplest behaviors first.
    - System haptically maps environment during reaching.
    - Geometric planning over sparse haptic map used when greedy reaching fails.
    - Improved success in more diverse situation compared with individual components.

2 Introduction

Humans and other animals readily reach into complex environments without visually observing the detailed contents. During the day-to-day manipulation tasks, humans frequently come into 'incidental contact' with objects in their environments as shown in Fig. 1. By incidental contact, we mean any contact that occurs unintentionally while performing goal-directed manipulation tasks. Being able to reach into various environments without the need of avoiding contact with objects, would be a generally useful capability for robots in a variety of application areas, including assistive robotics [1]. Within this tech report, we describe an integrated system for robotic control that enables a robot to reach locations in unmodeled, cluttered environments solely based on joint-angle, joint-torque, and tactile sensing (See. Fig. 2) from 'incidental contact'. The system builds on our previous research in a number of ways, including integrating a variety of system components, both published and unpublished. We designed our system to first use efficient, memory-free greedy reaching followed, if necessary, by resource-intensive geometric planning using a map. A motivating intuition for this structure is the common human experience of reaching to a location without paying much attention, and then realizing that one needs to pay careful attention in order to succeed.

(a) (b) (c) (d)

Figure 1: Humans and animals frequently come into contact with the environment while reaching into clutter. (a) A raccoon reaches into a bird house to find eggs and young. (b) When noodling, people find catfish holes from which to pull fish out. (c)-(d) A person makes contact along his forearm while reaching for objects in a cluttered cabinet and refrigerator. (All images used with permission)

3 Overview

3.1 System Architecture

Figure 3 illustrates the architecture of our system. At all times, our system uses the newest version of our model predictive controller from [2] to control the robot at 25 Hz. It

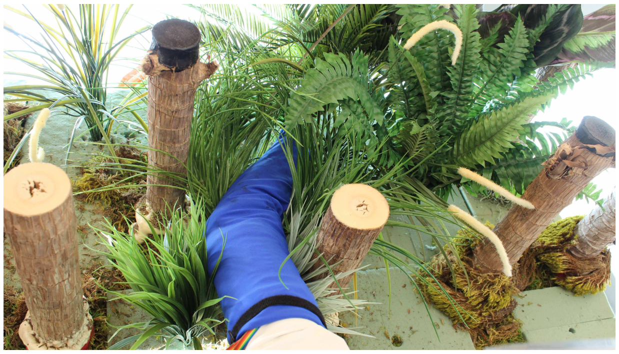


Figure 2: The DARCI Robot reaching through dense foliage using the integrated system described in this paper.

attempts to reach either an end-effector pose or an arm configuration while keeping contact forces low. This model predictive controller runs on top of gravity compensation and an impedance controller that simulates low-stiffness visco-elastic springs at the robot's joints running at 1 kHz.

When a desired end effector goal for the system is received, the system first attempts to bring the arm to an initial configuration which has performed well in similar circumstances using the 'Learned Initial Condition' module described below. The system then uses the model predictive controller to greedily reach to the goal location from this initial arm configuration. As we presented in [3], two greedy reaches from random locations can achieve over an 80% success rate in certain types of clutter, and we have found that using learned initial conditions (LIC) can result in a significantly higher success rate. Greedy reaching has the advantages of not requiring a map, having relatively low computational requirements, and making efficient use of redundant degrees of freedom. However, greedy reaching can become stuck in local minima and so does not always succeed in finding a solution. For example, in [3] around 10% of the situations encountered were not reached after 5 greedy reaches from random initial arm configurations.

In order to handle these situations, our system makes use of geometric planning based on a map of locations that our tactile recognition system has estimated to be impassable. The map is generated by the classification of object properties from incidental contact during the initial greedy reaches, and is continually updated throughout an attempt to reach a goal location. While more computationally intensive, planning has the advantage of being able to find solutions for situations requiring complex sequences of arm configurations, where a greedy behaviour fails. If the robot becomes stuck while attempting to follow a planned sequence of arm configurations, the system attempts another greedy reach toward the goal, as the initial stages of a planned path are sometimes able to bring the arm clear of the obstacles preventing the originial greedy reaching attempts from succeeding. If this reach fails, the system replans using the most recently updated map. Using this method, the maps over which trajectories are planned are relatively sparsely populated with known

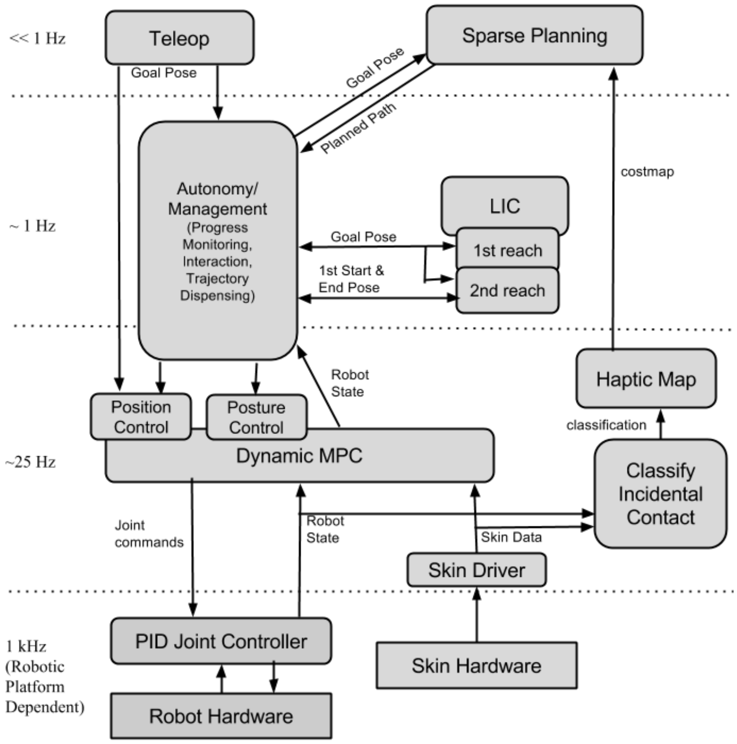


Figure 3: Block diagram showing the integrated system architecture. High update rate processes, such as low-level joint control, appear at the bottom of the diagram, while slower-updating processes are presented higher up. The teleoperation interface is only used to provide a single goal end-effector pose, after which the integrated system proceeds autonomously.

obstacles, but initial work has shown this to be sufficient to produce useful behaviors from the controller. At any point, if the planner fails to return an acceptable path in a reasonable length of time, the arm is pulled back to the most recent starting position, and a new plan is requested from this location, where the arm is less likely to be closely surrounded by obstacles. Besides having relatively large computational requirements, a disadvantage of the planning system is that, unlike greedy reaching, it does not reactively take advantage of the robot's redundant degrees of freedom, and instead needs to replan in the event of becoming stuck. However, this is outweighed by the ability to perform non-greedy actions which may be necessary to reach a desired goal in complex clutter.

The pseudocode in Algorithm 1 provides an overview of the way in which the integrated system functions.

4 System Components

We now provide brief summaries of our system's components.

Algorithm 1 Integrated System Procedure.

```
Require: $GoalPose\ g$
  BEGIN ONGOING CONTACT CLASSIFICATION
  BEGIN ONGOING HAPTIC MAPPING
  $HapticMap\ map \leftarrow blocked\_locations$
  $LIC1\_Config\ lic1\_cfg \leftarrow$ LIC1($g$)
  DYNAMICMPC($lic1\_cfg$)                       ▷ Setup at LIC1 start config
  DYNAMICMPC($g$)                               ▷ First greedy reach to goal
  if at g then
      return
  else
     $StuckPose\ s \leftarrow current\_pose$
  end if
  DYNAMICMPC($RetreatPose\ r$)
  $LIC\_Config\ lic2\_cfg \leftarrow$ LIC2($g,\ lic1\_cfg,\ s$)
  DYNAMICMPC($lic1\_cfg$)                       ▷ Setup at LIC2 start config
  DYNAMICMPC($g$)                               ▷ Second greedy reach to goal
  if at $g$ then
      return
  else
     repeat
        if $Path\ p \leftarrow$PLAN($g, map$) then   ▷ Request path from planner
      DYNAMICMPC($p$)                           ▷ Follow configuration plan
        else
      DYNAMICMPC($RetreatPose\ r$)              ▷ Pull out if planner fails
      CONTINUE                                  ▷ Replan from outside clutter
        end if
     until at g
  end if
```

4.1 Dynamic MPC

Moving a robot arm in cluttered, unknown, and unmodeled workspaces can be difficult as interaction with obstacles can block paths and generate high contact forces. We use a model predictive control (MPC) controller that explicitly models the robot arm dynamics with tactile sensing to move the robot arm quickly and control contact forces as the arm moves towards its goal. We implemented an updated version of our dynamic model predictive control (MPC) controller from [2] that runs on our humanoid robot, DARCI, and that adds additional functionality to work with the various modules of our integrated system. Our dynamic MPC controller moves towards a designated goal position while keeping contact forces and worst-case, unexpected-impact forces low. We added an integral controller term when near the goal location to compensate for errors in the robot's gravity compensation. Our MPC controller uses a control horizon of 3 and prediction horizon of 4 which gives the controller three time steps of control and predicts the arm output for 4 additional time steps over which it aims to minimize its cost function. We added the functionality to the controller to receive a joint configuration posture in addition to the option of a cartesian end-effector goal location. In the posture-control version of our control we removed the limit on the rate of change of contact forces to improve computational performance. Posture goals are the method by which goals are sent from the planner to the controller and the method by which the arm is extracted from clutter. Due to differences in the optimizaiton between pose- and posture-control modules, and in order to keep each optimization as small as possible, two separate control modules are run in parallel throughout the demonstration, one for pose-control and another for posture-control. When one is active, the other is set to a waiting state, wherein it does not solve the optimization or send commands to the low-level joint controllers. This avoids conflicting commands to joint controllers and reduces the computational requirements of the control system.

4.2 Learned Initial Conditions

In this section, we describe learning and prediction schemes for identifying good initial configurations during manipulation in clutter. We have shown that reaching a goal in clutter may require multiple attempts before succeeding [3]. However, if we can identify initial configurations which result in successful reaching, we can significantly decrease the number of required retries [4].

4.2.1 Learning initial conditions without detailed knowledge

Prior to observing the environment in which manipulation is to take place, and without detailed knowledge of the environment, we define the problem of the selecting the best initial condition as

$$\begin{array}{ll} \underset{x_0}{\text{maximize}} & P(x_\infty = g | x_0) \\ \text{subject to} & x_0 \in \text{open space}, \end{array} \tag{1}$$

where $x_0 \in \Re^6$ is the initial pose of the end effector before beginning a greedy reaching behavior, $x_\infty \in \Re^3$ is the final stopping position, and $g \in \Re^3$ is the goal position. x_0 must satisfy joint constraints. In addition, we constrain x_0 to lie in open space outside the cluttered region of interest.

Given an environment v for which we only know the category c, the marginal probability density function of the selection problem is written as Eqn. 2. If the properties of v are similar to the environments V_c, which have been explored, we can approximate the marginal probability distribution as follows:

$$\begin{aligned} P(x_\infty = g|x_0) &= \int_v P(x_\infty = g|x_0, v)dv \qquad (2) \\ &\approx \int_{V_c} P(x_\infty = g|x_0, v')P(v')dv', \qquad (3) \end{aligned}$$

where v' is a map in an experienced environment set V_c. Thus, given a goal from past trial experiences in the same or similar environments, we can predict the probability of the best condition. We will use 'LIC-1' (learning an initial condition for a first reach into a new cluttered environment) to denote the framework in Eqn. 3.

4.2.2 Learning initial condition with observations

After one attempt, we have obtained observations o about the environment v, and we can adapt the initial condition to improve the probability of success. This problem can be written as

$$\begin{aligned} \underset{x_0}{\text{maximize}} \quad & P(x_\infty = g|x_0, o) \\ \text{subject to} \quad & x_0 \in \text{open space}, \end{aligned} \qquad (4)$$

where x_0 denotes the restart condition and o denotes observed information from the previous trial. In this system, we define o as

$$o = \{x'_0, x'_\infty\}, \qquad (5)$$

where x'_0 denote the previous initial condition and x'_∞ is the final position of the previous trial. Similar to LIC-1, we compute the marginal probability conditioned on the observation. We denote this second framework 'LIC-2.'

For the implementation in this paper, we trained the model using a large number of successful- and failed-trial samples in a simulation environment, shown as Fig. 4. This clutter includes 60 fixed-floating spheres, each with a 0.05 m radius, in a 0.5 m x 0.9 m x 0.6 m rectangular parallelepiped area in front of a simulated DARCI robot. The robot tries to reach to 15 grid-distributed goals of size 5 x 3 in 20 different clutters from 28 initial conditions. The goals were placed behind of a set of spheres on a vertical, rectangular plane 0.6 m wide and 0.3 m tall, at 0.15 m intervals. The initial positions were equally distributed on a vertical, rectangular plane 0.6 m wide and 0.3 m tall, at 0.1 m intervals. We ran 22,684 trials for the sampling of trials. Using simulated or real-world trials that

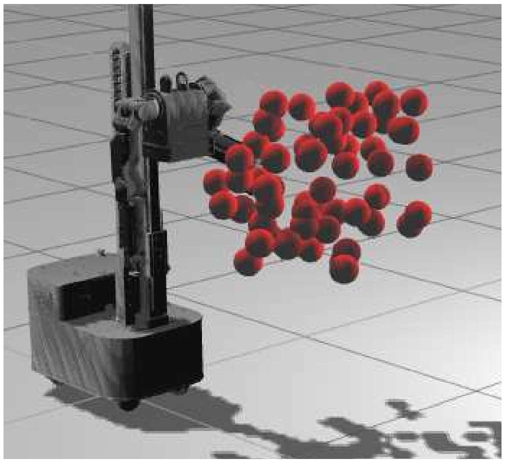


Figure 4: Training environment in GAZEBO. Training for LIC is performed in simulation prior to the real demonstration. We use 60 fixed-floating spheres with 0.05 m radius in a 0.5 m x 0.9 m x 0.6 m rectangular parallelepiped area in front of DARCI to simulate a densely cluttered environment.

more closely match the target environment would be likely to improve performance. Here, we used spheres in 3D as a generic notion of clutter.

During the demonstration, each module, trained in the environment, returns an initial configuration of the robot arm based upon the goal pose received (in the case of LIC-1) and also based on the initial and final (unsuccessful) pose of the first reach (in the case of LIC-2). The robot then moves to the indicated initial configuration before executing a greedy reaching behaviour using the dynamic MPC controller.

4.3 Greedy Reaching

Once the robot reaches an initial configuration identified by the LIC module, the central interaction manager sends the goal pose to the dynamic MPC controller, which executes a greedy reaching behavior toward the goal while maintaining low contact forces with the environment. Because the controller limits contact forces along the arm, it often moves along even rough obstacles without becoming stuck against them, enabling it to reach seemingly difficult-to-reach goals.

However, the controller can become stuck against relatively simple obstacles, such as artificial foliage, if it finds a local minimum such that greedily reducing the control error will not advance the end effector toward the goal. The controller is deemed to have failed or become stuck if it fails to reduce the distance from the goal by one tenth within a 4 second period. In these cases, the greedy nature of the controller prevents it from discovering alternative paths which might allow it to reach the goal successfully. In such cases, the complete system is able to compensate for this shortcoming by providing increasingly more-informed plans, both in the form of LIC-2 initial conditions, and through haptic mapping and geometric planning.

4.4 Extracting the Arm

After performing a greedy reach, the robot must extract its arm from the cluttered environment. To accomplish this, we have explored multiple methods. The first, more suitable for simpler environments, records the trajectory of the end effector as the greedy reach is performed, adding an additional point to the path once the end effector has traveled more than 1 *cm* from the previous point. Upon completing a greedy reach, the interaction module then uses the greedy dynamic MPC controller to bring the end effector to each pose in the recorded path, in the reverse order. The next goal is given once the end effector is within 5 *cm* of the currently-assigned goal. This method does not constrain the redundant degrees of freedom in the arm, but still preserves some of use of the clear path which was found by the dynamic MPC controller while reaching into the clutter.

A second method records the joint configurations of the robot as it performs the greedy reach, and then uses the posture-controlling dynamic MPC to return to each full arm configuration in the reverse order. The module records a new configuration each time any joint in the arm moves more than 3 degrees from its position in the previous history entry. In order to better take advantage of the ability of the MPC controller to resolve constraints on the arm, a configuration along the return path is considered reached when the angles of the arm are all within 3 degrees of the desired configuration. This method is often less successful in simple environments, where the former method allows the MPC controller to avoid simple obstacles, but is more successful in complex configurations with complex obstacles, where it can more exactly trace the clear path that was found during reaching.

4.5 Haptic Mapping

During manipulation in cluttered environments, unintentional or ‘incidental’ contact with objects can be frequent. The information from these incidental contacts could be potentially used to infer properties of the environment. These inferred properties can in-turn help in intelligent manipulation planning strategies. However, rapid identification of haptic properties of objects in unknown environments during exploration or navigation is a difficult problem. In this section, we demonstrate that data-driven methods can be used to rapidly categorize objects encountered through incidental contact on a robot arm.

We use hidden Markov models (HMMs) to model the time-series contact force data from the fabric-based tactile sensor and use the models to classify the objects in the environment into the categories of ‘rigid’ and ‘soft.’ The elements which constitute an HMM are (1) N, the number of states in the model; (2) M, the number of distinct observation symbols per state; (3) A = $\{a_{ij}\}$, the state transition probability distribution; (4) B = $\{b_j\,(k)\}$, the observation symbol probability distribution; and (5) P = $\{\pi_i\}$, the initial state distribution [5–7]. It is represented as given in eq. (6), where the parameter λ describes the HMM model.

$$\lambda = (A, B, \pi) \tag{6}$$

We trained the two HMM models (Rigid and Soft) using training data collected on the robot platform 'Cody' with an artificial skin on its forearm, on environments composed of small tree trunks (rigid objects) and artificial leaves (soft objects) [5] as shown in Fig. 5. We used the quasi-static MPC controller from [3] for manipulation in these cluttered environments. We had two HMM models which we trained on the leaf and trunk environments. We trained the HMMs by choosing the λ which locally maximizes $P\left(O|\lambda\right)$ iteratively using expectation-maximization (EM) techniques [6]. After we train the models λ_T for trunk and λ_F for leaf, we evaluate a new observation sequence $O = \{O_1, O_2, ...O_n\}$ according to eq. (7) which gives us the model which best matches the observation sequence. The third step in eq. (7) leads to the fourth step, if all the models are equally likely [5].

$$\begin{aligned} c^* &= \underset{c\in[T,F]}{\arg\max}\, P\left(\lambda_c|O\right) \\ &= \underset{c\in[T,F]}{\arg\max}\, \frac{P\left(O|\lambda_c\right) P\left(\lambda_c\right)}{P\left(O\right)} \\ &= \underset{c\in[T,F]}{\arg\max}\, P\left(\lambda_c|O\right) P\left(\lambda_c\right) \\ &= \underset{c\in[T,F]}{\arg\max}\, P\left(\lambda_c|O\right) \end{aligned} \tag{7}$$

During this demonstration for testing, we are using the dynamic MPC and the robot DARCI, with the flexible and stretchable fabric-based tactile sleeve, but still in an environment composed of trunks and leaves. The robot, DARCI, and the environment are shown in Fig. 2. We run the HMM models to classify, live and in real-time, the contact force data for every taxel on the tactile sleeve.

We classify the objects in the test environment into rigid and soft categories using the log-likelihood values of the two HMM models. We create a haptic map in Rviz visualization software by mapping all the rigid taxels at every time-instant. For visualizing the haptic map, we use point cloud/voxels for every taxel that is categorized as rigid. Each taxel with rigid contact is mapped using a dark brown sphere as shown in Fig. 6. This information is provided to the planner described in Sec. 4.6 so that it can avoid these areas of rigid contacts and come up with an intelligent planning strategy.

4.6 Planning with Contact

In this section, we describe a global search-based planner with a traverversability map constructed by the haptic classifier described in Sec. 4.5.

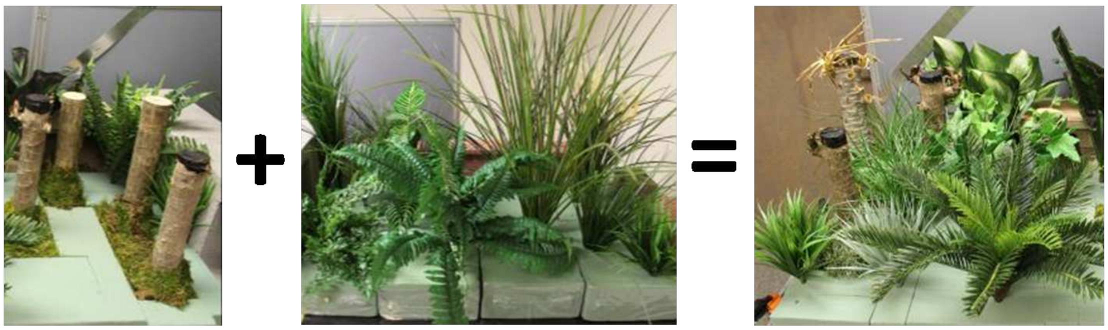


Figure 5: (Left) Trunk-only environment for training the HMM model for Trunk Category; (Middle) Leaf-only environment for training the HMM model for Leaf Category; (Right) Combined environment for testing.

4.6.1 Traversability Map

To use a planner in a cluttered environment, we first construct a 3D traversability map. We represent the workspace of the robot as a 3D voxel grid with 0.01 m x 0.01 m x 0.01 m voxel size in Cartesian space. Each voxel includes a traversability metric that shows the manipulation cost in that location. We define the traversability value as a scalar value between 0 to 100. The higher value a voxel has, the more difficult it is for the arm to pass through the voxel's location. In this demonstration, the robot knows what kind of object it is colliding with based on the haptic classifier of Sec. 4.5. This allows for updating the traversability map online during reaching. For this demonstration, we assign manipulation costs of 0, 50, and 100 into empty area, movable or soft object area, and fixed-rigid object area, respectively.

The area of map is defined as a rectangular box, 0.6 m x 0.7 m x 0.6 m in front of the robot. It is initially populated with zeros, assuming that the unknown environment is empty and that there is little cost associated with manipulating the arm in that area. The map records the contact information using Point Cloud Library's (PCL) Voxel Grid [8].

4.6.2 Traversability Planner

The traversability planner has two main steps: goal posture selection and trajectory planning. The goal posture is randomly selected from a list of valid arm postures. Valid arm postures are joint configurations such that the end-effector reaches a Cartesian goal, and the entire arm is placed in low-cost area. In detail, to create the list of the initial posture, 72 uniformly distributed orientations are sampled using the sampleSO3 function from OpenRAVE [9]. To check the cost of a path, we construct a traversability checker that computes the traversability of each vertex location from the arm collision meshes at each joint state, and rejects the state when the vertices are located inside of fixed-rigid object area of the map.

For trajectory planning, we use a global search-based planner, RRT-Connect [10] from

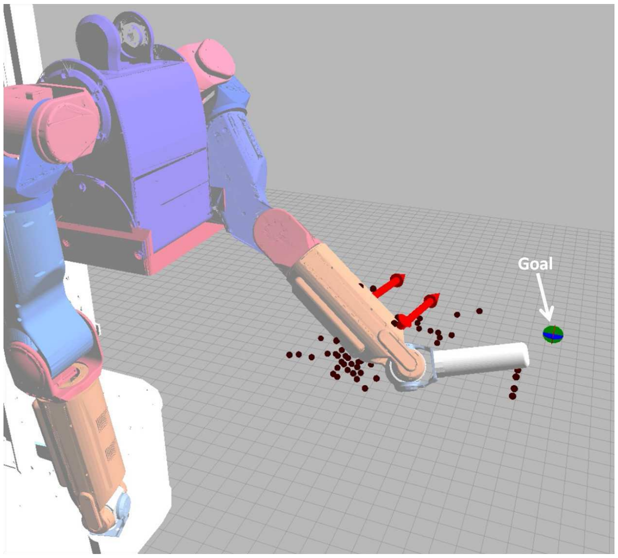


Figure 6: Haptic Map of detected rigid contacts.

OMPL [11]. It plans a path over the traversability map in joint space. Any arm posture in a high-cost configuration is rejected by the traversability checker. In this demonstration, we assume all other area is traversable except the rigid-fixed contact area. One example of a robot configuration returned by the planner using haptic map is shown in Fig. 7

4.7 Implementation

We now describe our software and hardware implementation of the system.

4.7.1 Tactile Sensor

For tactile sensing, we use the fabric-based tactile-sensing sleeve we described in [12]. The sleeve is made of five layers of stretchable fabric. The inner and outer layers are electrically insulating, and isolate the inner layers from the robot and external world, and provide protection from abrasion. The middle of the skin contains two layers of electrically conductive fabric (a silver-plated Nylon/elastic fiber) separated by an electrically resistive fabric (a conductive-polymer coated Nylon/elastic fiber). The inner counductive layer consists of 25 individual patches of conductive fabric, each of which forms a sensing region, or ‘taxel’ for ‘tactile pixel.’ Each patch is supplied with 5V via a pullup resister and an Arduino board. The outer conductive layer is a single sheet covering the entire sleeve, and is connected to the ground of the Arduino. As the central resistive fabric is compressed, the conductivity across the compressed portion of the fabric increases, and a drop in voltage can be detected by the Arduino in the curcuit of the underlying taxel. This process is nonlinear, and depends upon both the force applied and the area over which contact

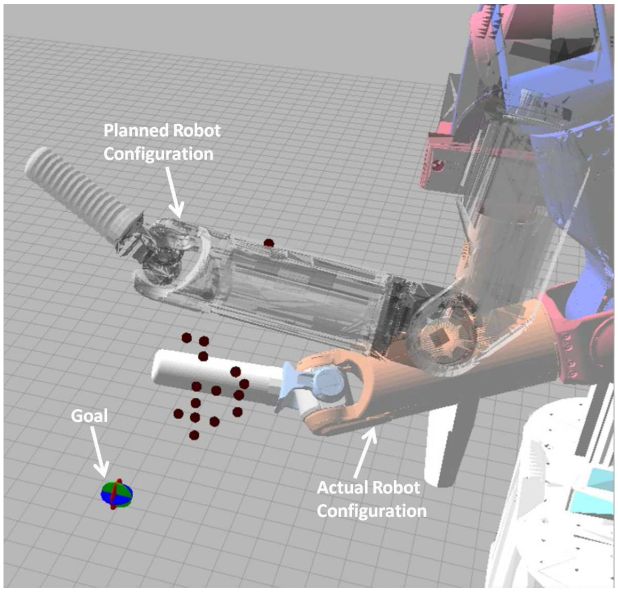


Figure 7: Planned Robot Configuration with Haptic Map.

occurs. However, we have found in practice that good performance can be obtained in our various systems by operating on sensor measuresments directly.

4.7.2 Robot Platform

The robot used in this work is the humanoid robot DARCI, an M1 Mobile Manipulation Platform from Meka Robotics, which includes a mobile base, a torso on a linear actuator, and two 7-Degree of Freedom arms. For the demonstration described here, the mobile base was not moved while the robot was performing the reaching task, and the torso remained fixed at its maximum height. We perform all demonstrations using the tactile sensing sleeve on the left arm of the robot, which is extended with a 3D-printed cylidrical extension of ABS plastic. The arms of the robot use a series elastic actuators at the joints, and are controlled to provide gravity compensation and an impedance controller that simulates low-stiffness visco-elastic springs at the robot's joints.

4.7.3 Software

The software for this demonstration consists primarily of Python code, with some portions being written in C++. The system is coordinated using the Robot Operating System (ROS) [13] for communication between the various modules, as well as for communication with the low-level controllers on the robot arm. The modules described above (Sec. 4) are typically each contained in a single process, or 'node,' in the ROS framework. Individual modules make heavy use of various software libraries related to their specific functions, as noted above. In particular, the Model Predictive Controller uses the CVXGEN [14] library for solving a convex optimization in determining the control inputs to the low level controller at each time-step. The state of the system is observed using the ROS Rviz visualization engine to visualize the state of the robot, the location and sensor readings of contacts on the tactile sensor, the active goal location, and the current state of the haptic map. Rviz also allows goals to be identified using the 'interactive marker' interface. This interface is used extensively in development and testing of this demonstration. During the demonstration itself, a goal location is first identified by manually bringing the end effector of the disengaged robot to a desired goal location, and a Python script stores the location of the end effector based on the robot's kinematics. This script later sends this goal position to the system.

5 Preliminary Trials

Our initial demonstrations of the combined system are performed on the trunk-and-foliage environment described in Sec. 4.5. Our results, in testing the demonstration system, agree strongly with the results of [3], in showing that many of the reachable goal poses can be reached by a single greedy reach. This is especially true when combined with the LIC-1 module, which provides informed starting configurations. These informed starting configurations serve to avoid many of the failures that would otherwise be expected from a greedy reach, an effect that is likely enhanced by the parallel-columnar nature of our rigidly fixed obstacles. On a few occasions, its observed that the LIC-1 initial condition would fail, and in these cases the LIC-2 condition has somnetimes succeeded, usually by starting from a slightly different location and so avoiding whatever obstacle was responsible for the failure of the first reach. There were a few cases in which both the LIC attempts failed. There were also few cases in which the planner suceeded in reaching to a goal after LIC-1 failed. While contrived examples can show that the planner is able to produce useful plans in cases where LIC fails, the sparse nature of the haptic map typically necessitates more reaches before the obstacles to be avoided is sufficiently well-explored in the haptic map for the planner to effectively avoid the real obstacle. A probable solution would be to add constraints such that the ground and the ceiling are treated as obstacles in the planner. We have been unable to identify a goal inside of our current cluttered environment for which the geometric planner is successful in a reasonable timeframe after the failure of both LIC-1 and LIC-2. However, it seems likely that such situations do exist, and that more complex clutter may significantly enhance the relative ability of the planner to provide value to the complete system.

6 Conclusion

We have presented an integrated robotic system capable of haptically reaching locations in cluttered environments. The system does not require detailed information about the environment in advance. When provided with a goal location, it moves to an arm configuration that it has learned from offline simulation works well in similar circumstances and then greedily attempts to reach the goal. If this fails, after extracting its arm from the environment it moves to another arm configuration that it has learned works well based on the nature of failure in the first reach. It then greedily reaches to the goal again. While the system is operating, it uses tactile recognition to detect impassable locations based on incidental contact and continually updates a map of the environment with this information. If the system does not reach the goal via these two greedy reaches, it plans and re-plans paths to the goal based on this constantly updating map, withdrawing the arm from the clutter if the planner is unable to find a path in a reasonable amout of time. In our demonstration, the robot successfully reached goals using greedy reaching and planning. Further testing and debugging of system components and integration challenges is ongling, and careful evaluation of our final systems performance using the robot DARCI will be performed, with detailed results being provided in a subsequent report.

References

[1] P. Grice, M. Killpack, A. Jain, S. Vaish, J. Hawke, and C. Kemp, "Whole-arm tactile sensing for beneficial and acceptable contact during robotic assistance," in *ICORR, 2013 IEEE*, Seattle, WA, USA, Jun. 2013.

[2] M. D. Killpack and C. C. Kemp, "Fast reaching in clutter while regulating forces using model predictive control," in *Humanoid Robots, IEEE-RAS International Conference on*, Atlanta, GA, USA, Oct. 2013.

[3] A. Jain, M. D. Killpack, A. Edsinger, and C. C. Kemp, "Reaching in clutter with whole-arm tactile sensing," *International Journal of Robotics Research (IJRR)*, vol. 32, no. 4, pp. 458–482, April 2013.

[4] D. Park, Y. K. Kim, J. Rehg, and C. C. Kemp, "Learning to reach into the unknown: Selecting an initial condition when reaching in clutter," May 2014, submitted for publication.

[5] T. Bhattacharjee, A. Kapusta, J. M. Rehg, and C. C. Kemp, "Rapid categorization of object properties from incidental contact with a tactile sensing robot arm," in *Humanoid Robots, IEEE-RAS International Conference on*, Atlanta, GA, USA, Oct. 2013.

[6] L. R. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," in *Readings in Speech Recognition*, A. Waibel and K. F. Lee, Eds., Kaufmann, San Mateo, CA, 1990, pp. 267–296.

[7] W. Chai and B. Vercoe, “Folk music classification using hidden markov models,” in *Proceedings of International Conference on Artificial Intelligence*, vol. 6, no. 6.4. Citeseer, 2001.

[8] R. B. Rusu and S. Cousins, “3D is here: Point Cloud Library (PCL),” in *IEEE International Conference on Robotics and Automation (ICRA)*, Shanghai, China, May 9-13 2011.

[9] R. Diankov and J. Kuffner, “Openrave: A planning architecture for autonomous robotics,” Tech. Rep., 2008.

[10] J. Kuffner and S. LaValle, “Rrt-connect: An efficient approach to single-query path planning,” in *Robotics and Automation, 2000. Proceedings. ICRA ’00. IEEE International Conference on*, vol. 2, 2000, pp. 995–1001 vol.2.

[11] I. A. Şucan, M. Moll, and L. E. Kavraki, “The Open Motion Planning Library,” *IEEE Robotics & Automation Magazine*, vol. 19, no. 4, pp. 72–82, December 2012, http://ompl.kavrakilab.org.

[12] A. Bhattacharjee, T. Jain, S. Vaish, M. D. Killpack, and C. C. Kemp, “Tactile sensing over articulated joints with stretchable sensors,” in *IEEE World Haptics Conference (WHC), The 5th Joint EuroHaptics Conference and IEEE Haptics Symposium*, April 2013.

[13] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, “Ros: an open-source robot operating system,” in *ICRA workshop on open source software*, vol. 3, no. 3.2, 2009.

[14] J. Mattingley and S. Boyd, “Cvxgen: a code generator for embedded convex optimization,” *Optimization and Engineering*, vol. 13, no. 1, pp. 1–27, 2012. [Online]. Available: http://dx.doi.org/10.1007/s11081-011-9176-9

Appendix

1 Contents

- Presentation at International Symposium on Robotics Research (ISRR) 2013

Manipulation in Clutter with Whole-Arm Tactile Sensing

Charlie Kemp, PhD

Associate Professor, Department of Biomedical Engineering

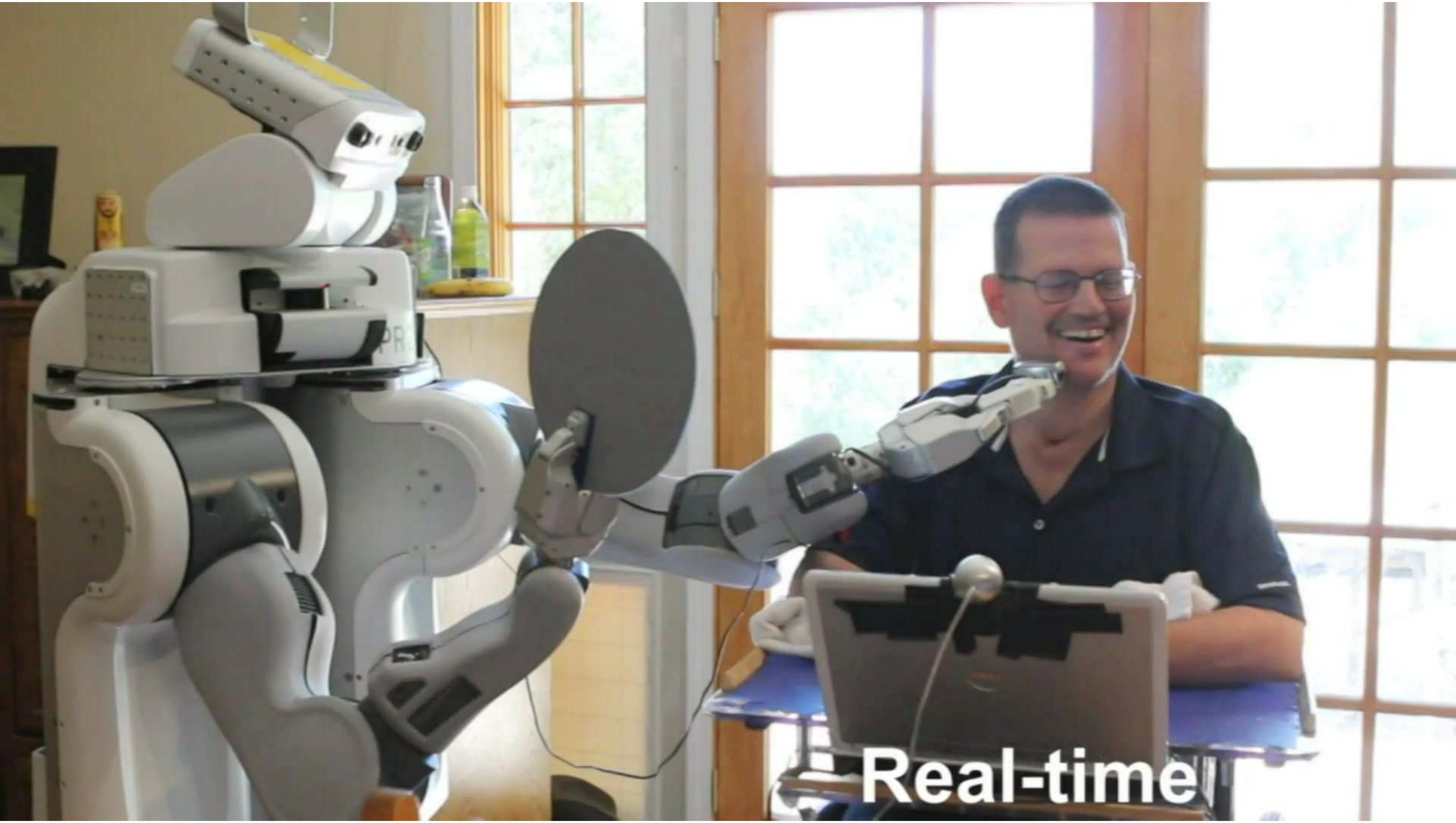
Adjunct, School of Interactive Computing

Adjunct, School of Electrical and Computer Engineering

Georgia Institute of Technology

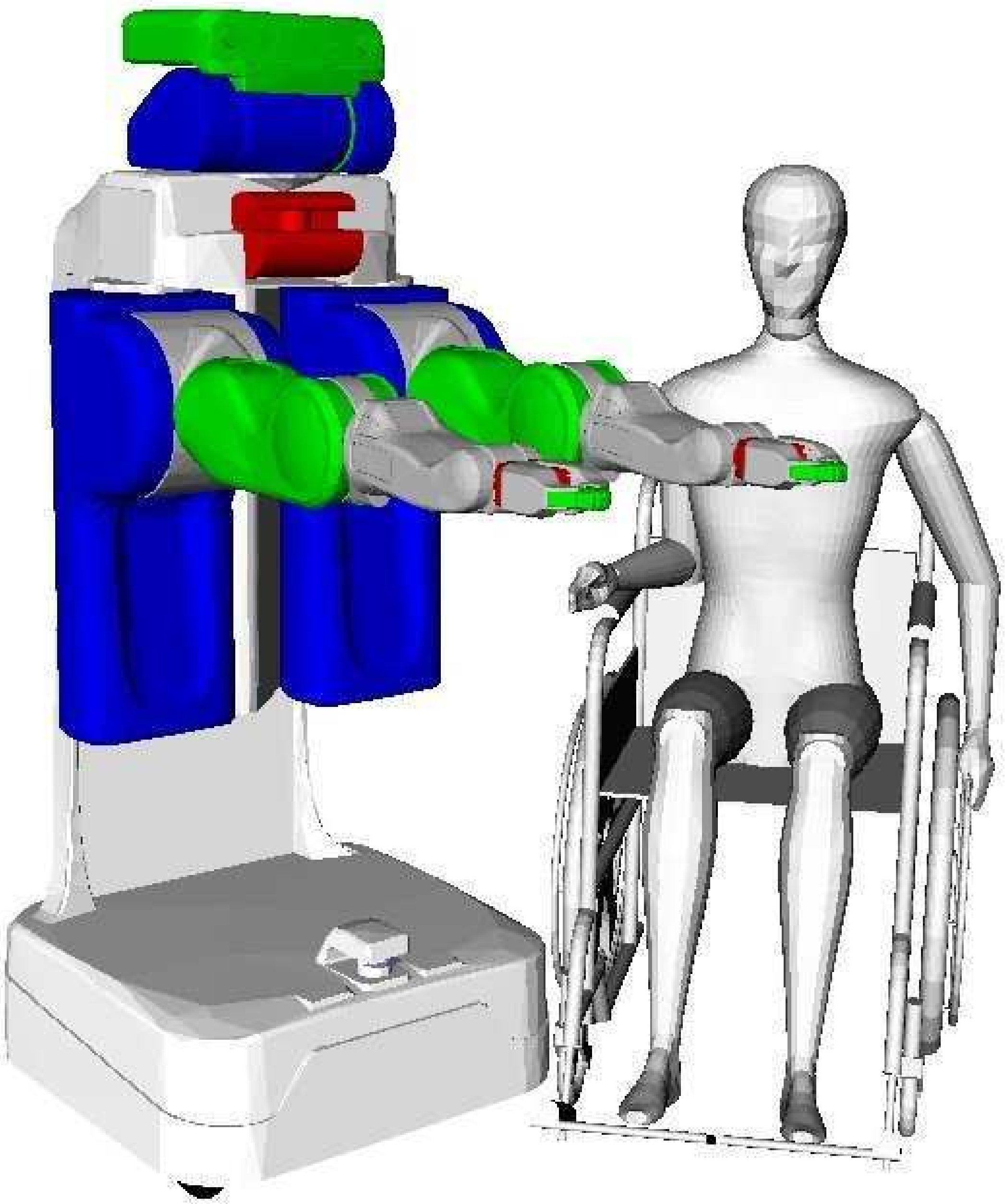
Robotics@GT
& Intelligent Machines

Healthcare
Robotics

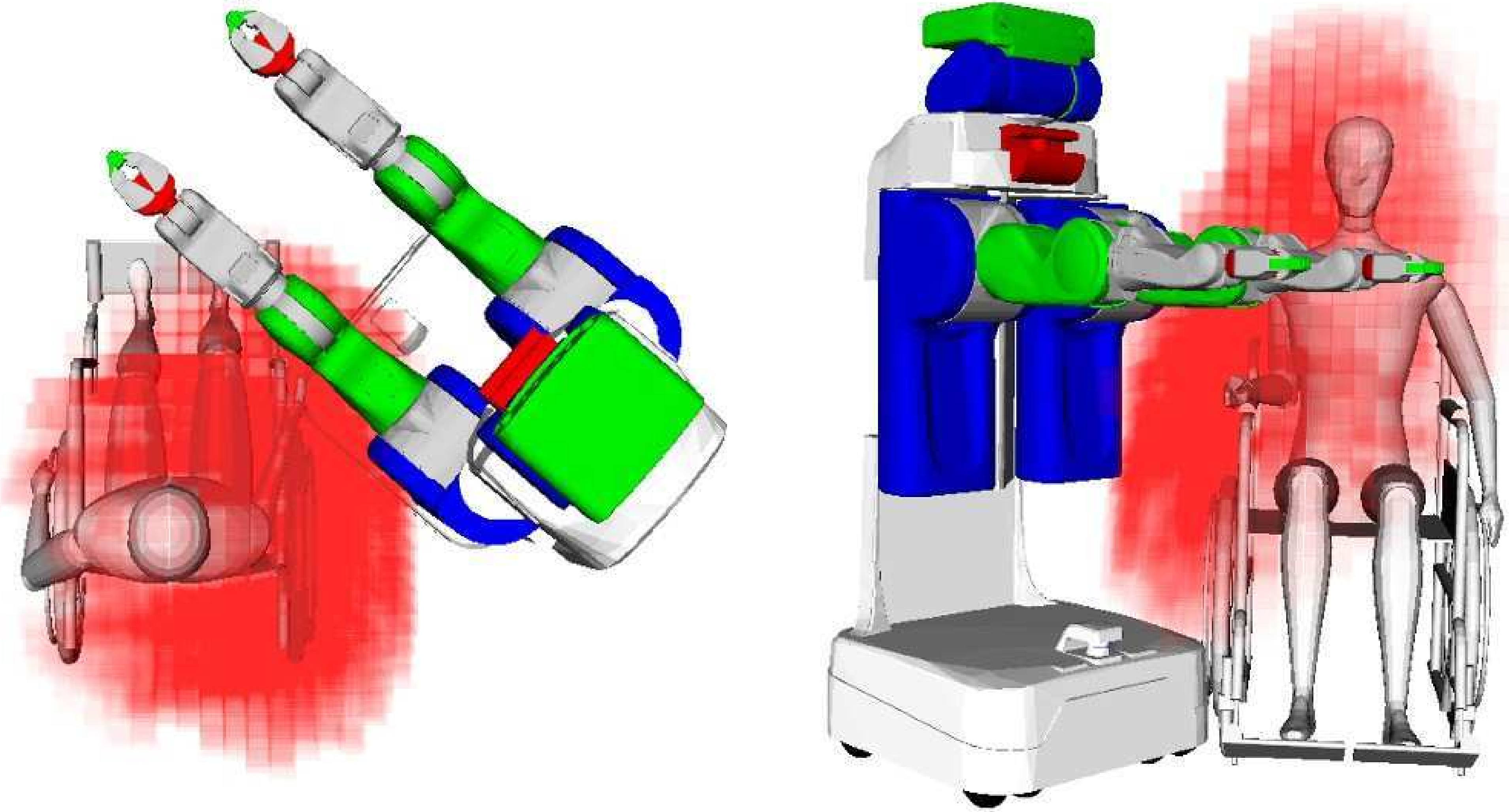


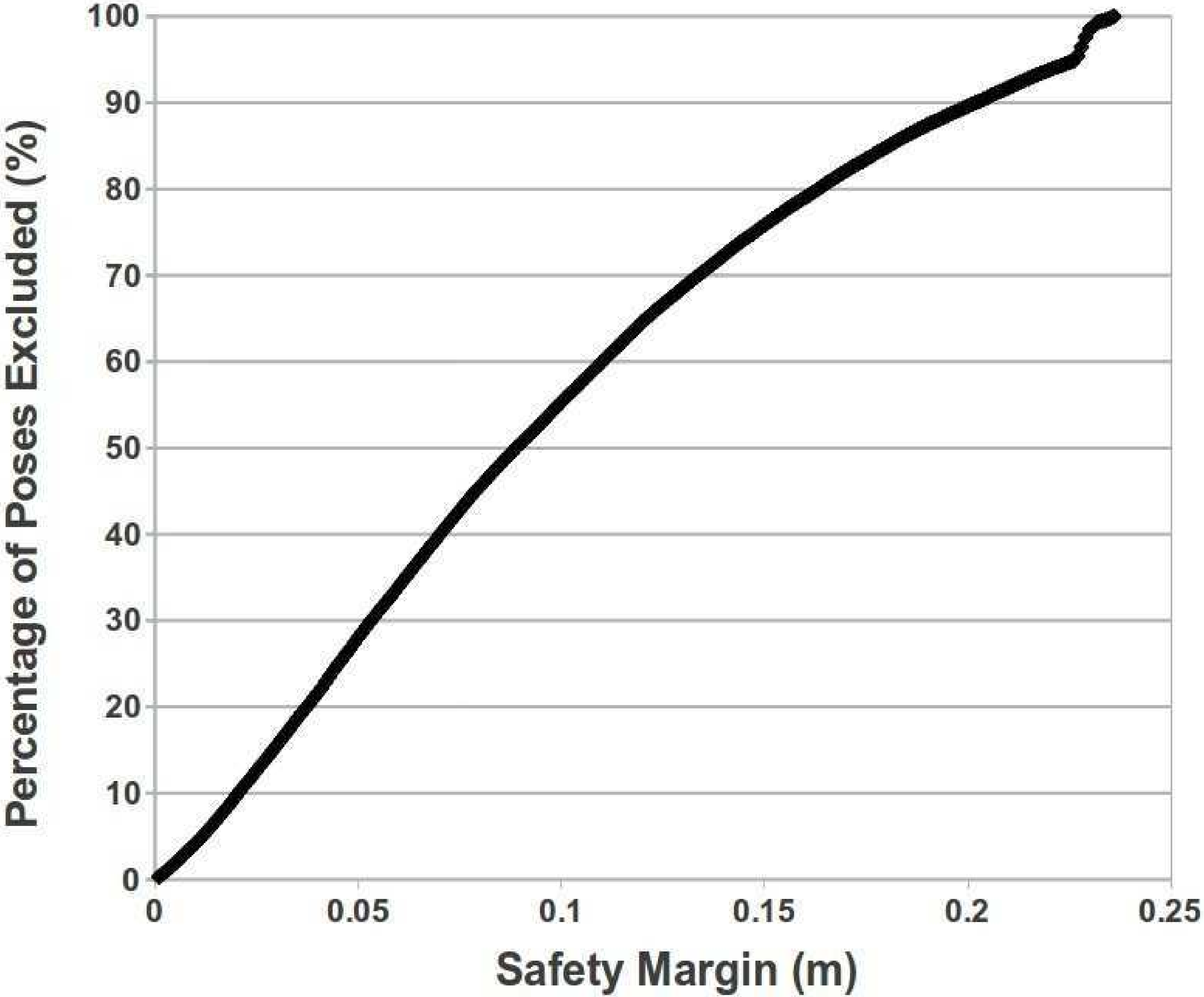
PR2
Real-time

Geometric Simulation to Investigate Value of Contact

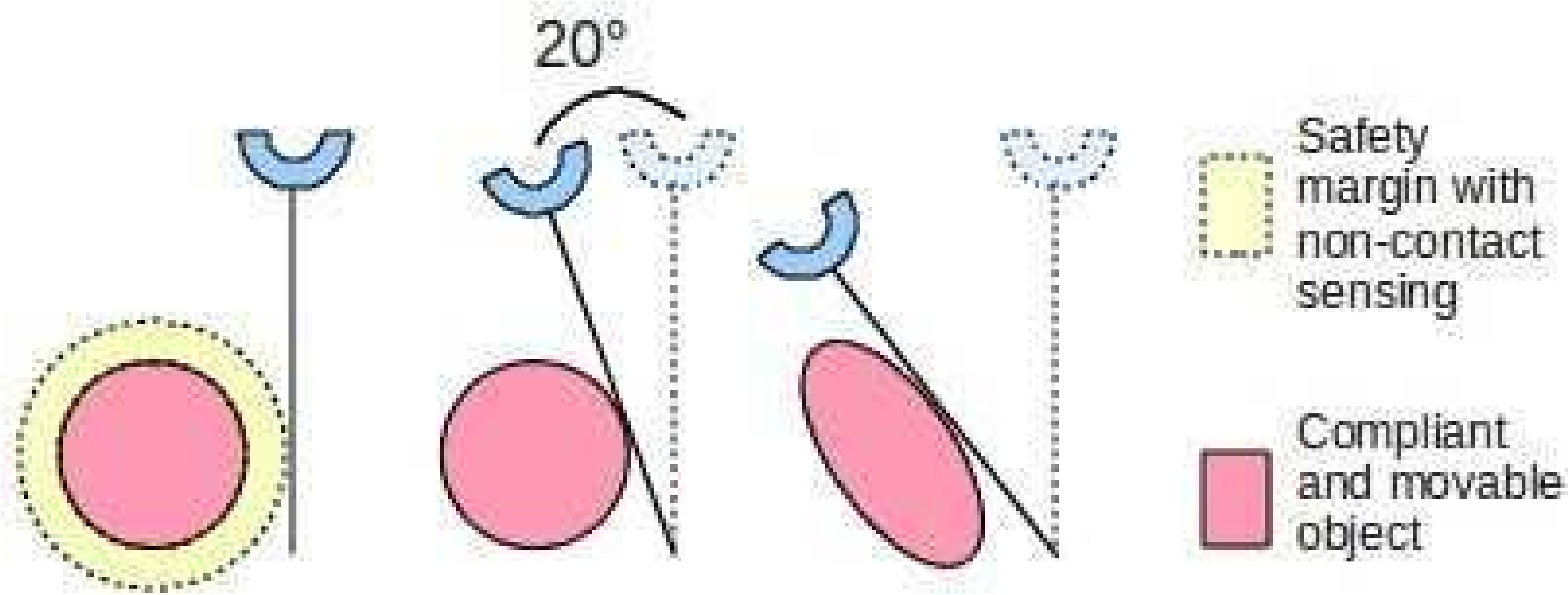


Unreachable End Effector Poses with Safety Margin of 4cm




Percentage of Poses Excluded (%)
100
90
80
70
60
50
40
30
20
10
0
0
0.05
0.1
0.15
0.2
0.25
Safety Margin (m)

Gaining Workspace Through Contact


20°
Safety margin with non-contact sensing
Compliant and movable object

No contact with safety margin

Light contact

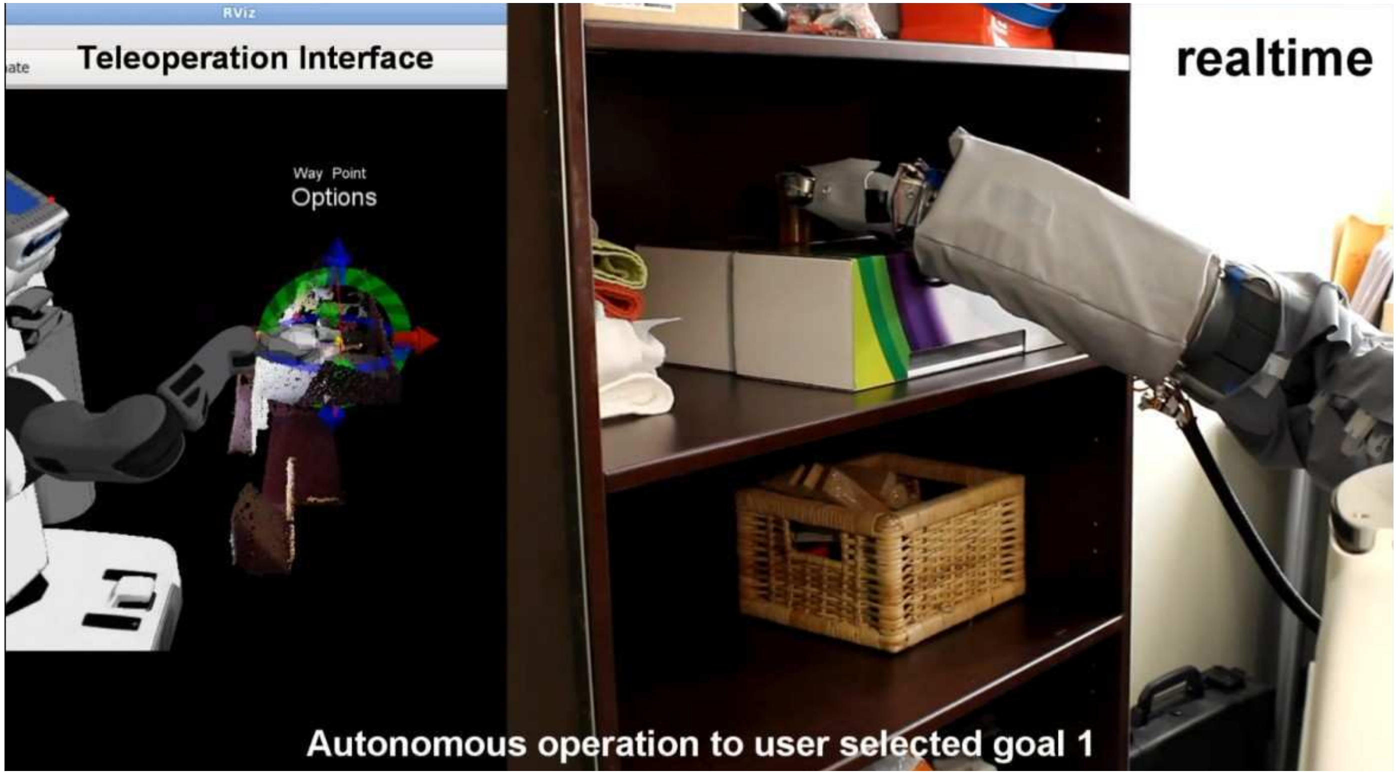
Harder contact Resulting in motion and deformation

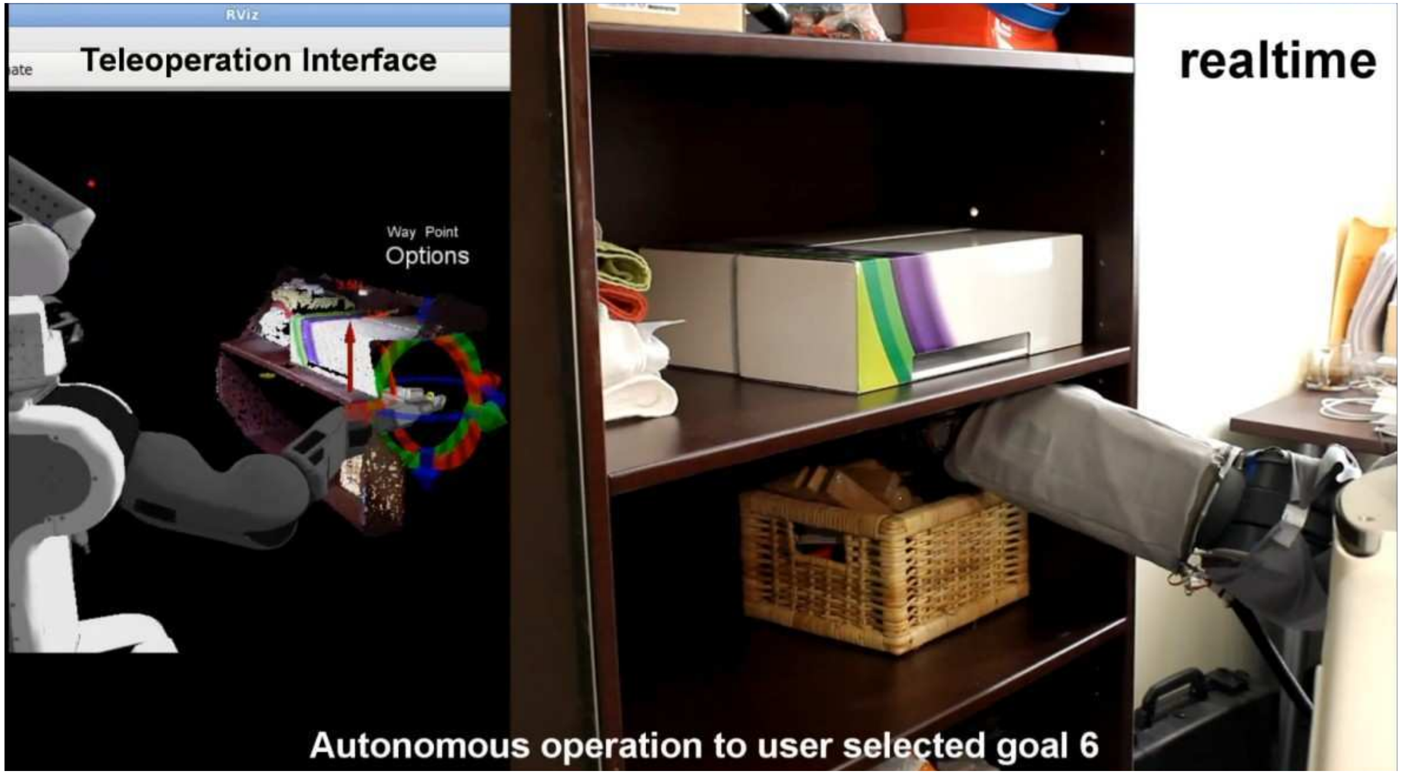
Types of Assistive Tasks

- **Activities of Daily Living (ADLs)**
  - Feeding, toileting, transferring, dressing, and hygiene
  - Manipulation near the user’s body
  - Predictive of ability to live independently
- **Instrumental Activities of Daily Living (IADLs)**
  - Housework, food preparation, shopping, …
  - Manipulation of objects in the environment.
- **Enhanced Activities of Daily Living (EADLs)**
  - Hobbies, social activities, …
  - Manipulation plays many roles

Types of Assistive Tasks

- **Activities of Daily Living (ADLs)**
  - Feeding, toileting, transferring, dressing, and hygiene
  - **Manipulation near the user’s body**
  - Predictive of ability to live independently
- **Instrumental Activities of Daily Living (IADLs)**
  - Housework, food preparation, shopping, …
  - **Manipulation of objects in the environment.**
- **Enhanced Activities of Daily Living (EADLs)**
  - Hobbies, social activities, …
  - Manipulation plays many roles


RViz
ate
Teleoperation Interface
realtime
Way Point
Options
Autonomous operation to user selected goal 1


RViz
ate
Teleoperation Interface
realtime
Way Point
Options
Autonomous operation to user selected goal 6

Dominant Strategy for Robotic Manipulation has been to **Avoid Contact**

- Between the robot’s arm and **the world**
- Between the robot’s arm and **other parts of its body**
- Between the robot’s arm and **people**

Contact with the World is Common

Reaching a high shelf

Tying a rope to a pole

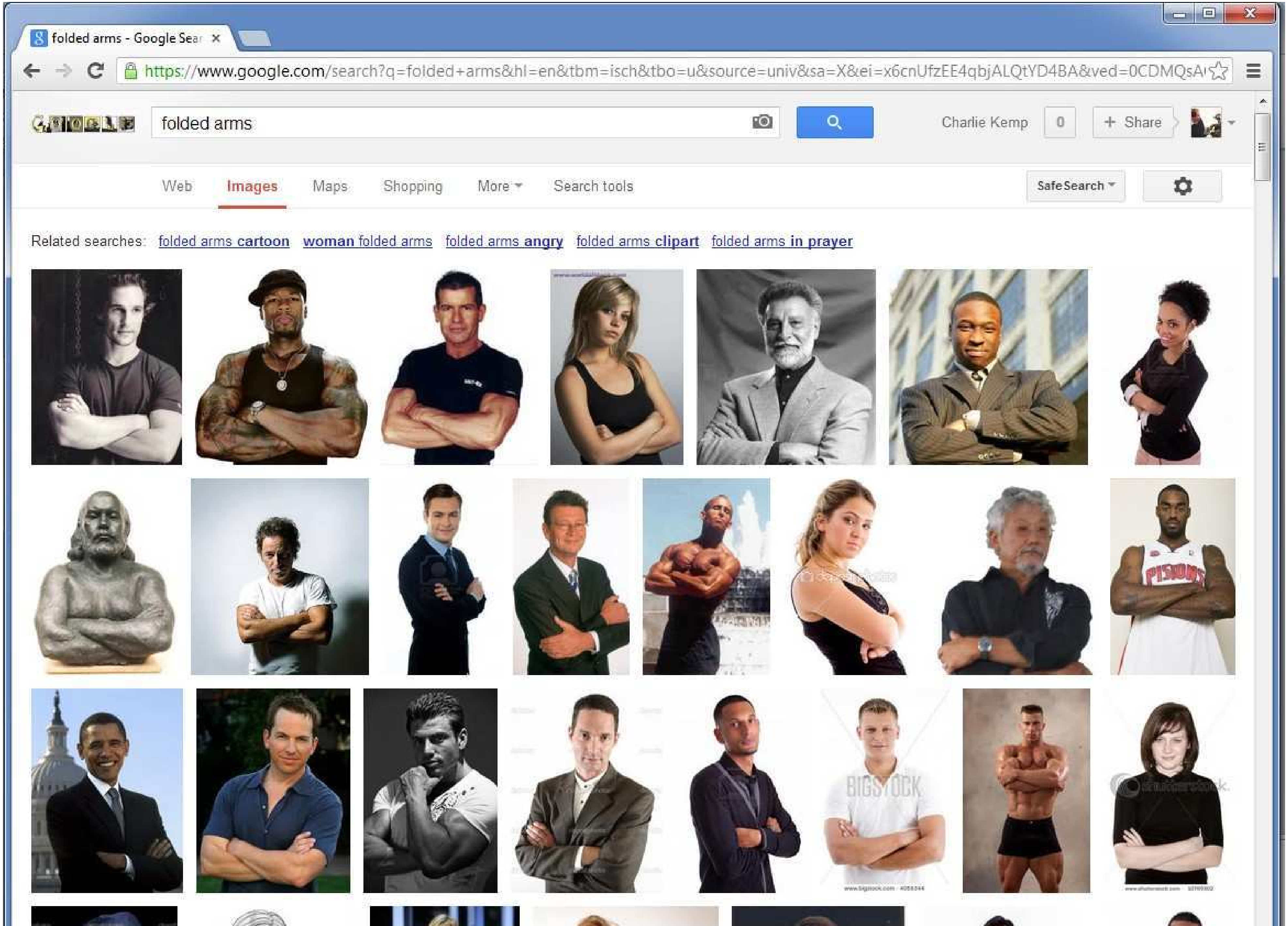
Installing a car seat

Cleaning a car trunk

Plumbing

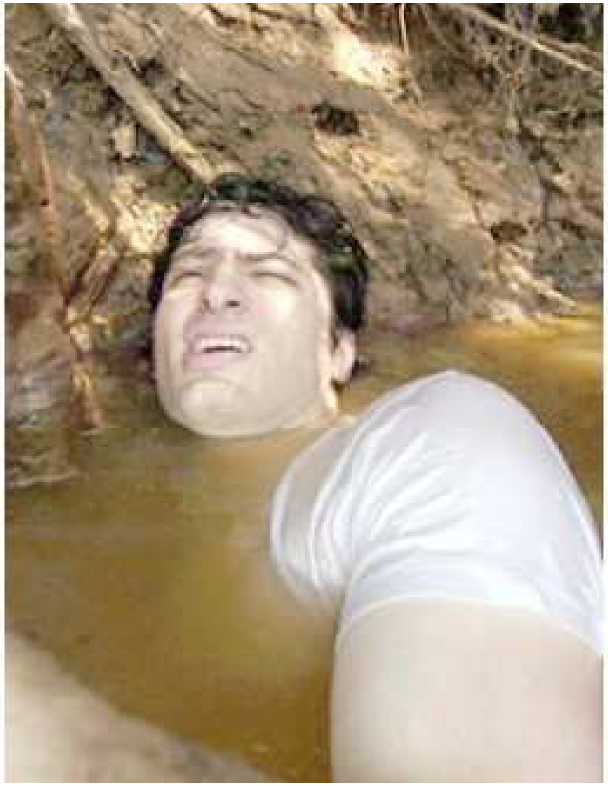
Carrying boxes

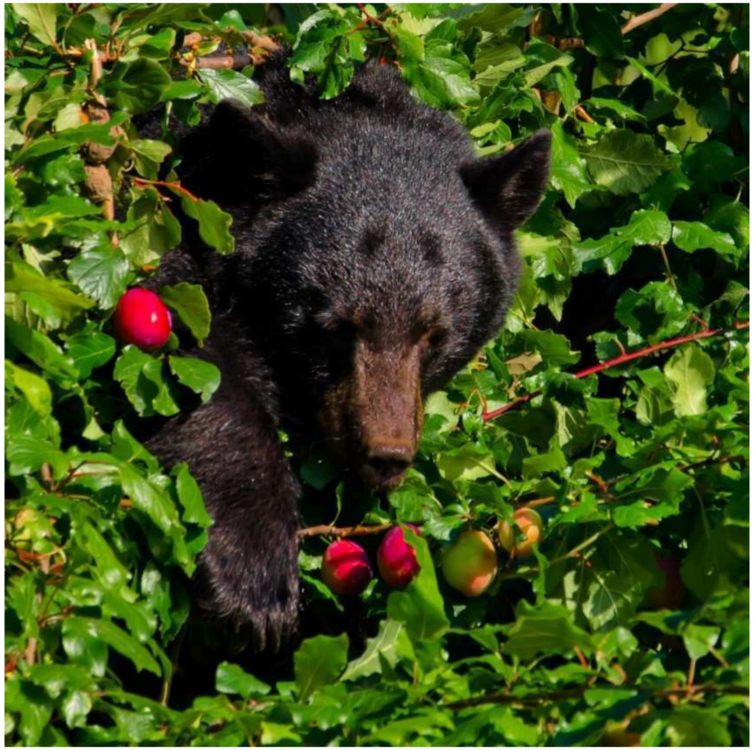
Self-contact is Common


folded arms - Google Sear ×
https://www.google.com/search?q=folded+arms&hl=en&tbm=isch&tbo=u&source=univ&sa=X&ei=x6cnUfzEE4qbjALQtYD4BA&ved=0CDMQsA
folded arms
Charlie Kemp
0
+ Share
Web
Images
Maps
Shopping
More
Search tools
SafeSearch
Related searches: folded arms cartoon woman folded arms folded arms angry folded arms clipart folded arms in prayer
PISTONS
BIGSTOCK

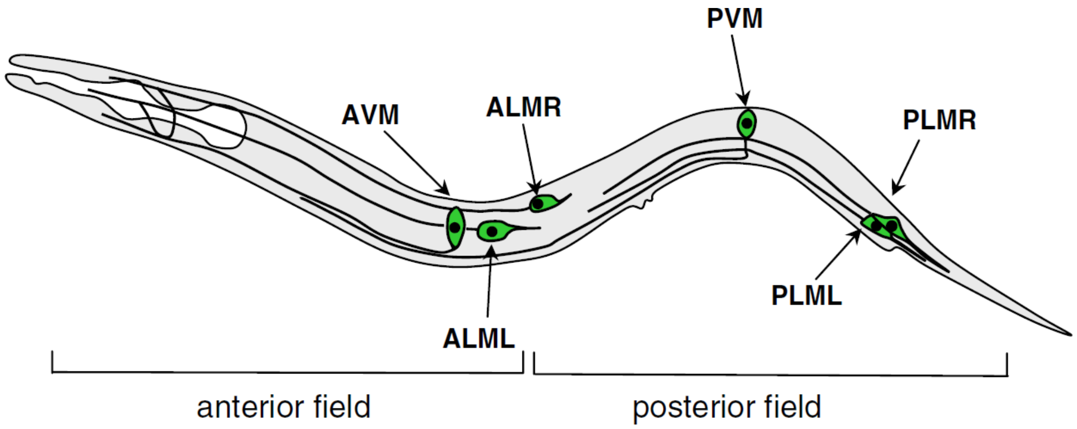
Contact with People is Common

(e.g., when providing assistance)

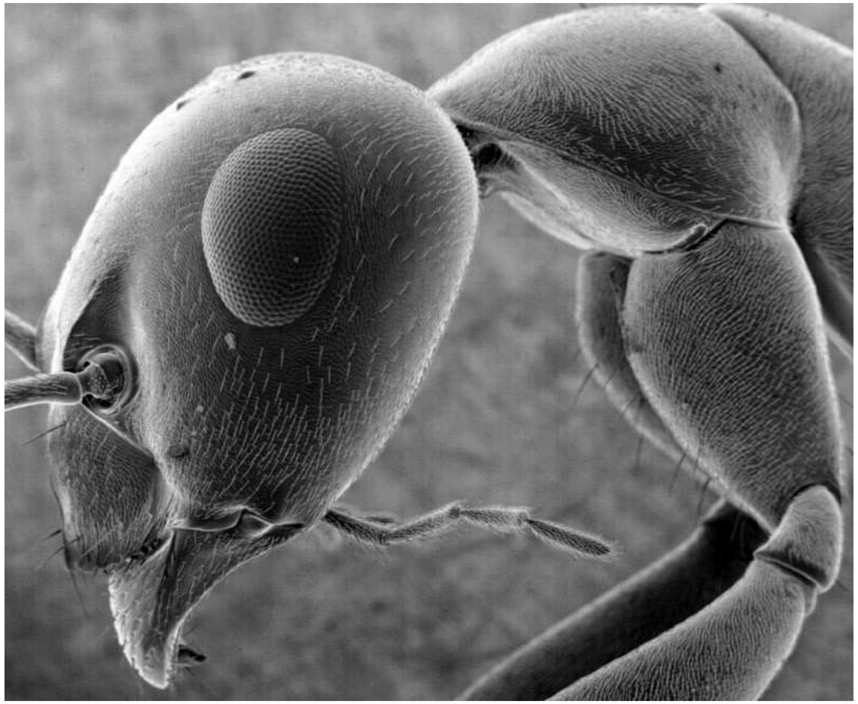




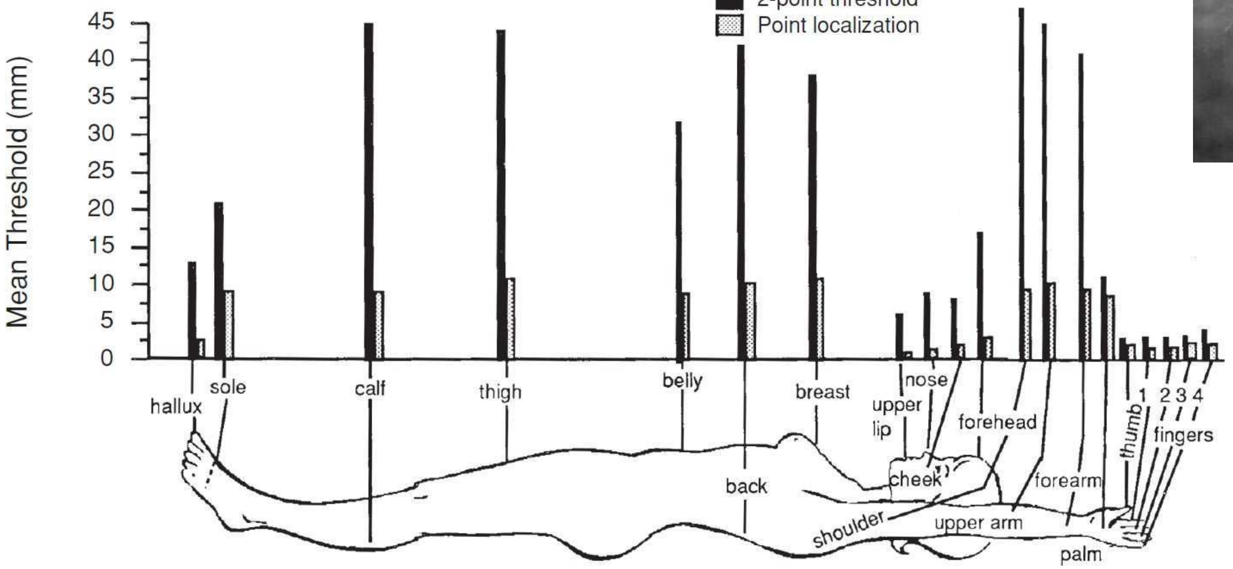
Whole-body tactile sensing is everywhere.



Nematode (~mm)



Ant (~cm)



Human (~m)

Controllers that Allow Contact

- Assume
  - Low contact forces have no associated penalty
  - The robot has
  - Low-stiffness compliant joints
  - Whole-arm tactile sensing

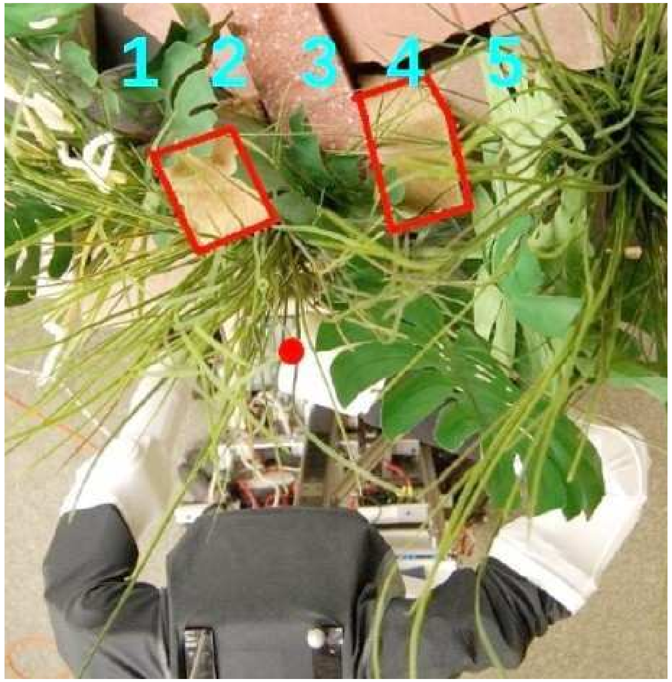
Problem: Reach a Target in Extreme Clutter

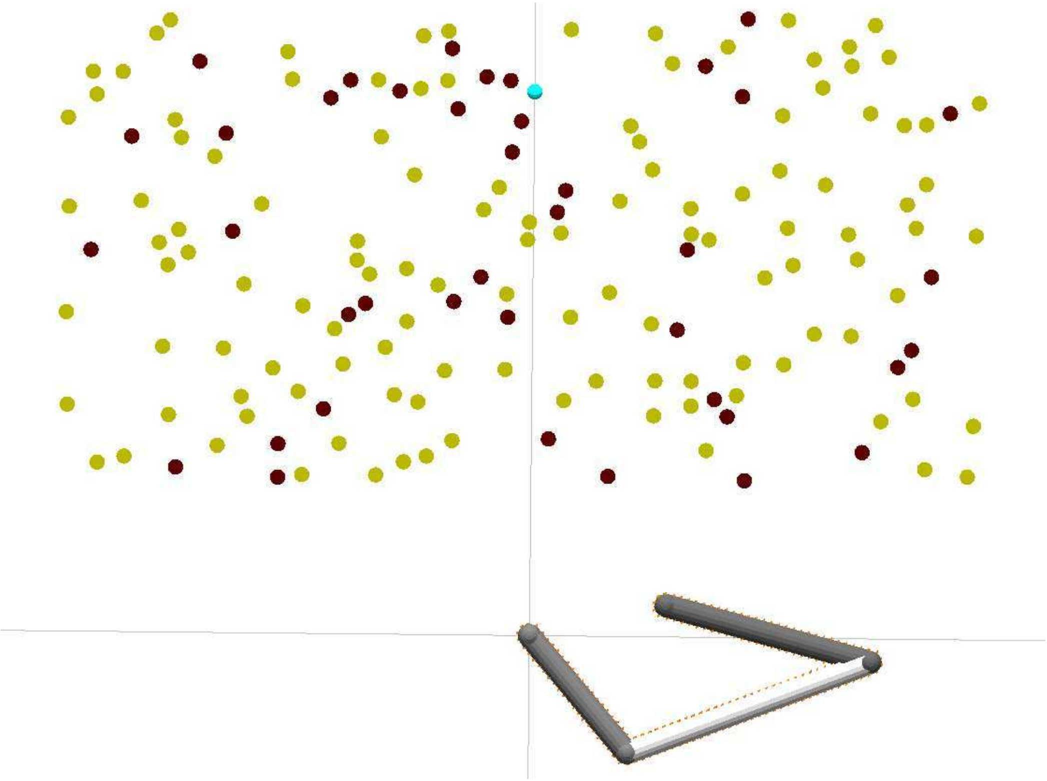
- Clutter
  - *"Clutter refers to everything that might limit access to the object."* - *Generality and Simple Hands by* Matthew T. Mason, Siddhartha Srinivasa, and Andres S. Vazquez, ISRR 2009.
  - To clutter a place is *"to fill or cover with scattered or disordered things that impede movement or reduce effectiveness"* – *Merriam-Webster*

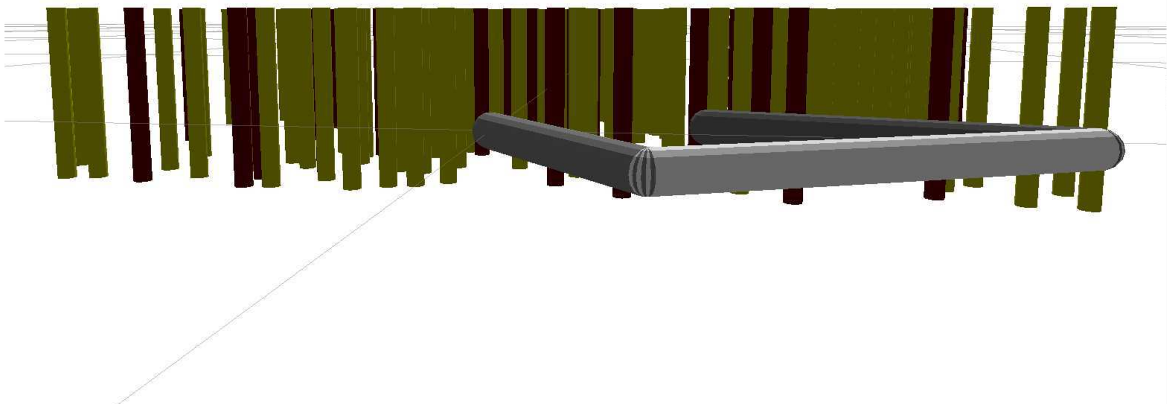
- Extreme clutter
  - **Physical challenge:** All solutions require contact with parts of the environment other than the target.
  - **Perceptual challenges:** Line of sight to the target is completely occluded and inferring how the environment will respond to applied forces requires contact.
  - **Challenge due to disorder:** No detailed model of the environment is available prior to encountering the scene.

Methodology: Optimize Empirical Performance

(success rate and contact forces in simulated clutter)

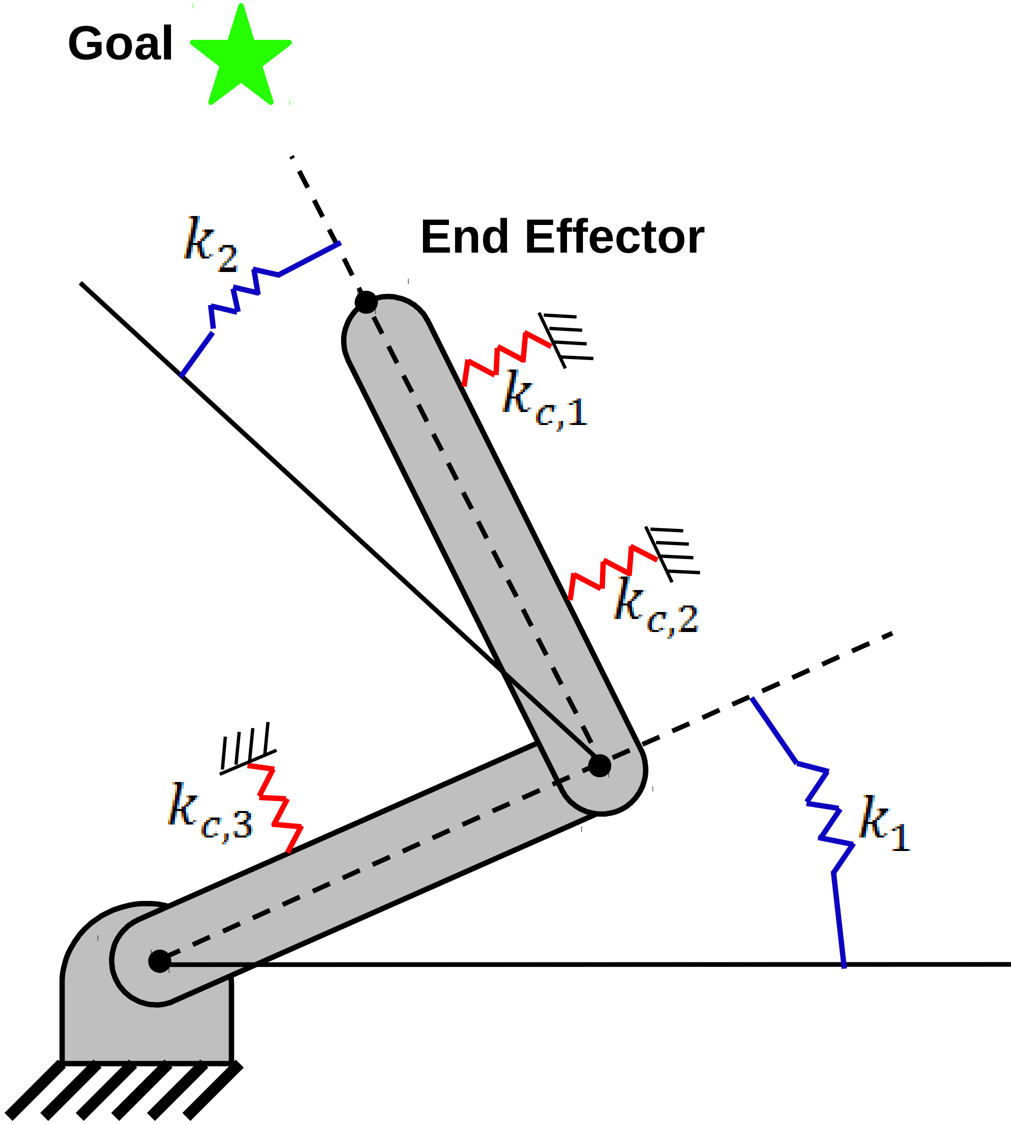




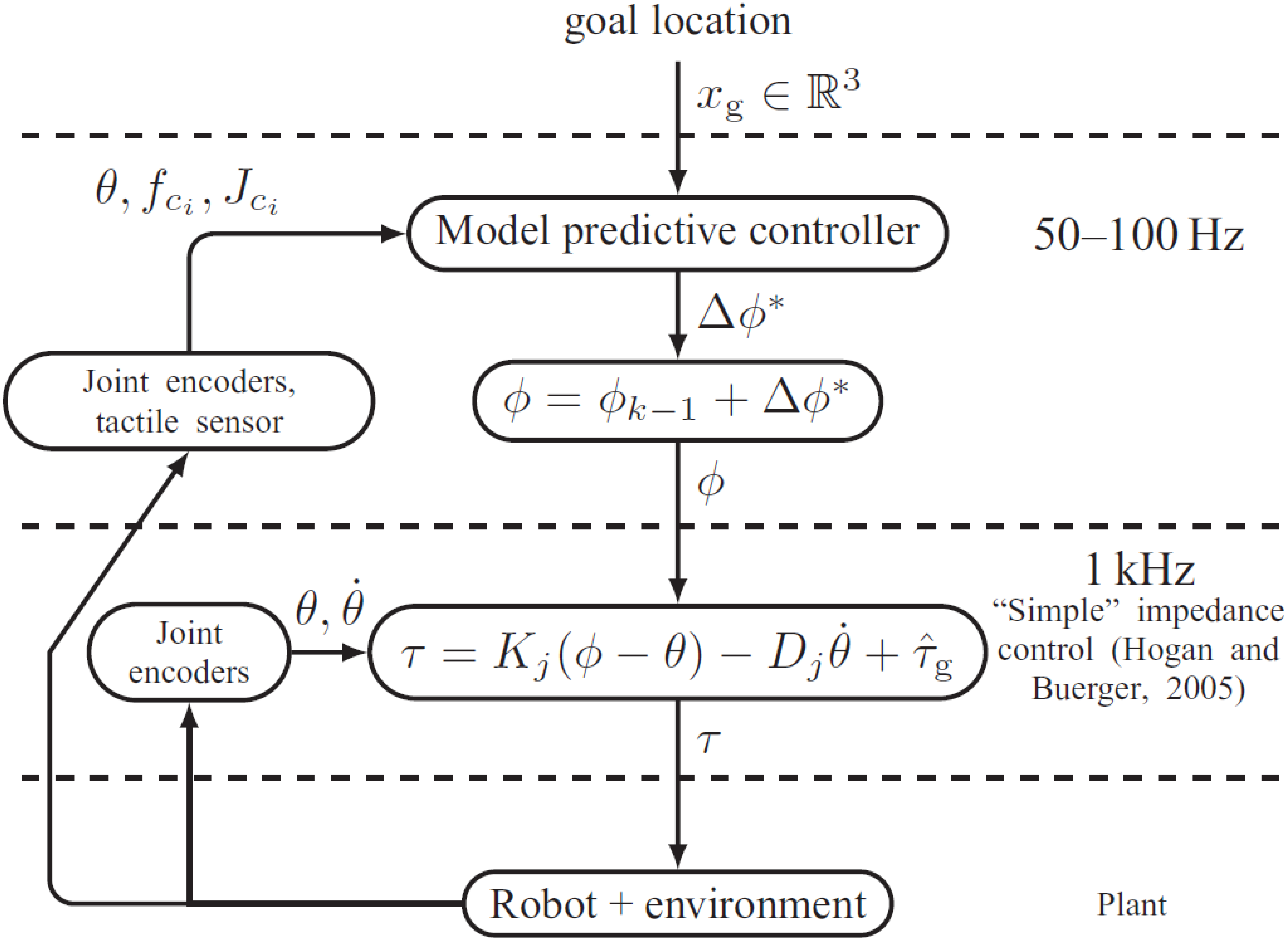


Quasi-static Model Predictive Control

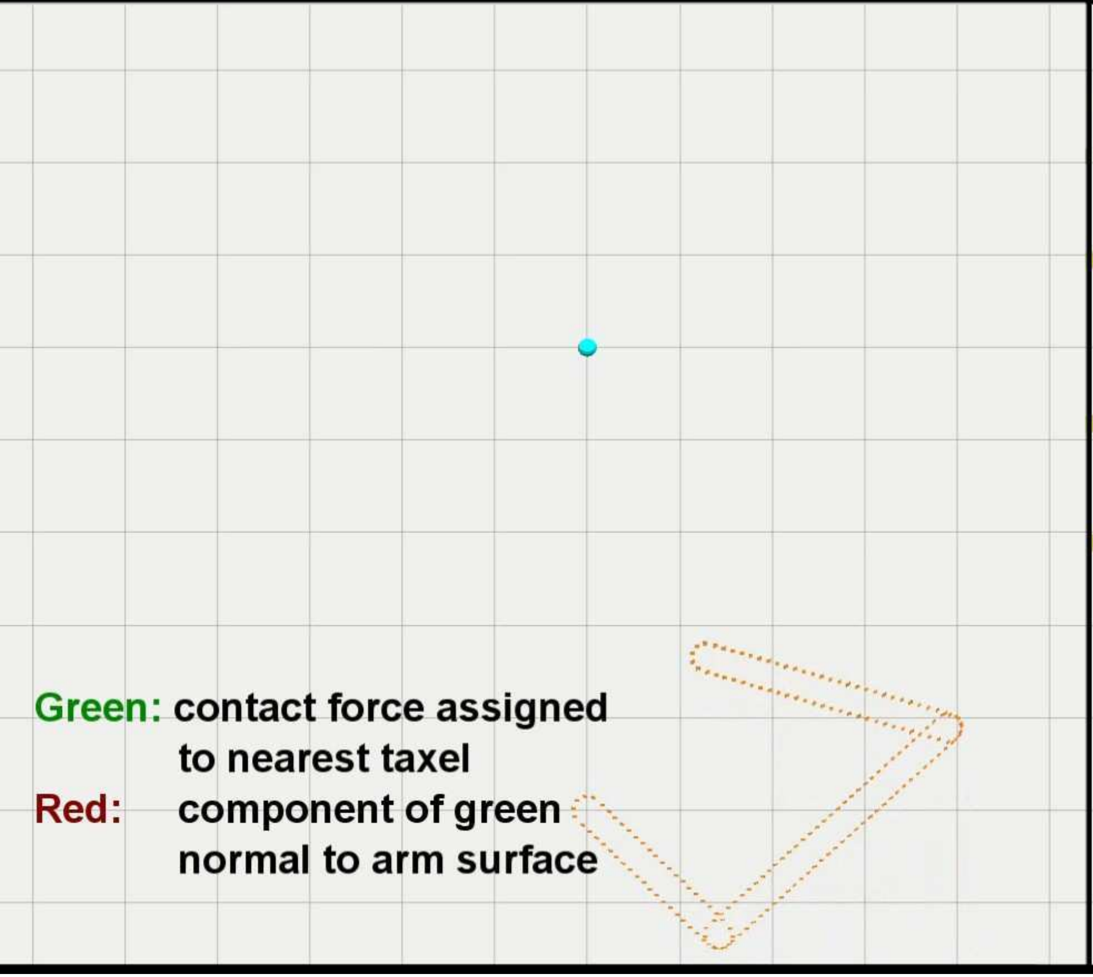
- At each time step
  - Generates a linear quasi-static mechanical model based on tactile sensing
  - Uses quadratic programming to find a change to the equilibrium angles of the springs at the joints that
    - Minimizes the predicted distance from the hand to the goal
    - Subject to constraints on the predicted contact forces



Jain, Advait, et al. "Reaching in clutter with whole-arm tactile sensing." *The International Journal of Robotics Research* 32.4 (2013): 458-482.


goal location
$x_{\mathrm{g}} \in \mathbb{R}^3$
θ, f_{c_i}, J_{c_i}
Model predictive controller
50–100 Hz
$\Delta\phi^*$
Joint encoders, tactile sensor
$\phi = \phi_{k-1} + \Delta\phi^*$
ϕ
1 kHz
"Simple" impedance control (Hogan and Buerger, 2005)
$\theta, \dot{\theta}$
Joint encoders
$\tau = K_j(\phi - \theta) - D_j\dot{\theta} + \hat{\tau}_{\mathrm{g}}$
τ
Robot + environment
Plant

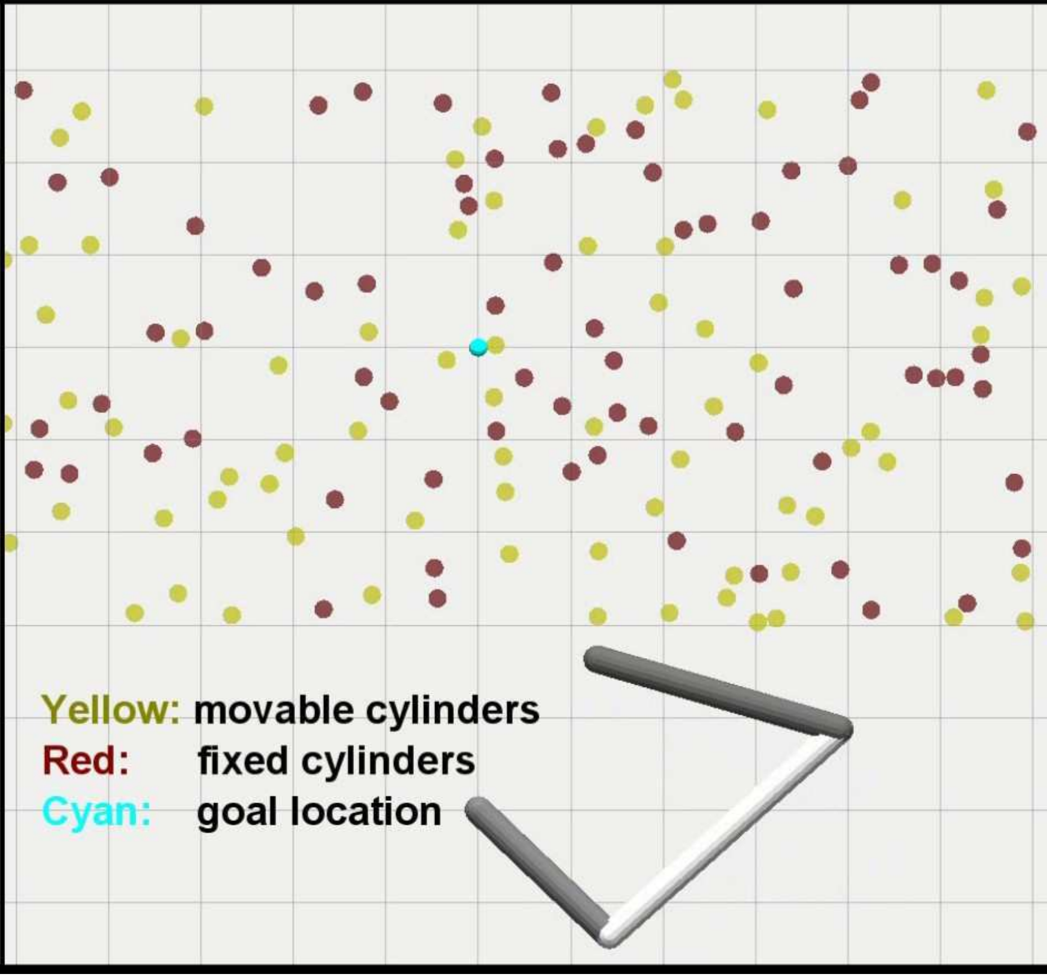
Whole-arm tactile sensing



Green: contact force assigned to nearest taxel

Red: component of green normal to arm surface

Simulated arm reaching in clutter

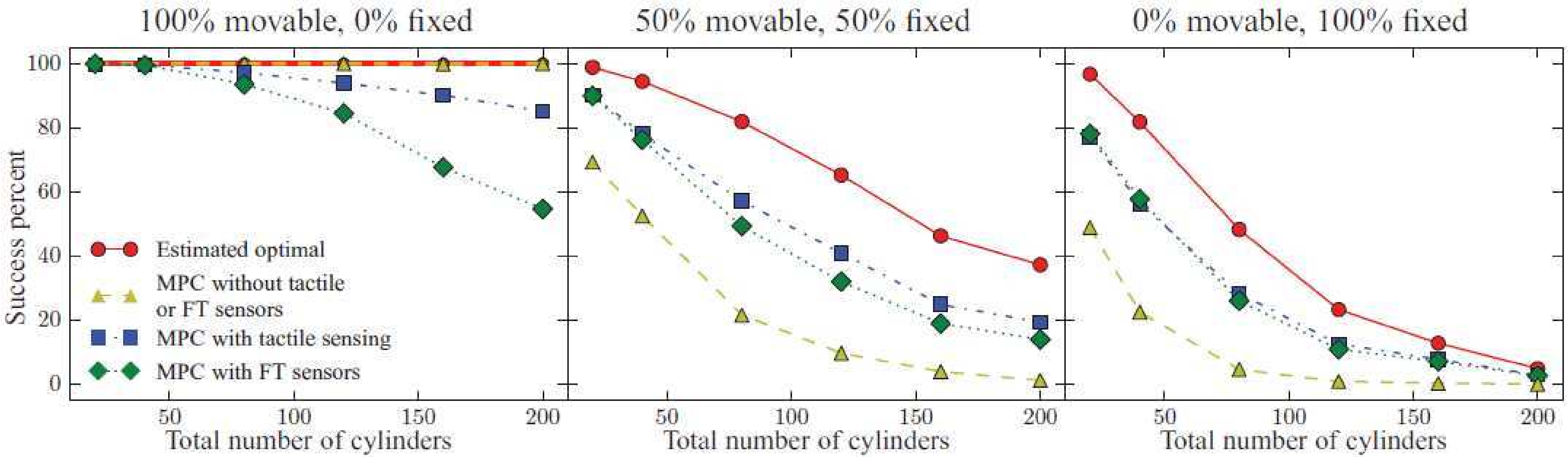


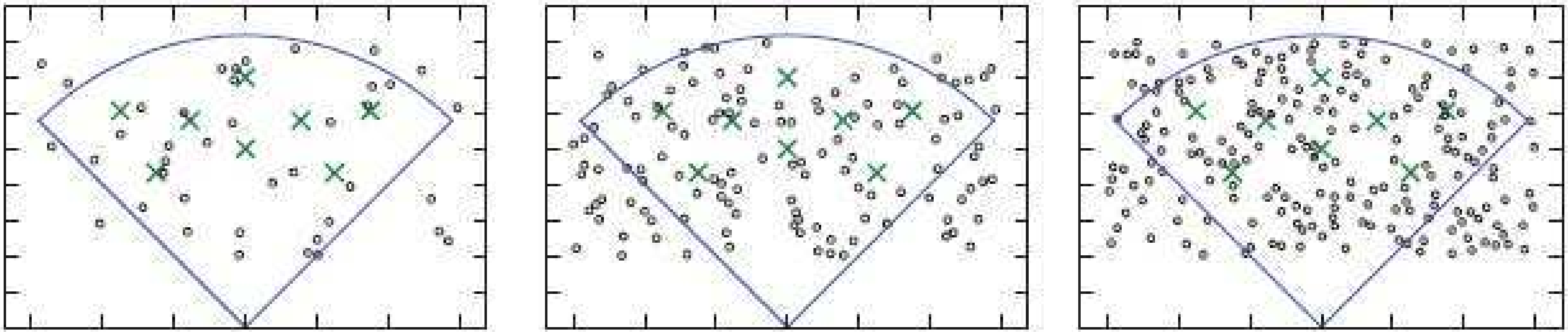
Yellow: movable cylinders

Red: fixed cylinders

Cyan: goal location

Single Reach Performance (64800 trials)

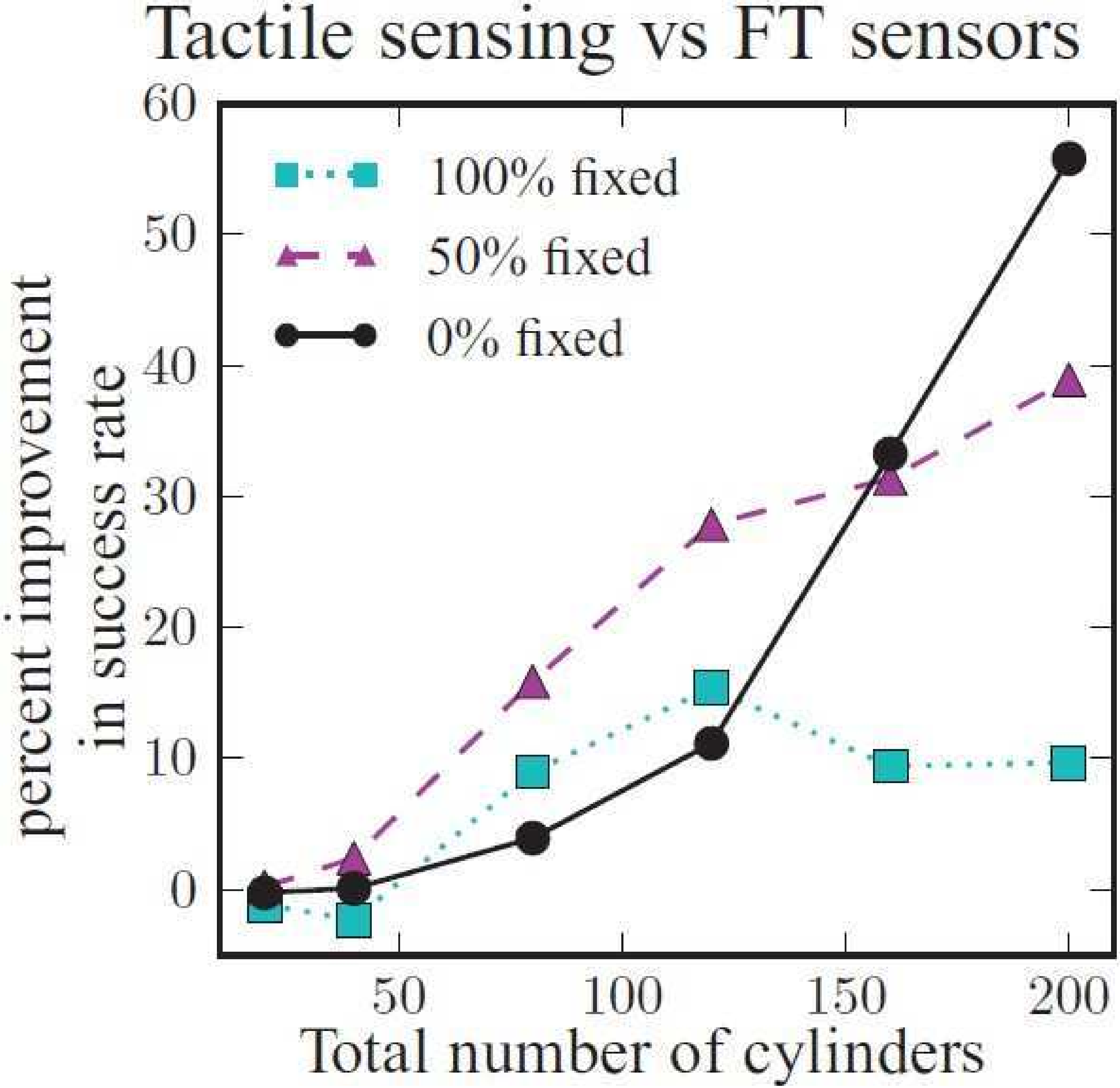


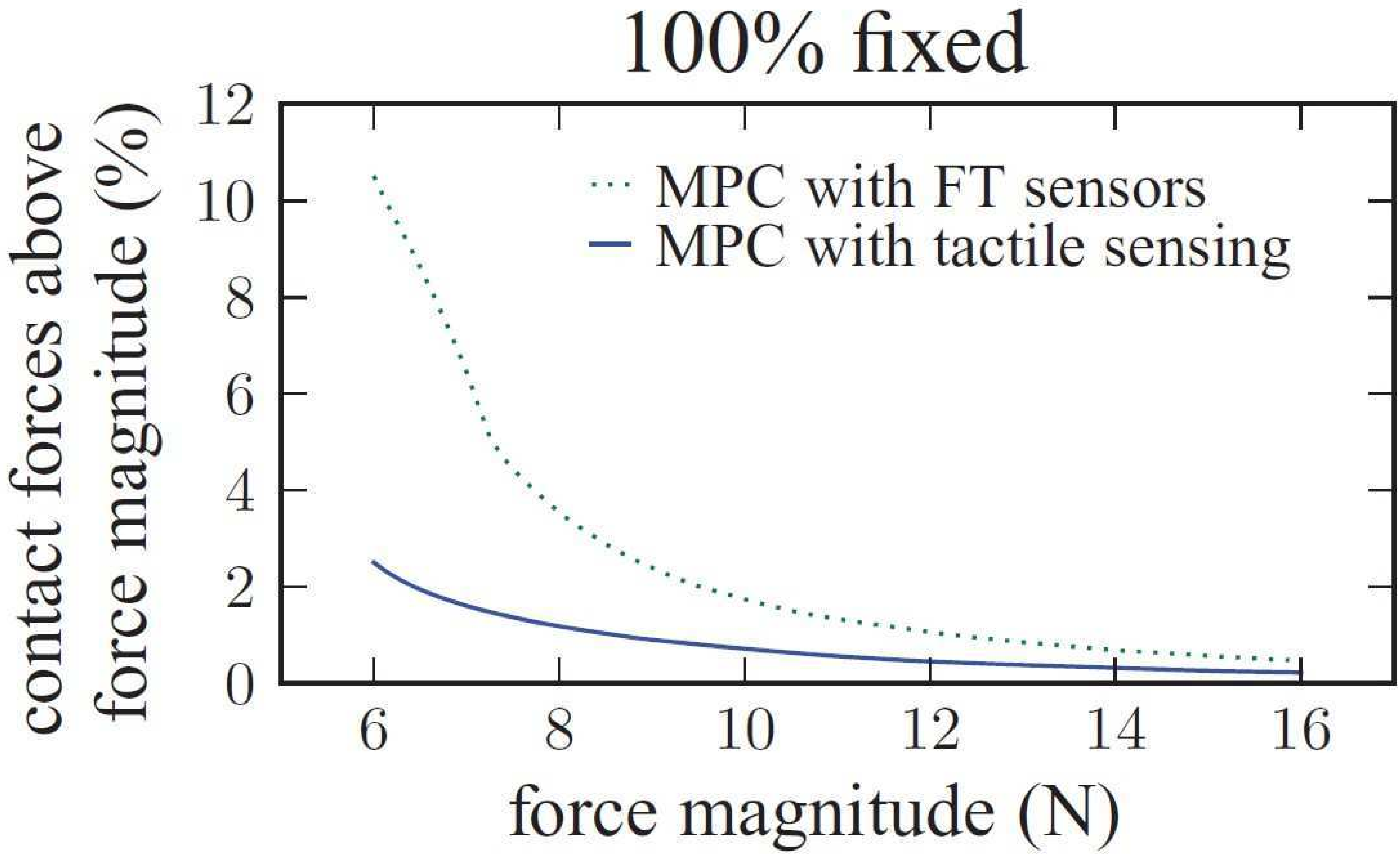


(b) 40 cylinders

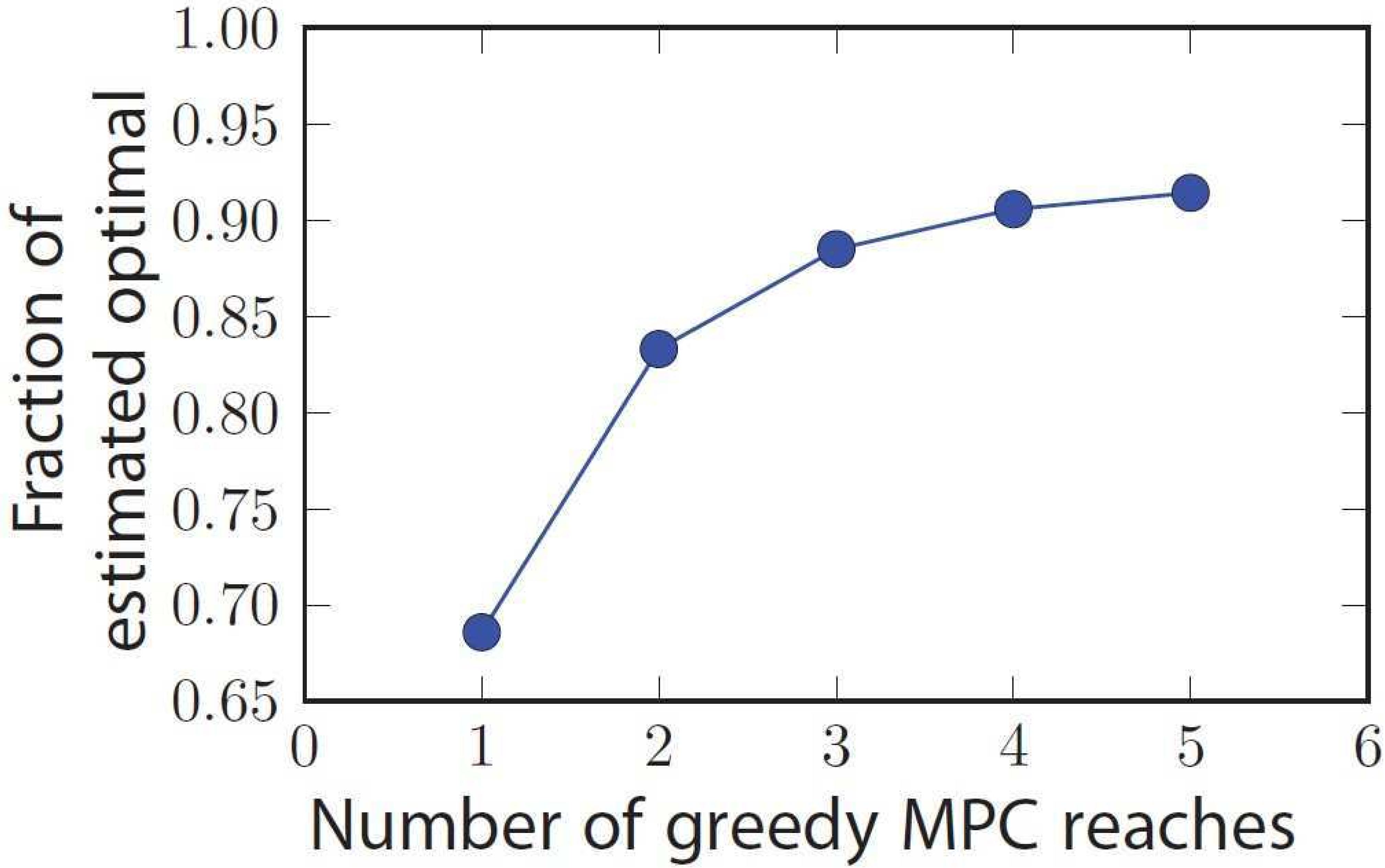
(d) 120 cylinders

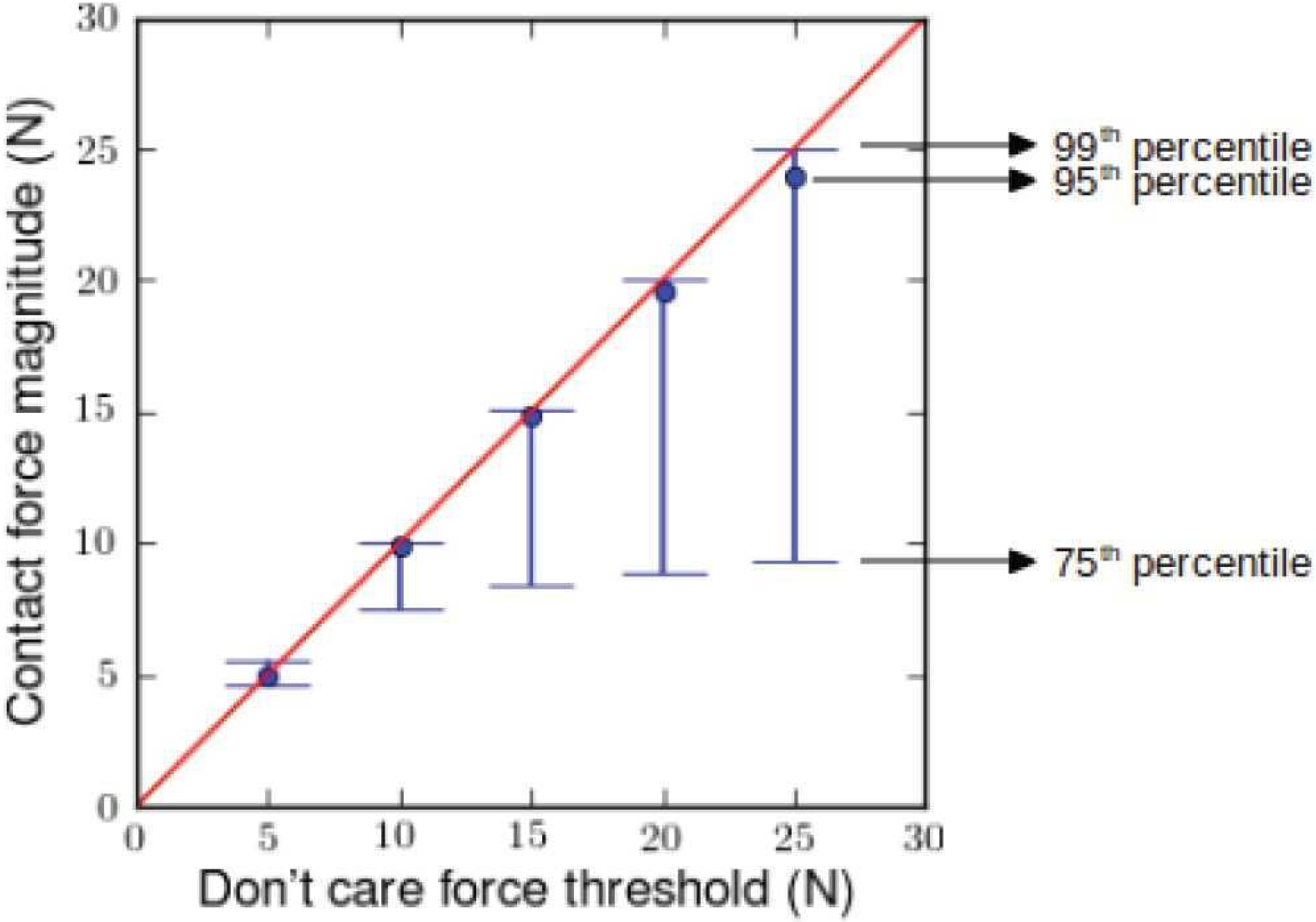
(f) 200 cylinders

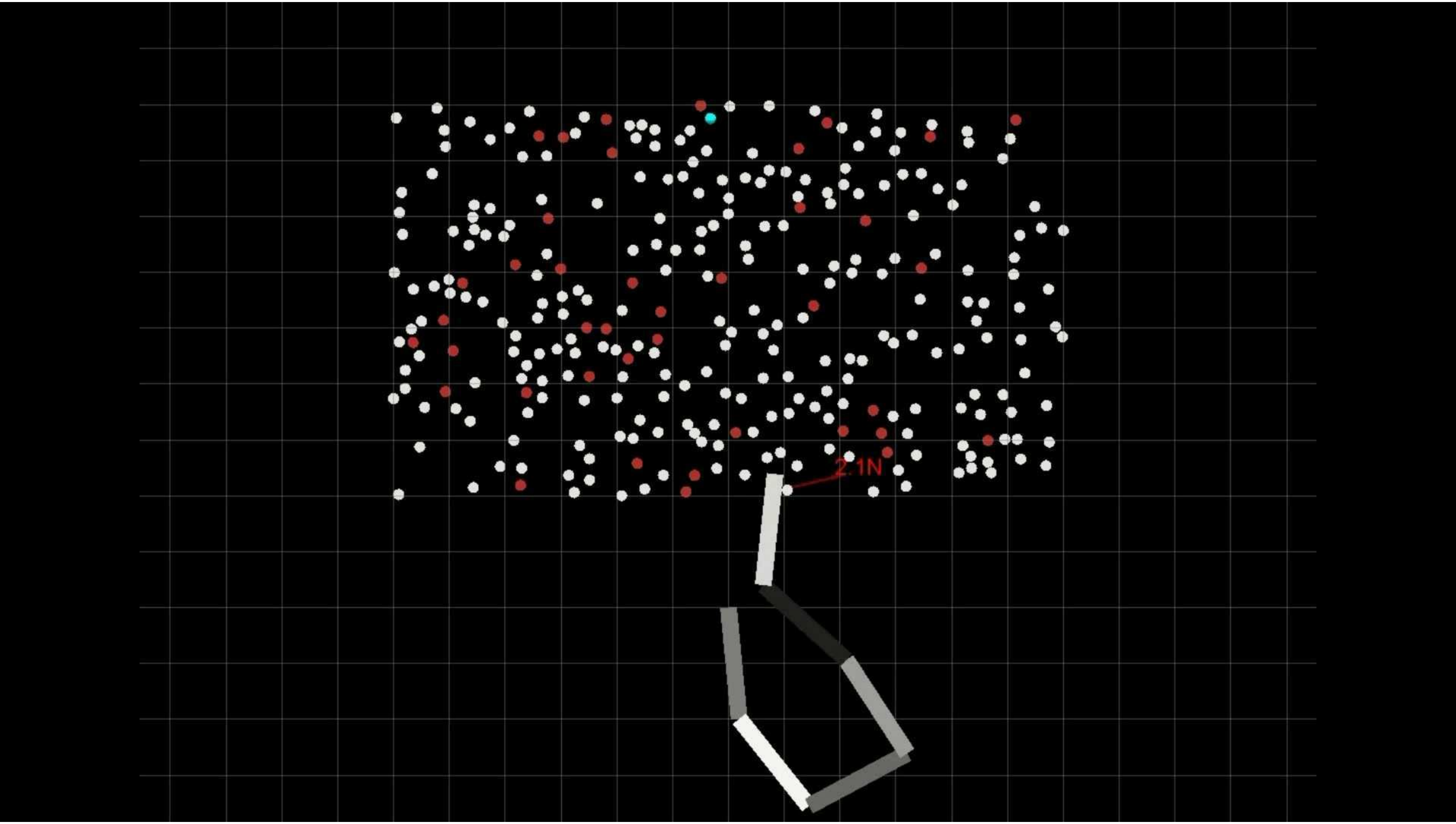

Tactile sensing vs FT sensors
100% fixed
50% fixed
0% fixed
percent improvement in success rate
0
10
20
30
40
50
60
50
100
150
200
Total number of cylinders


100% fixed
contact forces above force magnitude (%)
12
10
8
6
4
2
0
MPC with FT sensors
MPC with tactile sensing
6
8
10
12
14
16
force magnitude (N)

91.4% of Optimal with 5 Greedy Reaches

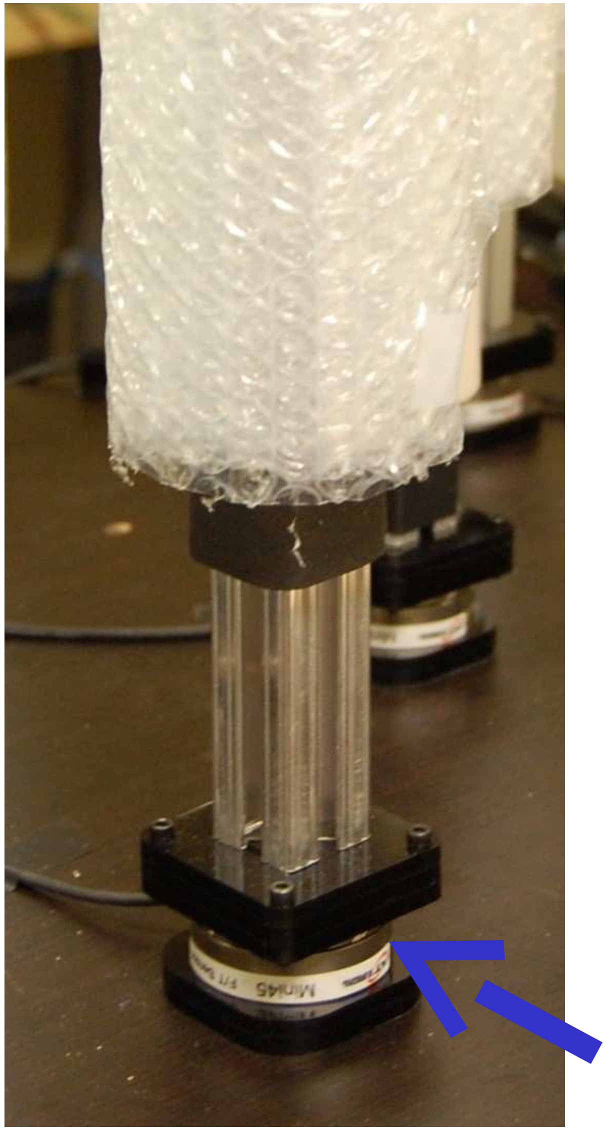



30
25
20
15
10
5
0
0
5
10
15
20
25
30
Contact force magnitude (N)
Don't care force threshold (N)
99th percentile
95th percentile
75th percentile

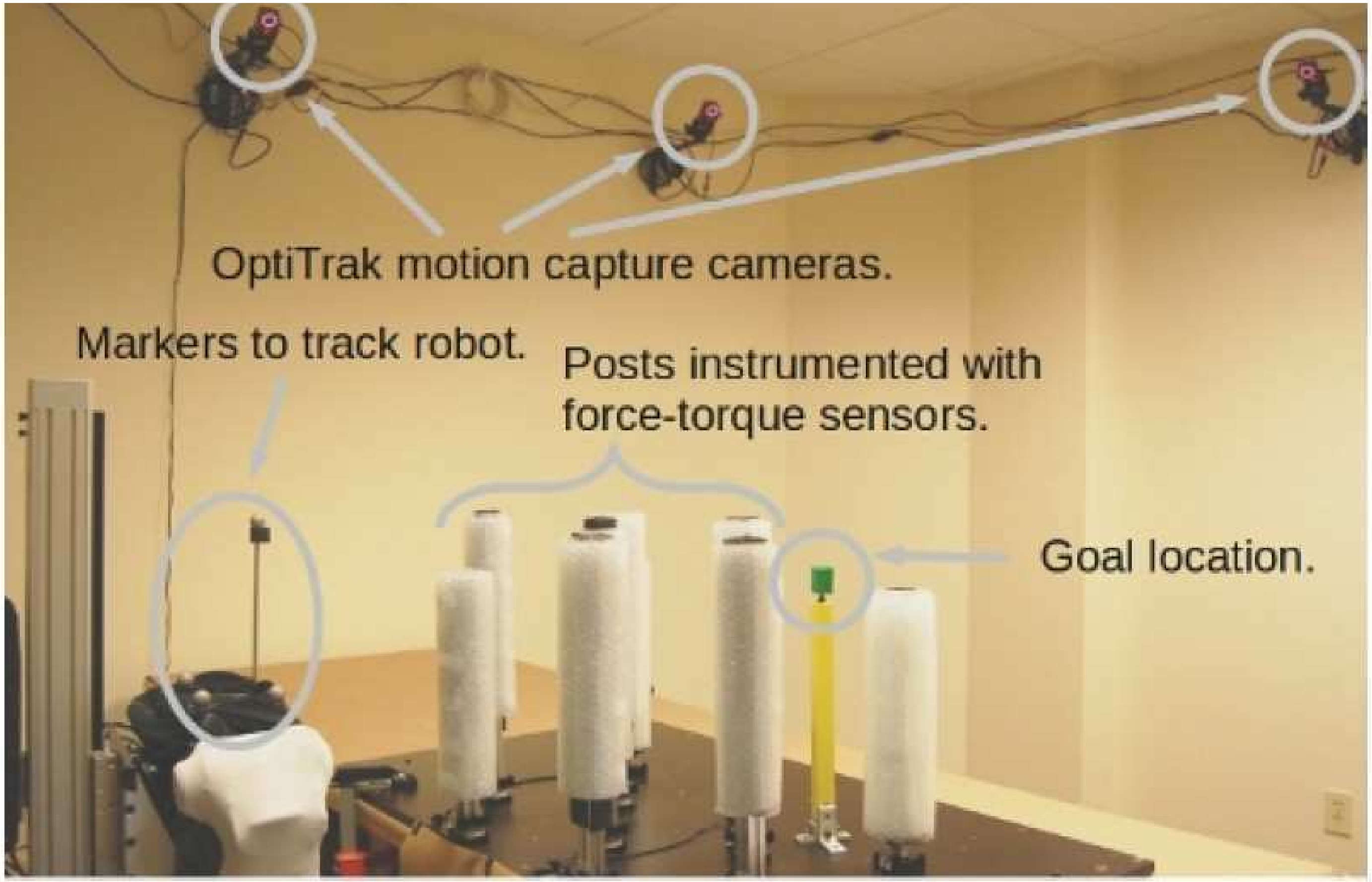

2.1N

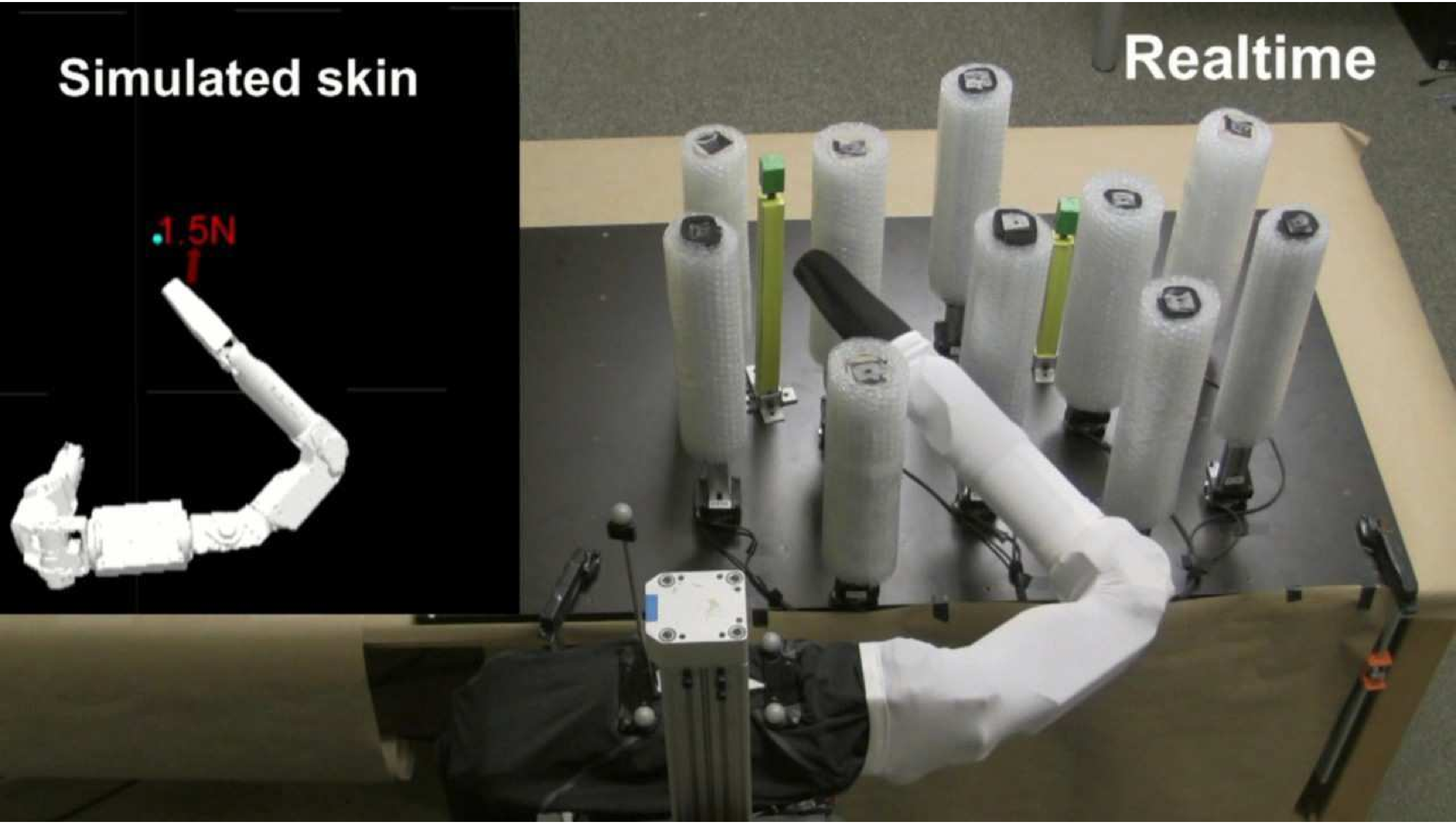
Evaluation of Our Quasi-static Model Predictive Controller with Real Robots

Hardware-in-the-loop Simulation of Whole-arm Tactile Sensing



6-axis force-torque sensor


OptiTrak motion capture cameras.
Markers to track robot.
Posts instrumented with force-torque sensors.
Goal location.

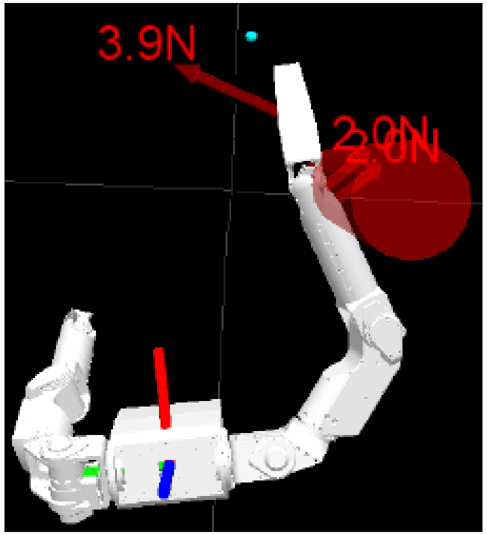

Simulated skin
1.5N
Realtime

4 x realtime

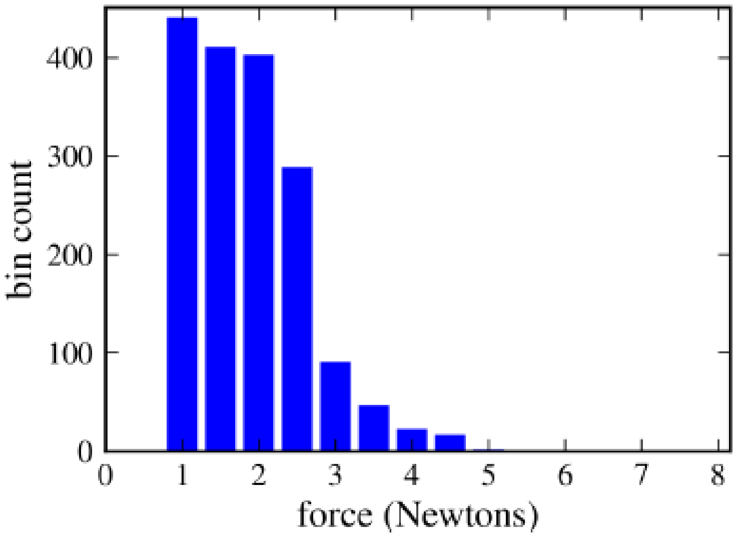
MPC vs. Baseline Controller for 5 Targets given Same Initial Conditions

- 5/5 vs. 3/5 targets reached successfully
- 5.6 N vs. 17.7 N average max contact force
- 5.5 N vs. 14.3 N average force above 5N

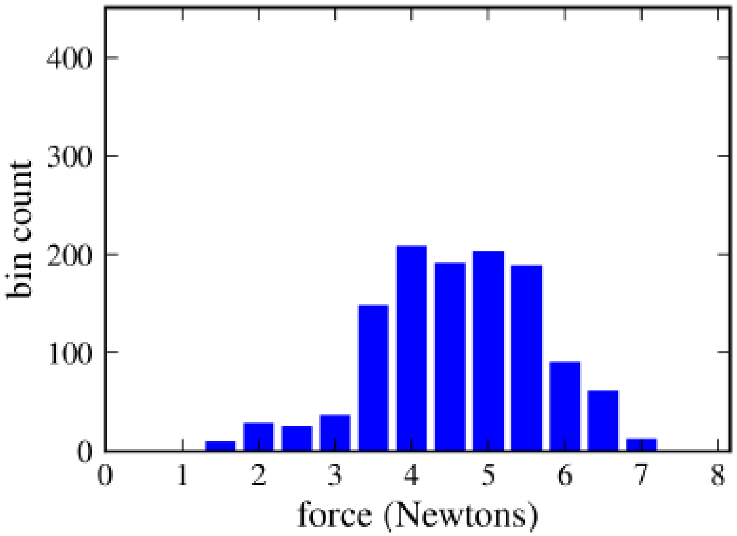
Achieving Lower Forces in a Region Defined as Fragile

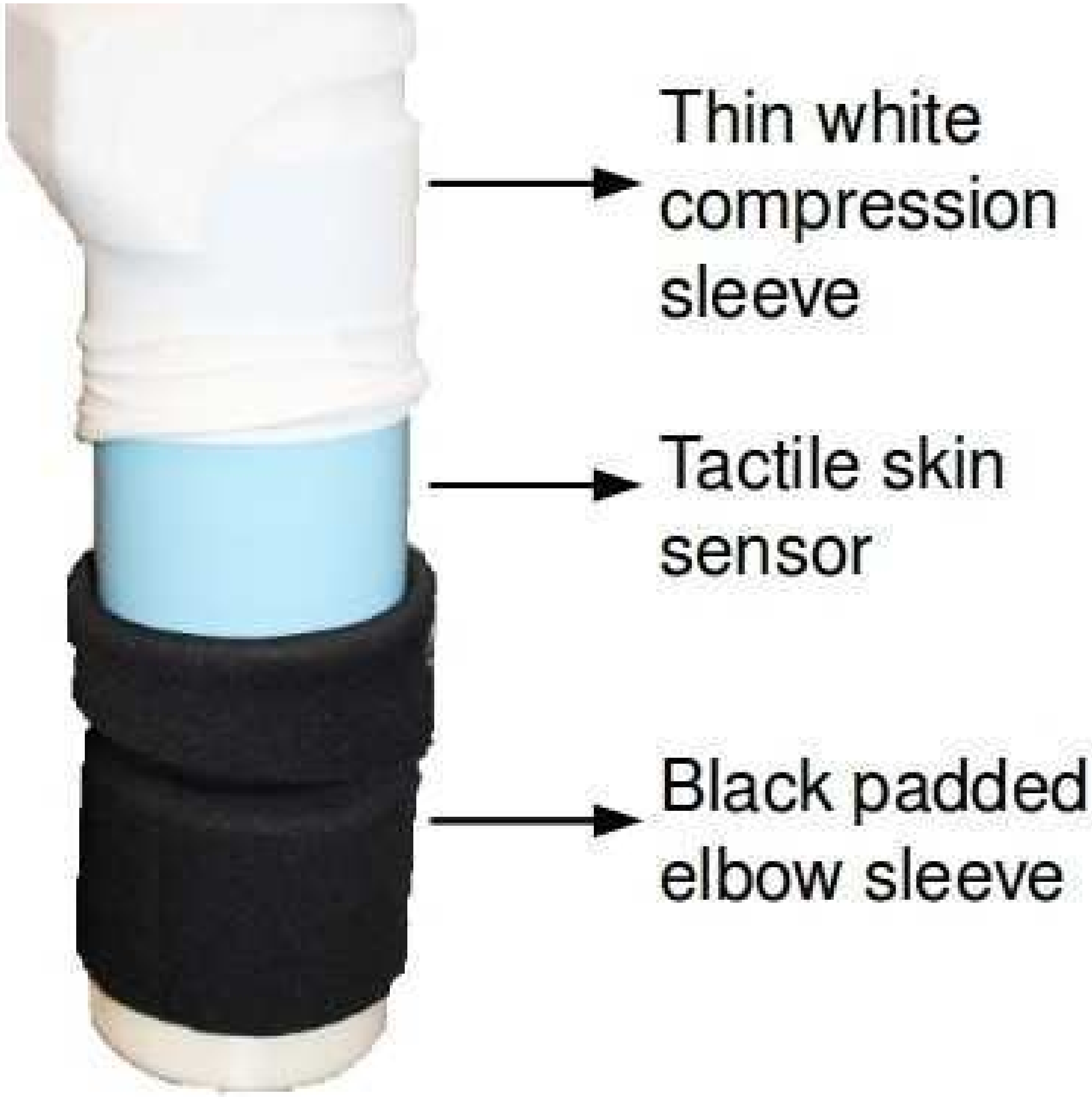


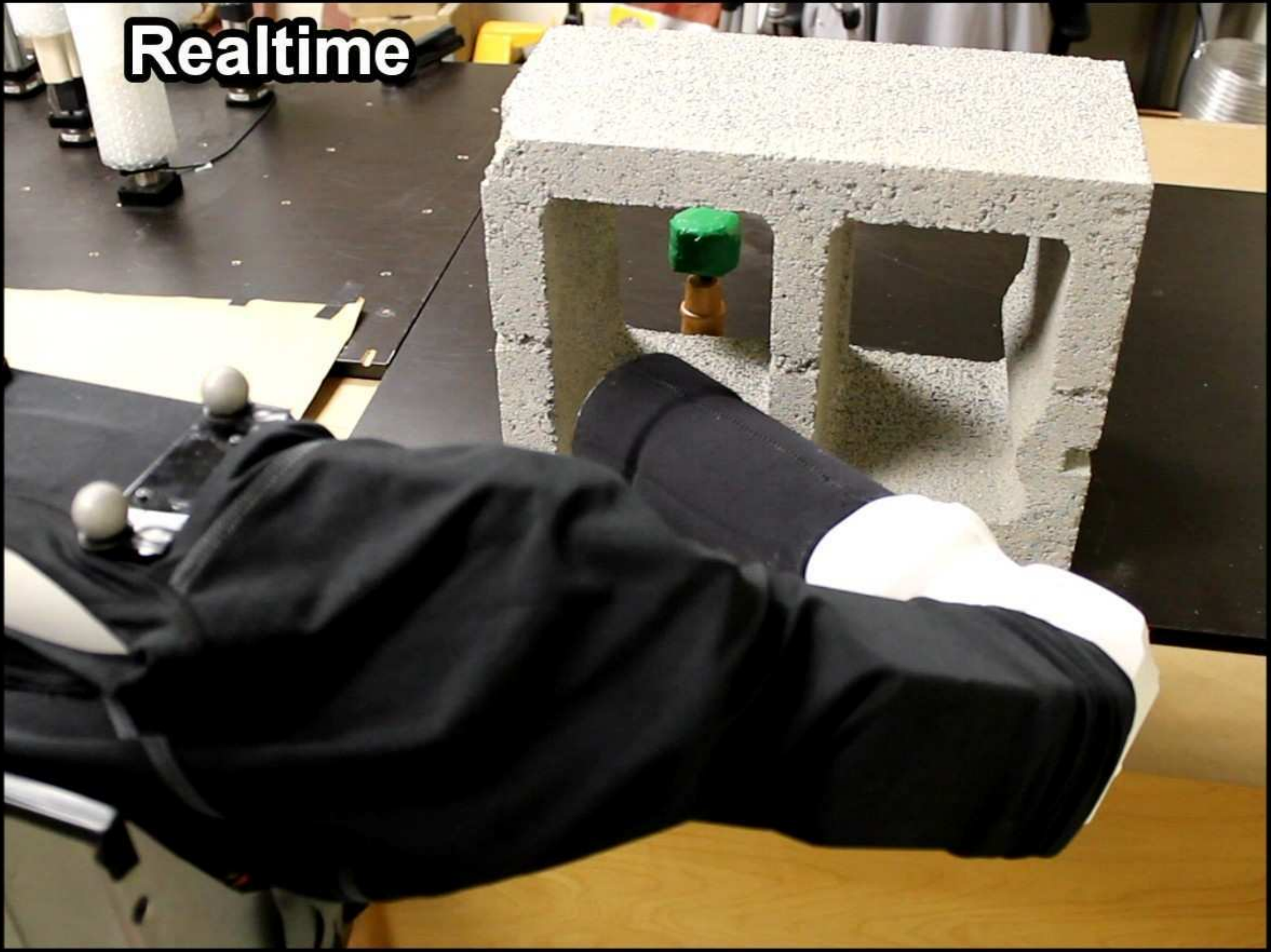
Fragile Region



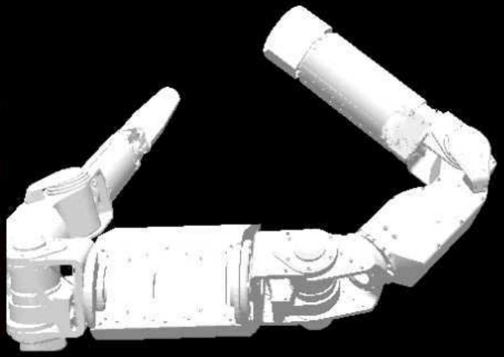
Everywhere Else



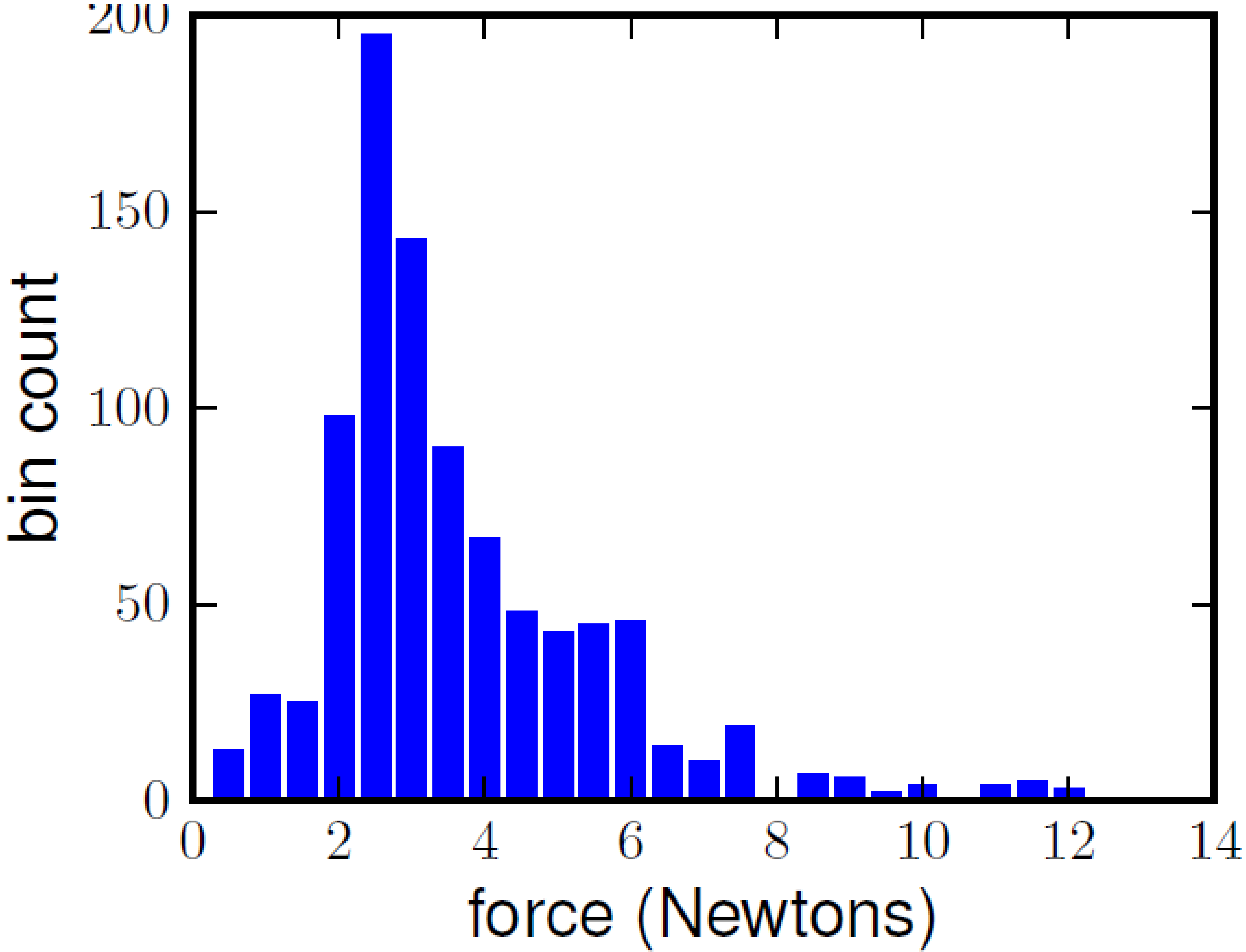

Thin white compression sleeve
Tactile skin sensor
Black padded elbow sleeve



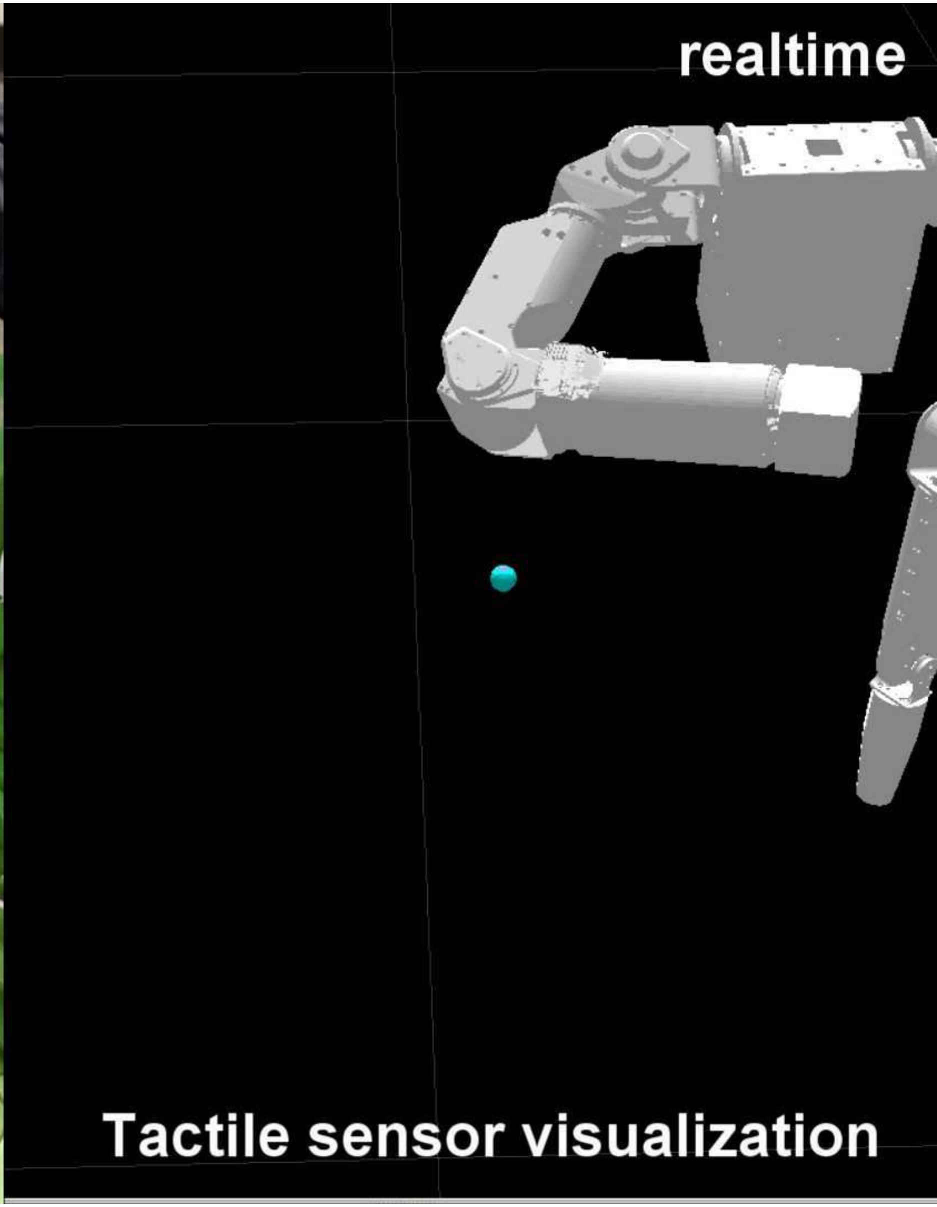
Tactile sensor visualization



Histogram of Contact Forces in Cinder Block Example



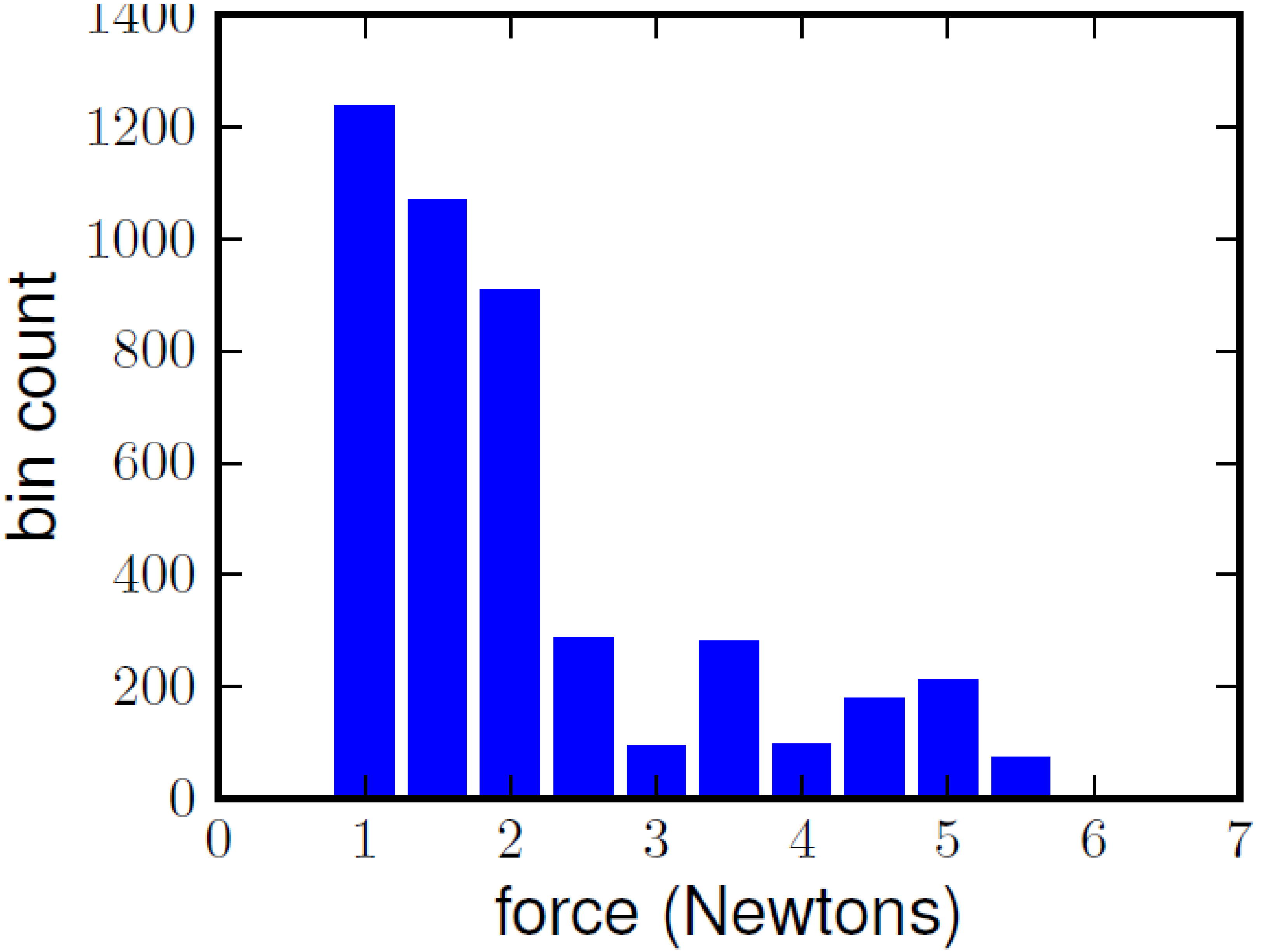
Pushing aside leaves

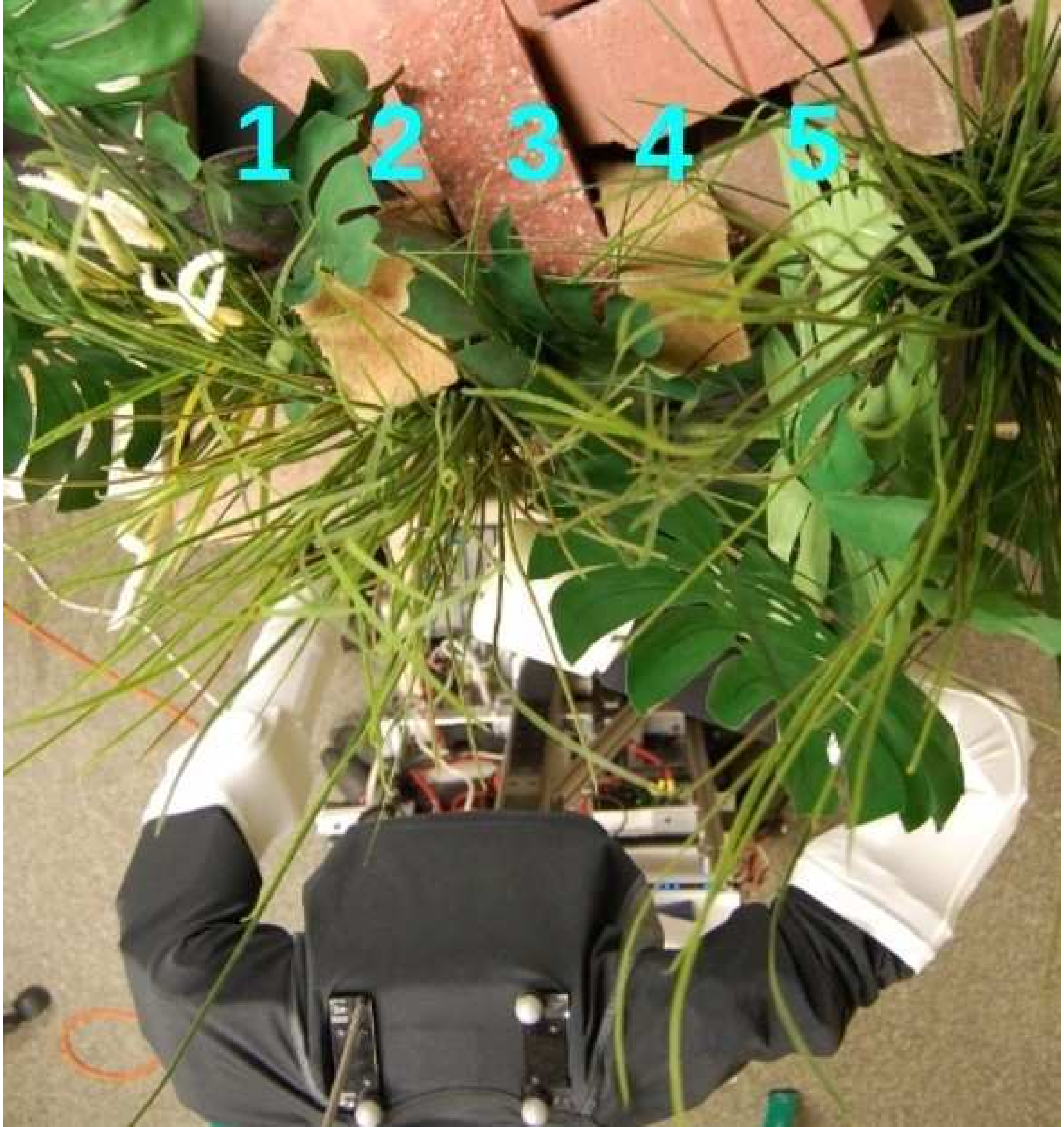


realtime

Tactile sensor visualization

Histogram of Contact Forces in Foliage Example




1
2
3
4
5

Model Predictive Controller vs Baseline Controller in foliage

5 goal locations

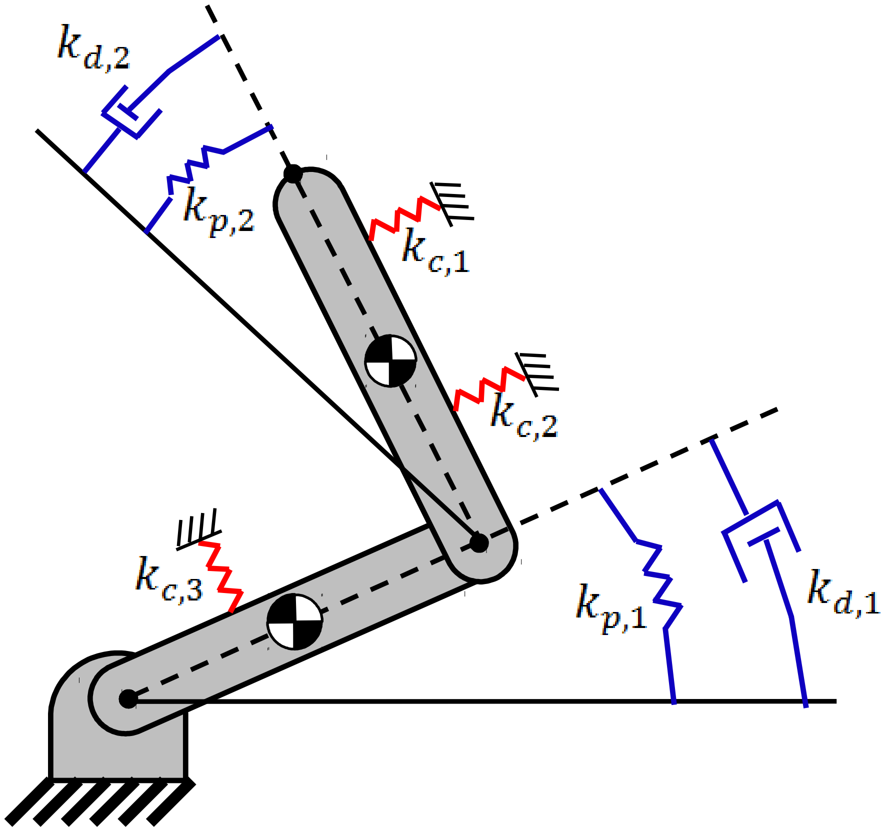
4 attempts for each goal location

MPC vs Baseline Comparison

	MPC	Baseline Controller
Success Rate	3/5	1/5
Exceeded safety threshold of 15N	0/20 attempts	19/20 attempts
Avg. max. force	5.5N	14.5N
Avg. contact force above don't care threshold of 5N	5.2N	9.2N

Dynamic Model Predictive Controller

- Results in superior performance to Quasi-static MPC
  - Faster reaching
  - Lower maximum forces
  - Comparable success rate
- Model Predictive Control (MPC) with
  - Forward model of the robot's dynamics
  - Constraint on predicted collision forces



Marc Killpack and Charles C. Kemp. "Fast Reaching in Clutter While Regulating Forces Using Model Predictive Control", IEEE-RAS International Conference on Humanoid Robots, 2013.

Optimization Performed at Each Time Step

minimize

$$\alpha \left\| \Delta \boldsymbol{x}_{des} - \boldsymbol{J}_{ee}(\boldsymbol{q}[t_0+H+1] - \boldsymbol{q}[t_0]) \right\|^2$$

distance to goal cost

$$+\beta \sum_{t=t_0}^{t_0+H} \sum_{i=1}^{N} \max\left(\boldsymbol{n}_{c_i}^T \boldsymbol{K}_{c_i} \boldsymbol{J}_{c_i}(\boldsymbol{q}[t+1]-\boldsymbol{q}[t_0]) - \left(f_{threshold} - \left\|\boldsymbol{f}_i^{measured}[t_0]\right\|\right), \boldsymbol{0}\right)$$

$$+\zeta \sum_{t=t_0}^{t_0+H} \sum_{i=1}^{N} \max\left(abs(\boldsymbol{n}_{c_i}^T \boldsymbol{K}_{c_i} \boldsymbol{J}_{c_i}(\boldsymbol{q}[t+1]-\boldsymbol{q}[t])) - \Delta f_{rate,i}, \boldsymbol{0}\right)$$

force over threshold cost

$$+\mu \sum_{t=t_0}^{t_0+H} \left\|\Delta \boldsymbol{q}_{des}[t]\right\|^2$$

control effort cost

subject to : $(for\ t = t_0 \ldots t_0 + H)$

state prediction

$$\begin{aligned}
\begin{bmatrix} \dot{\boldsymbol{q}}[t+1] \\ \boldsymbol{q}[t+1] \end{bmatrix} &= \boldsymbol{A}_d[t] \begin{bmatrix} \dot{\boldsymbol{q}}[t] \\ \boldsymbol{q}[t] \end{bmatrix} + \boldsymbol{B}_d[t] \begin{bmatrix} \boldsymbol{q}_{des}[t] \\ \sum_{i=1}^{N} \boldsymbol{J}_{c_i}^T \boldsymbol{f}_i^{measured}[t_0] \\ \boldsymbol{q}[t_0] \end{bmatrix} \\
\boldsymbol{q}_{des}[t+1] &= \boldsymbol{q}_{des}[t] + \Delta \boldsymbol{q}_{des}[t]
\end{aligned}$$

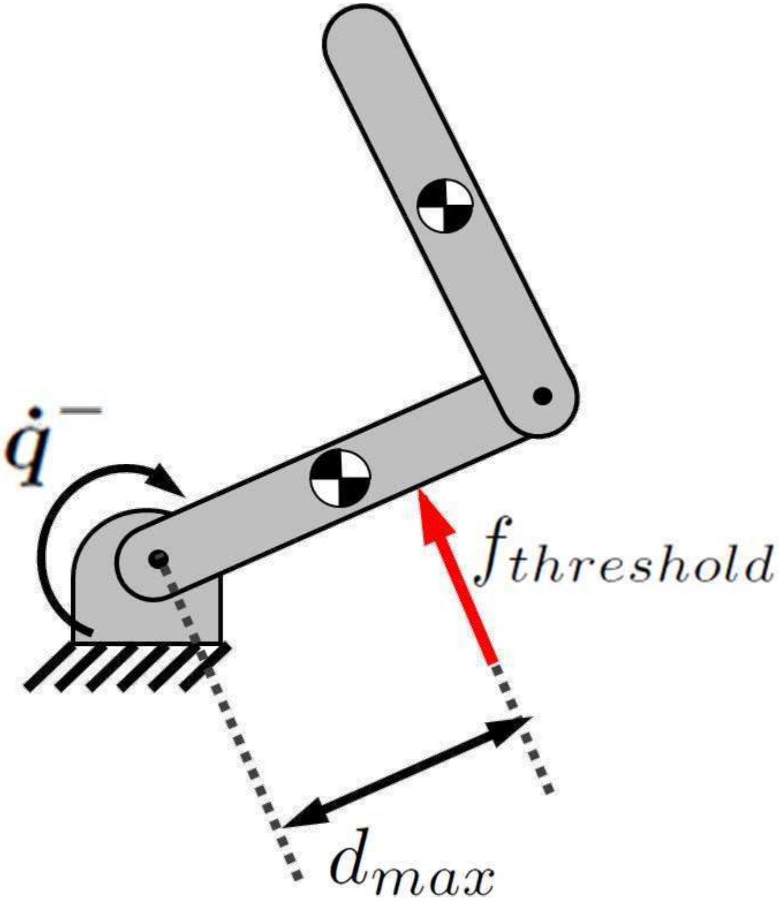
joint limits and actuator constraints

$$\begin{aligned}
\boldsymbol{q}[t+1] &\leqq \boldsymbol{q}_{max} \\
\boldsymbol{q}[t+1] &\geqq \boldsymbol{q}_{min} \\
abs(\Delta \boldsymbol{q}_{des}[t]) &\leqq \Delta \boldsymbol{q}_{max,des}
\end{aligned}$$

$$abs(2\boldsymbol{M}(\boldsymbol{q})\dot{\boldsymbol{q}}[t+1]) \leqq \boldsymbol{\tau}_{max} \Delta t_{impact}$$

collision model constraint

Collision Model
(impulse-momentum constraint)



Perfectly Elastic Collision Model $\dot{\boldsymbol{q}}^+ = -\dot{\boldsymbol{q}}^-$

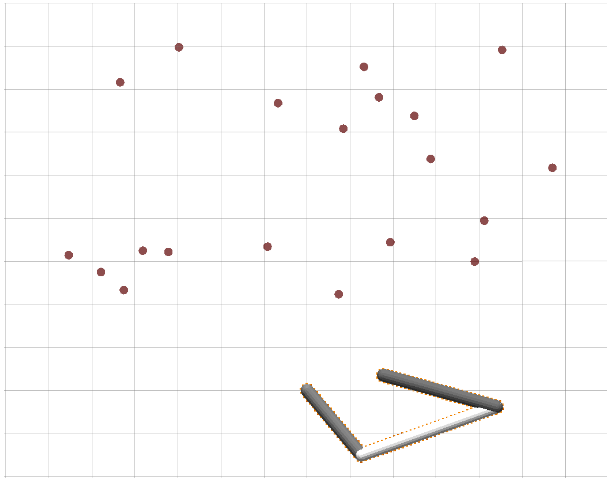
$$\boldsymbol{M}(\boldsymbol{q})(\dot{\boldsymbol{q}}^+ - \dot{\boldsymbol{q}}^-) = 2\boldsymbol{M}(\boldsymbol{q})\dot{\boldsymbol{q}} = \boldsymbol{d}_{max} f_{threshold} \Delta t_{impact}$$

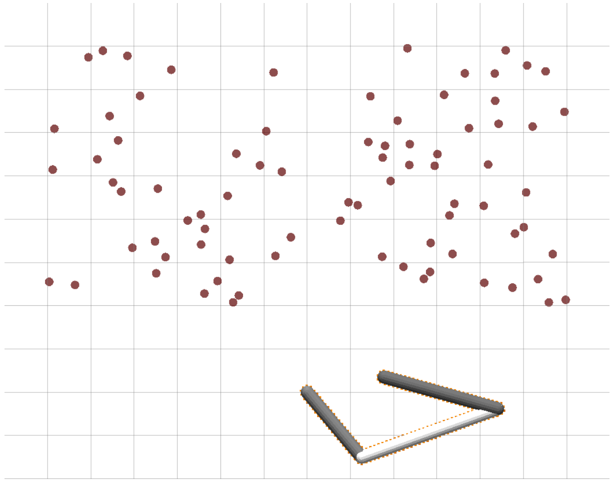
Controller Implementation Details

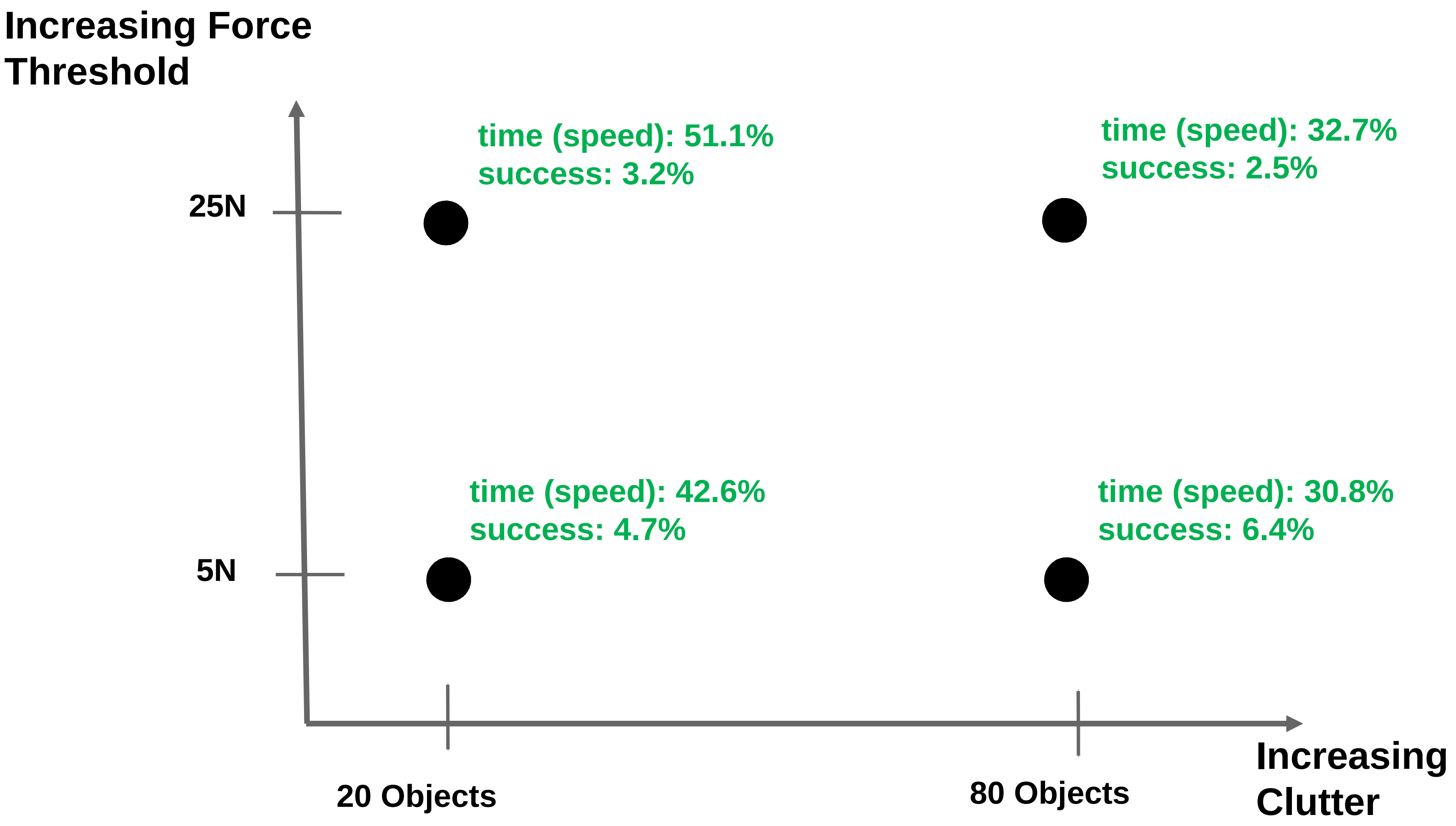
- CVXGEN (Mattingley et al.) – very efficient "embedded" convex optimization
  - Runs in approximately 5-10 ms (~100 Hz)
- Time Horizon of 5 steps
  - 0.01s per step
  - Controller looks ahead 0.05 seconds (20th of a second)
- Parameters tuned via Simulated Annealing using trials in simulation

Evaluation

- 2x2 testing
  - high/low force threshold (5 and 25 N)
  - high/low clutter (20 and 80 fixed objects)
- Expect dynamic MPC to outperform quasi-static MPC
  - With high force threshold (e.g. slipping)
  - With lower clutter (e.g. acceleration in freespace)





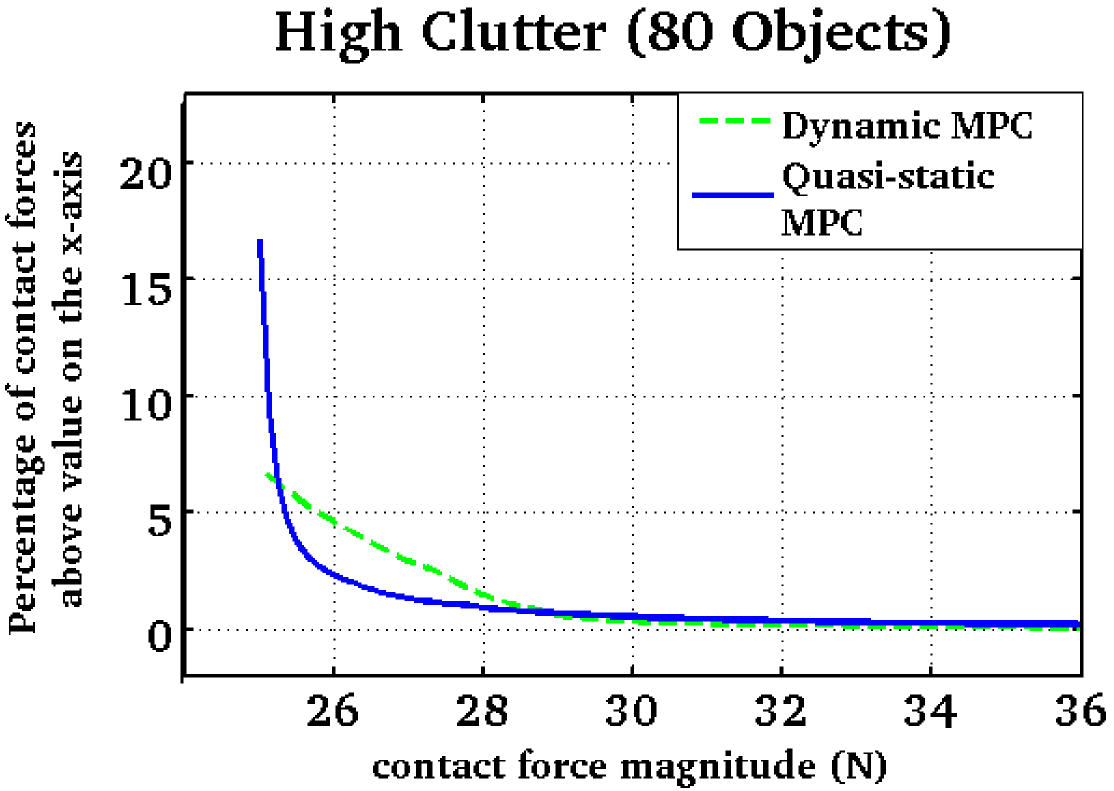

Increasing Force Threshold
25N
5N
time (speed): 51.1%
success: 3.2%
time (speed): 32.7%
success: 2.5%
time (speed): 42.6%
success: 4.7%
time (speed): 30.8%
success: 6.4%
20 Objects
80 Objects
Increasing Clutter

Dynamic MPC was between 1.45 and 2.05 times faster than Quasi-static MPC

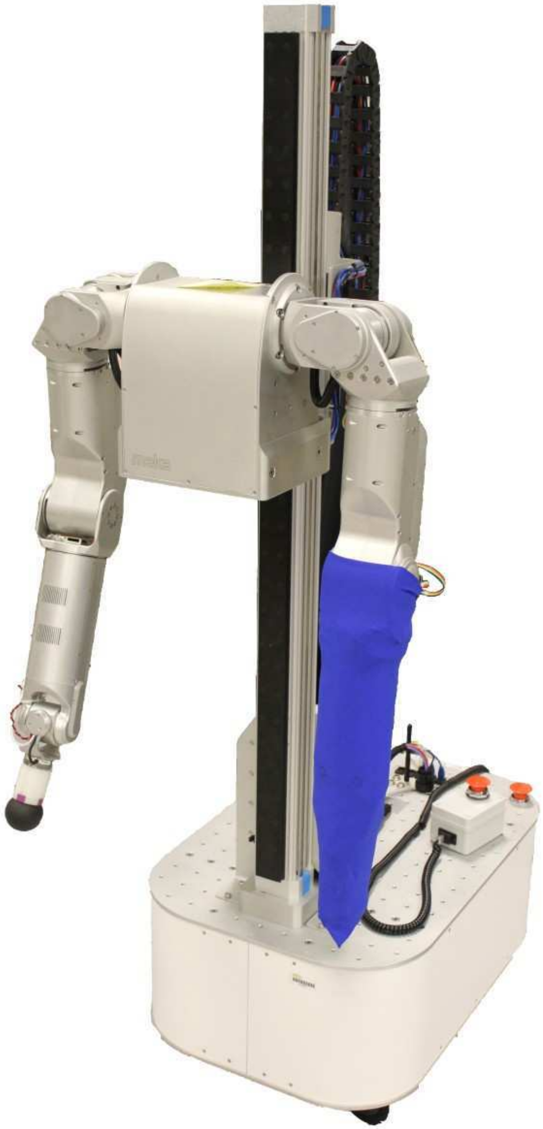
(With Comparable Success Rates)

Percentage of Contact Forces Over a Given Force Value

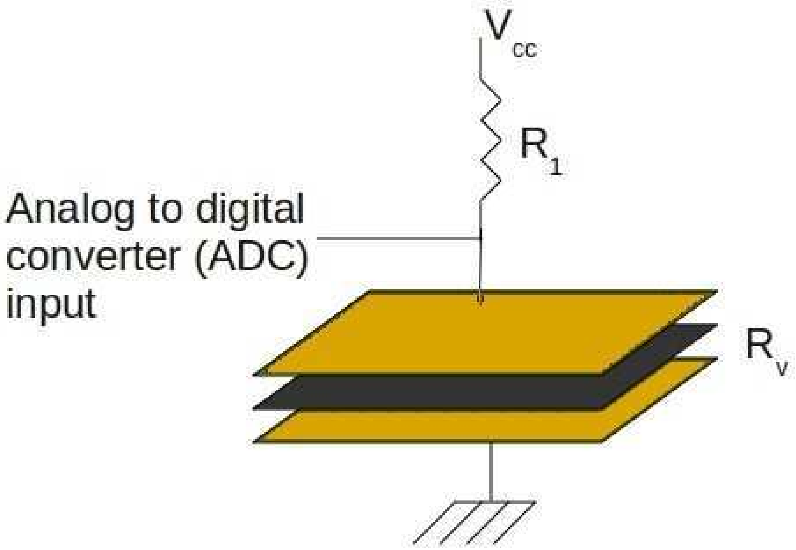
High Force (25 N)

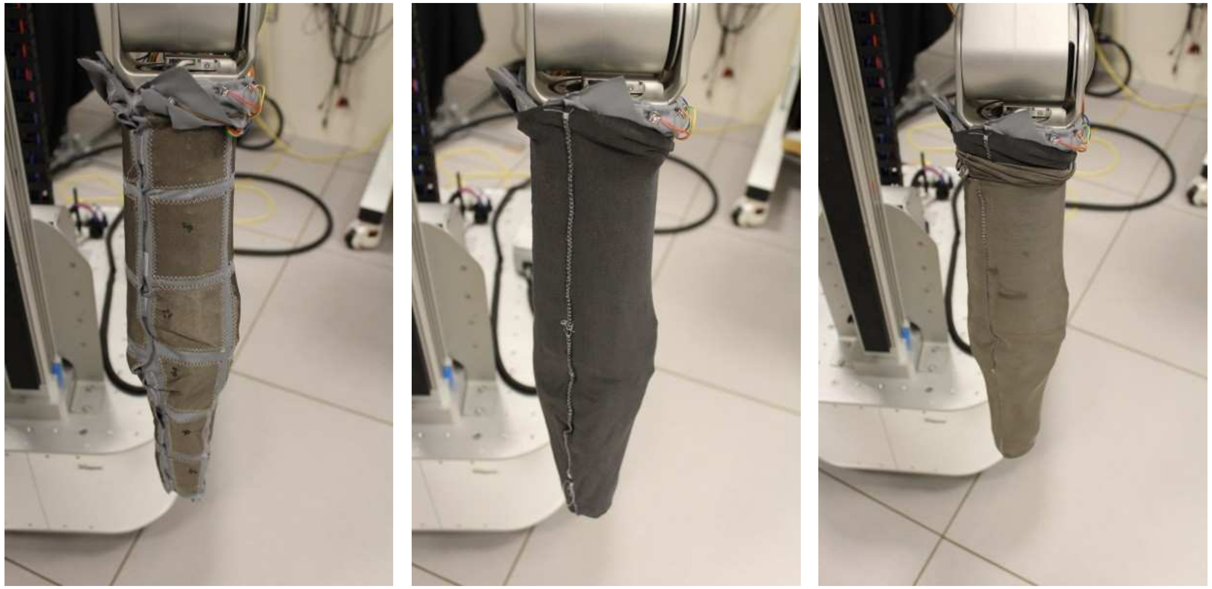


Real Robot Testing for Dynamic MPC



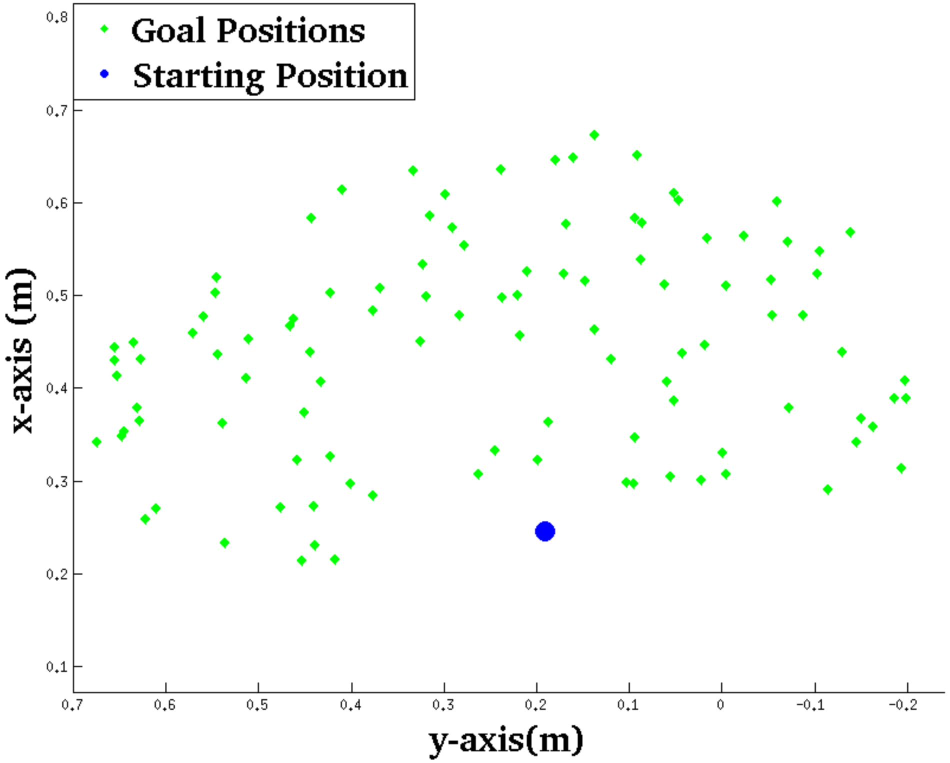
Fabric-based Resistive Tactile Sensor





Tapomayukh Bhattacharjee, Advait Jain, Sarvagya Vaish, Marc D. Killpack, and Charles C. Kemp. “Tactile Sensing over Articulated Joints with Stretchable Sensors”, IEEE World Haptics Conference (WHC 2013), 2013

3D Goal Selection

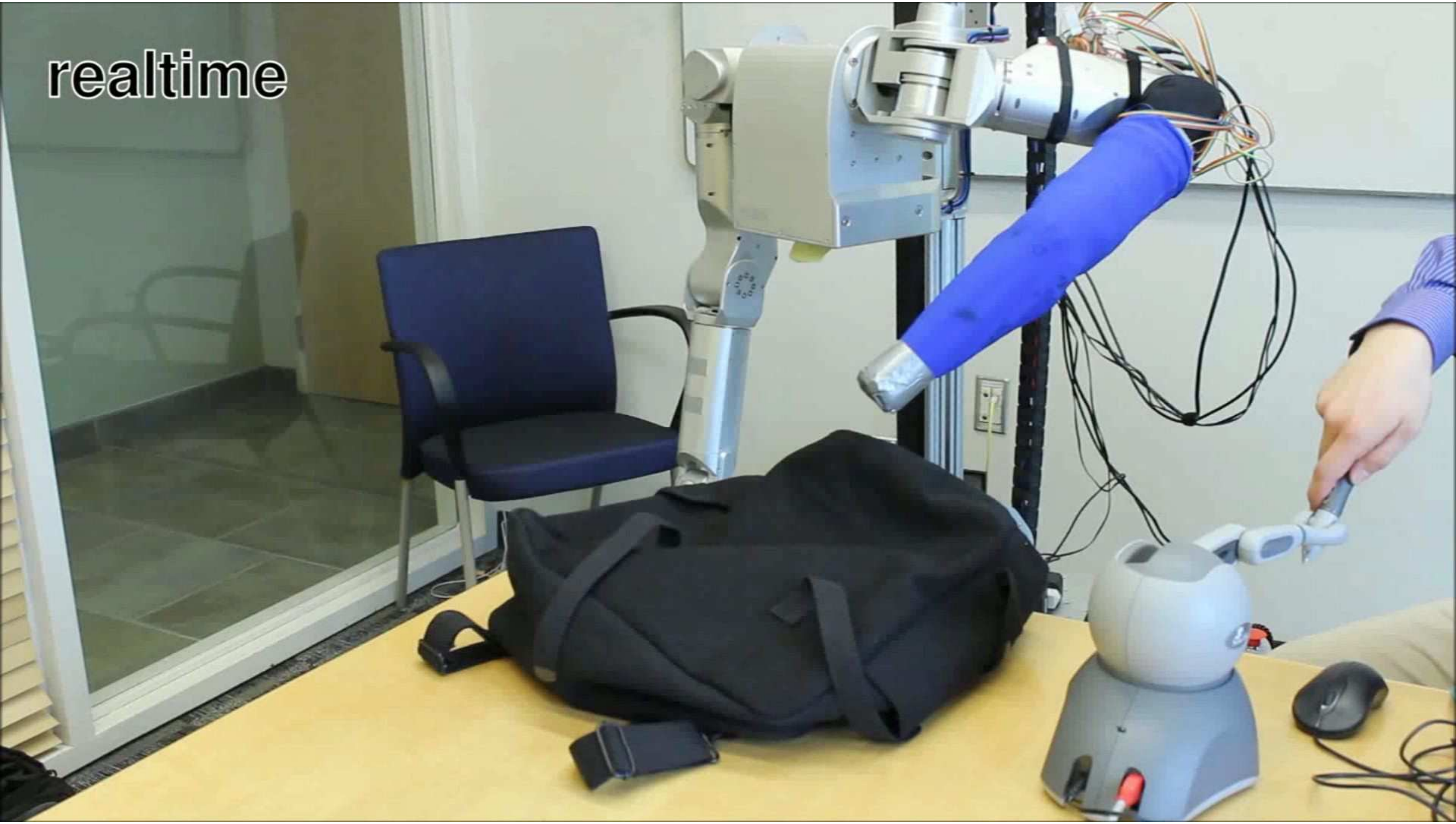


Real Robot Reaching in Foliage

(with dynamic MPC)

Successful Reach Attempts

Real Time Teleoperation



Task-relevant Tactile Perception

Task-relevant Tactile Perception

- Assume
  - category of environment is known
  - environment contains categories of materials
- Approach
  - Data-driven: use labeled tactile data from real reaching
  - Use HMM for each category
  - Update 3D map with detected categories of contact

Category of Foliage

=

Leaves

+

Trunks

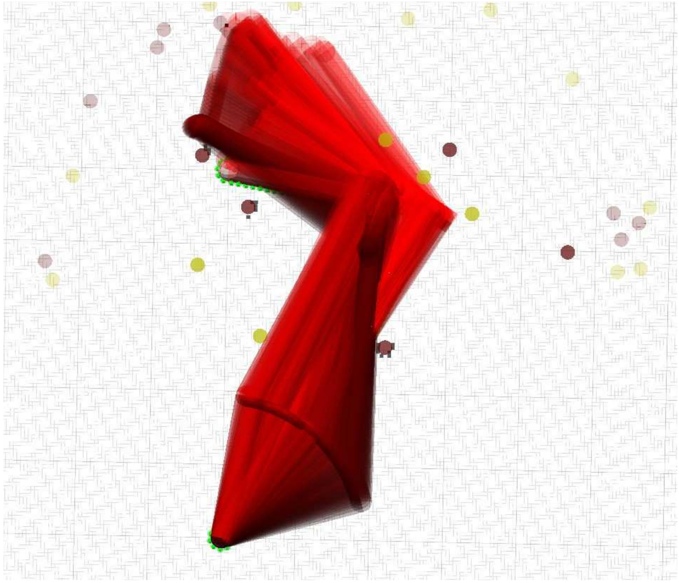
Fast Online Categorization of Incidental Contact into Environment Specific Categories: Leaves vs. Trunk

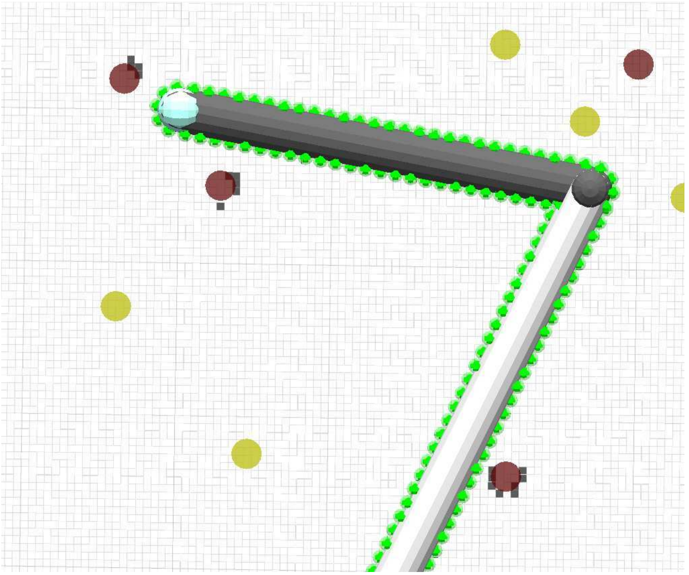
Performance of HMMs for Rapid Categorization	**Taxel-based Training Set**	**Segmentation-based Training Set**
Force Observations with 20 States	**81.4%**	70.75%
Force Observation with 10 States (shown in video)	**80.24%**	72.22%
Force Observation with 5 States	72.91%	61.76%
Force and Motion Observations with 20 States	73.47%	58.50%
Force and Motion Observations with 10 States	71.98%	55.55%
Force and Motion Observations with 5 States	70.22%	54.41%

Planning and Re-planning with Quasi-static MPC

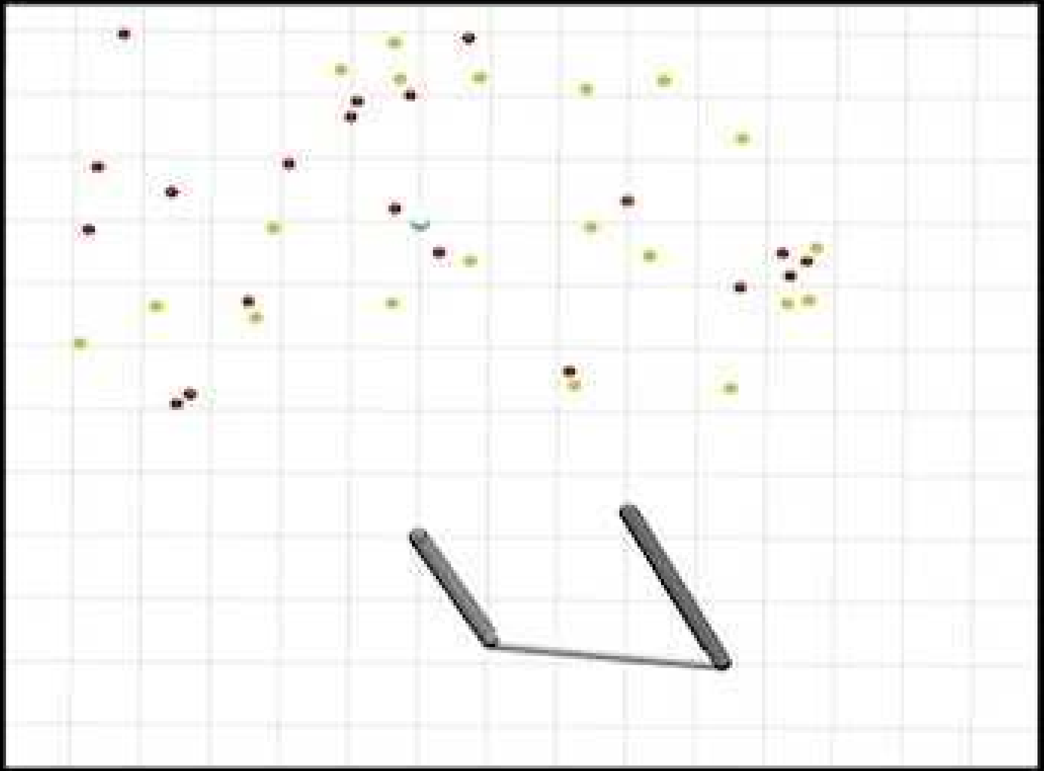
Procedure

1. Ensure start state is valid
2. Plan a path using RRT-Connect in joint space with current map of rigid contact
3. MPC tracks joint trajectory & regulates forces
4. Updates map with rigid contact when detected
5. Goto step 1 when MPC is stuck in a local minimum (unable to reach local goal waypoint)

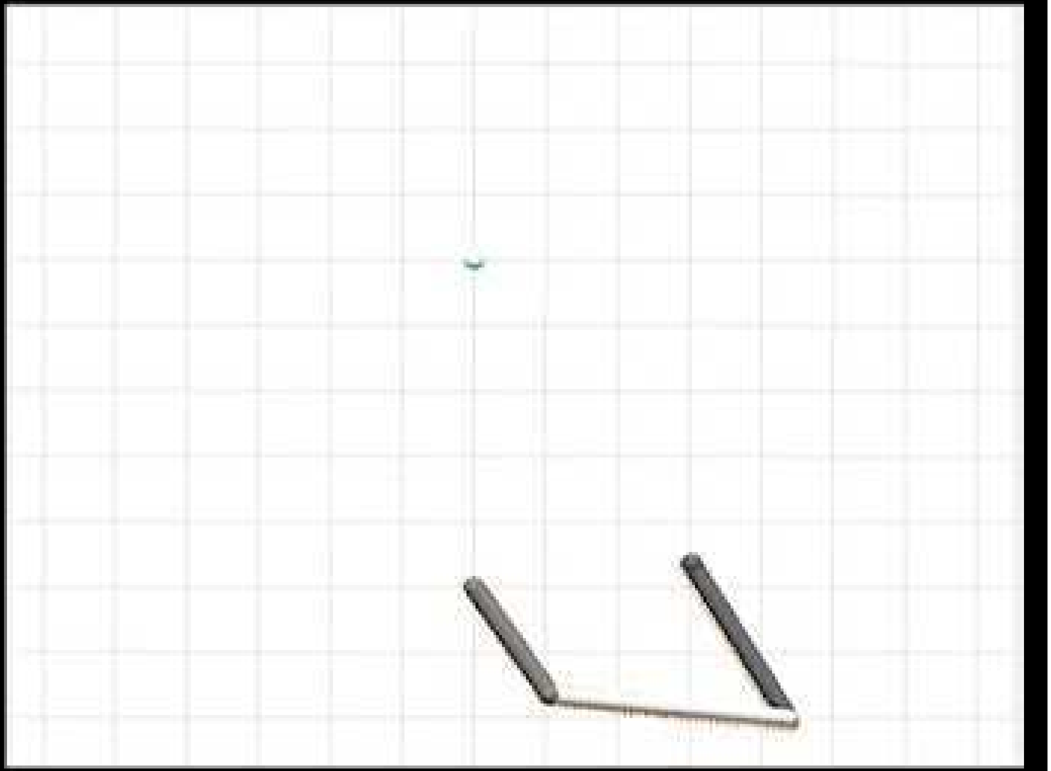




World

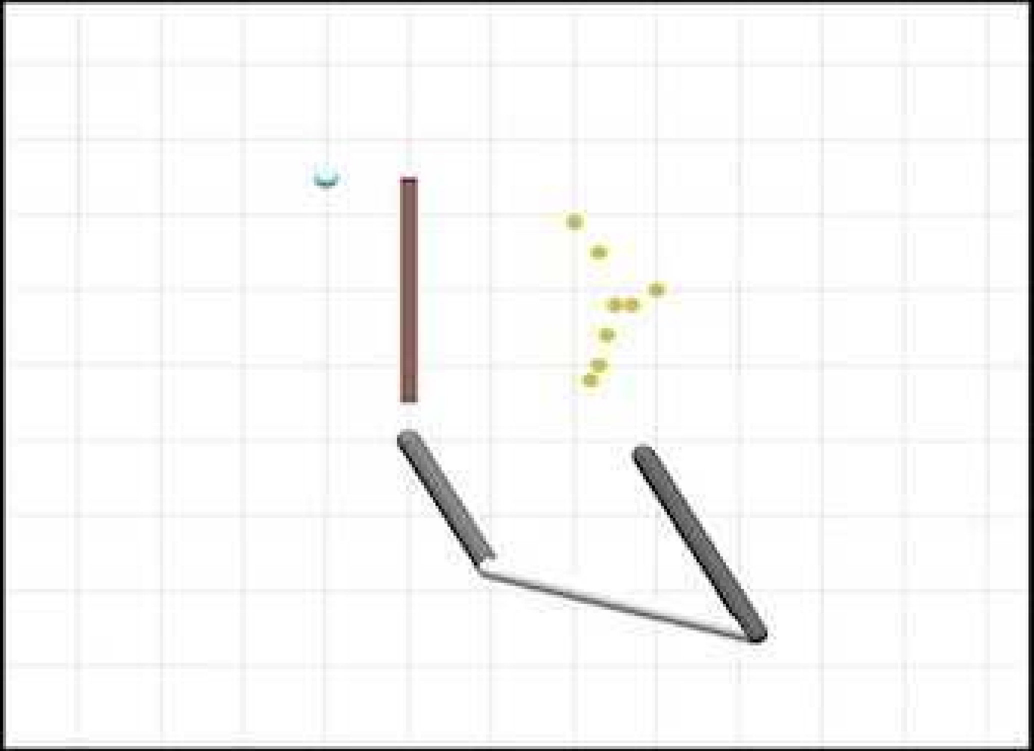


Robot's perspective

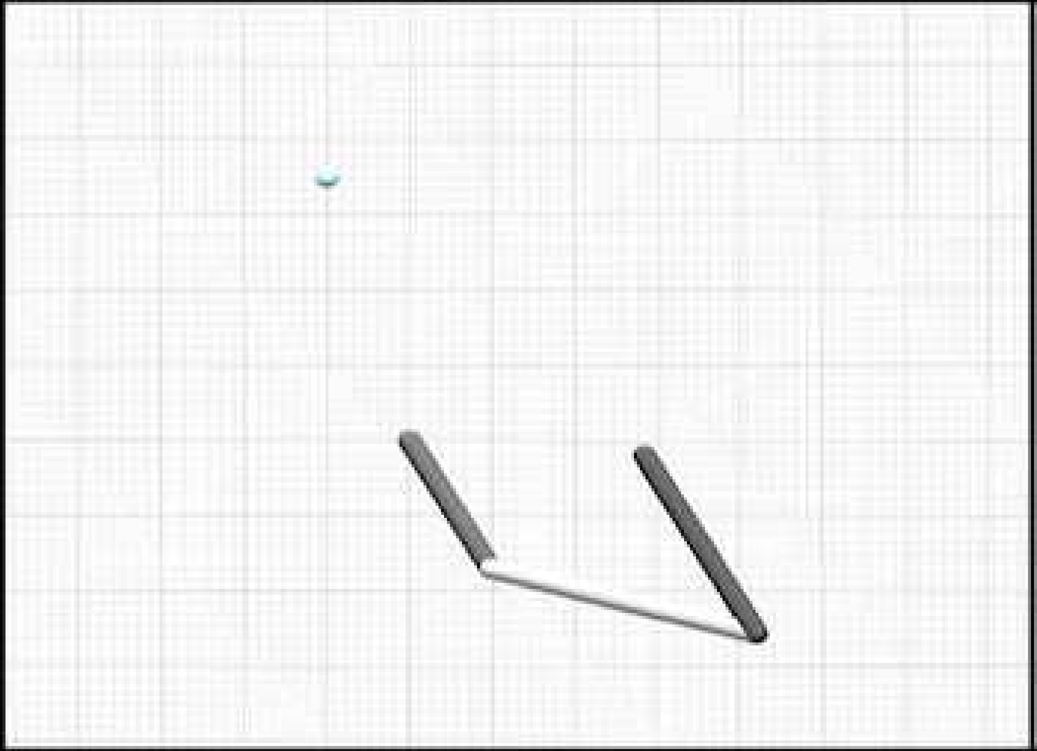


10x

World

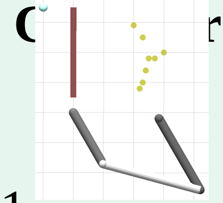
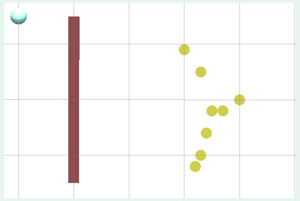
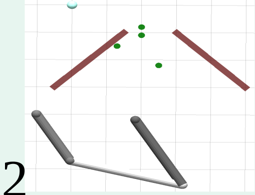
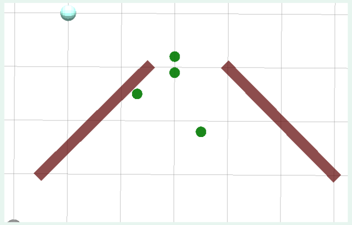
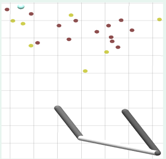
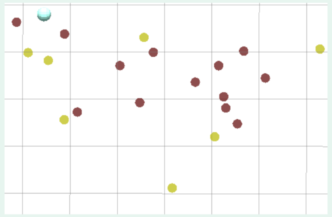
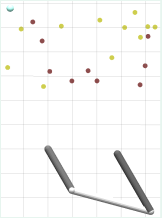
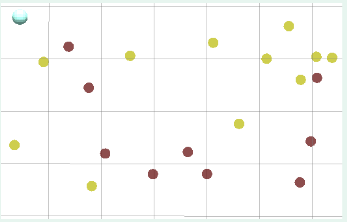
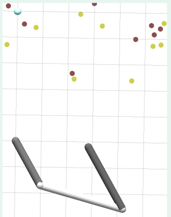
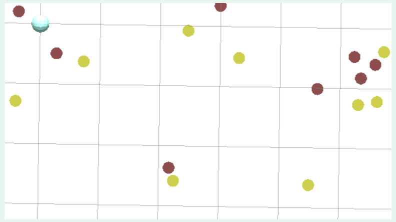


Robot's perspective



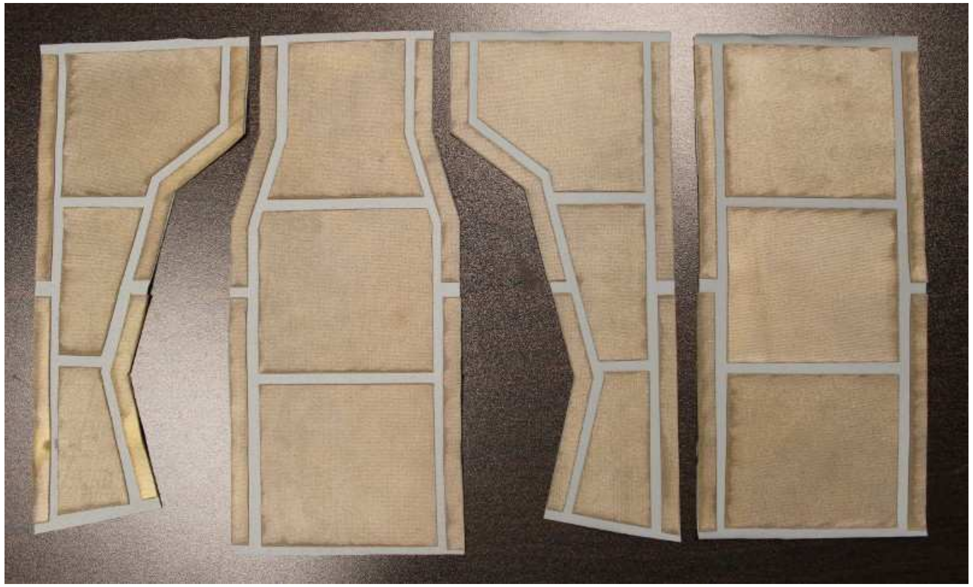
10x

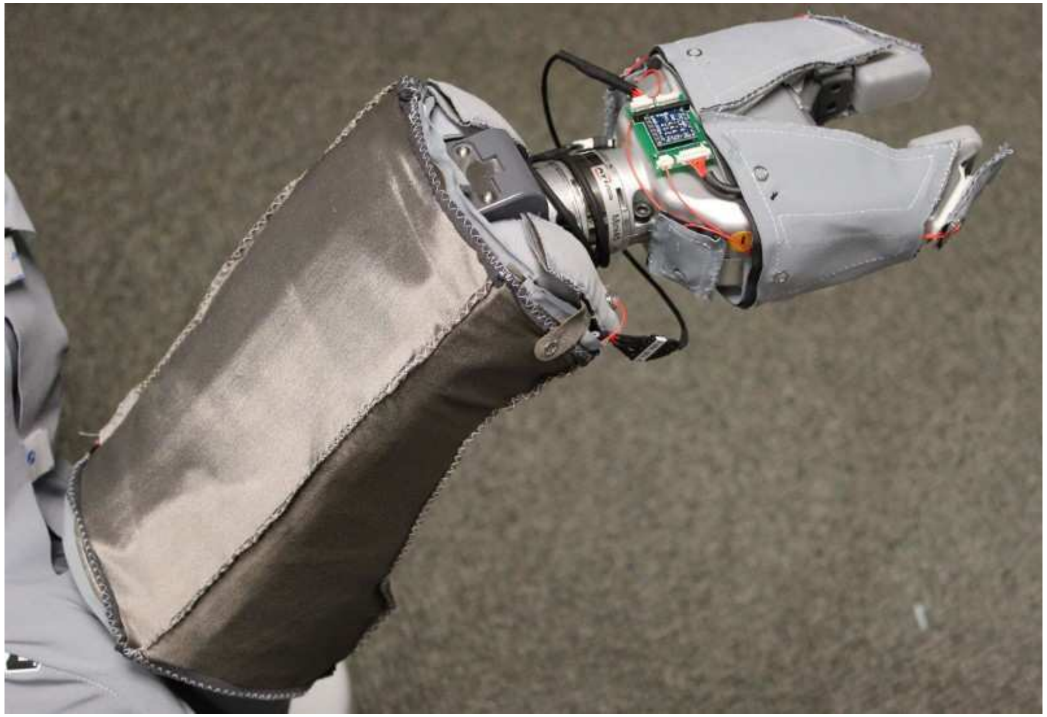
Motion Planning with Tactile Sensing

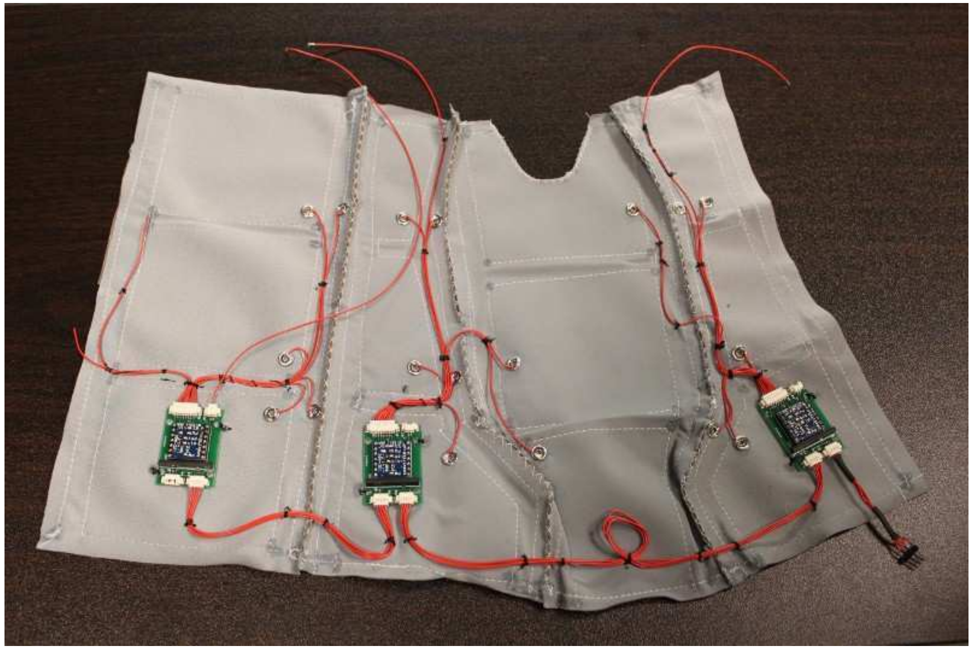
	Map with Robot in Initial C ratio	Map in Isolation	RRT-Connect with Online Updated Map of Rigid Contact	Successive Quasi-static MPC Reaches
1			Success	Failed
2			Success	Failed
3			Success	Failed
4			Failed	Failed
5			Success	Failed

Having Henry Try Out the Quasi-static Controller

Stretchable Fabric Tactile Sensors



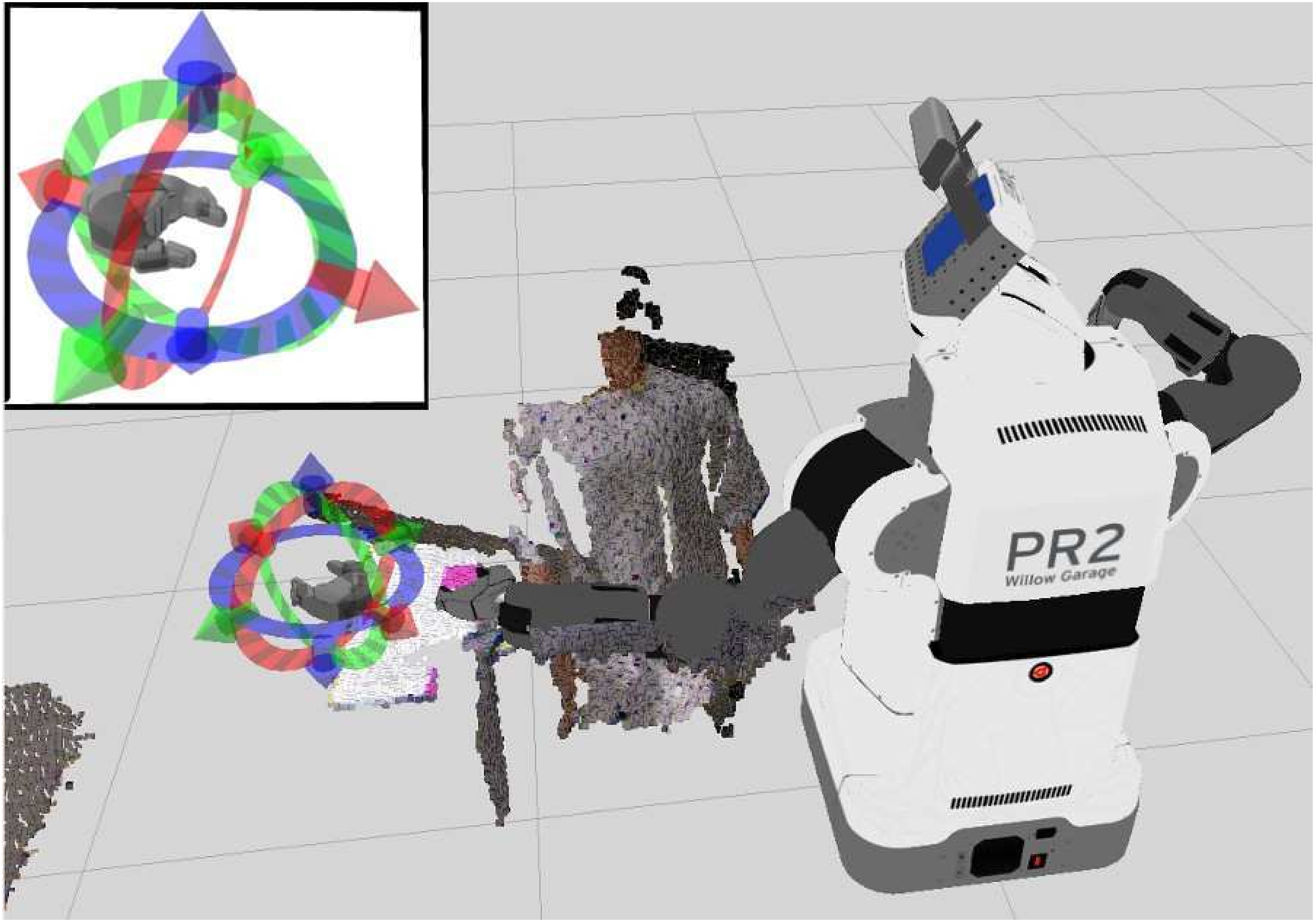


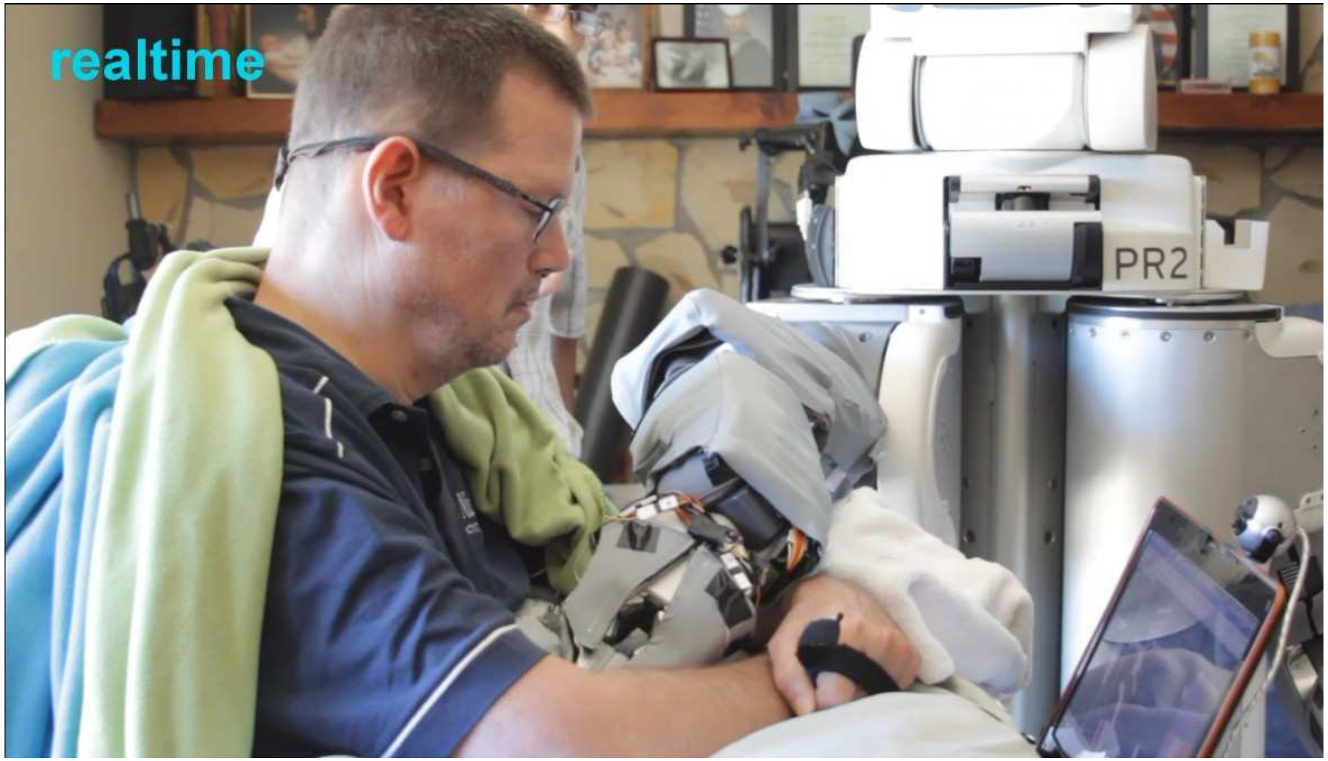


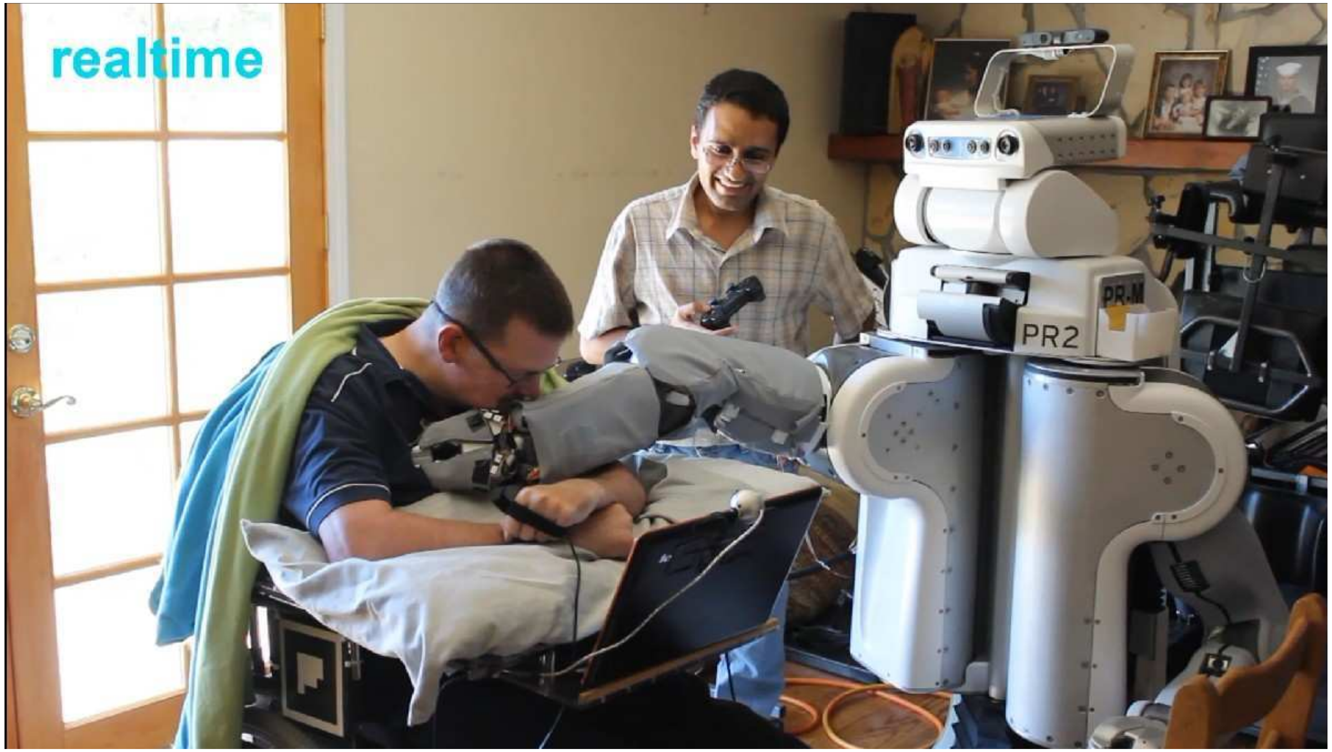
- **41 discrete tactile sensor elements (taxels)**
    - 3 on upper arm
    - 22 on forearm
    - 16 on the gripper
- **Open hardware**

http://www.hsi.gatech.edu/hrl/project_fabric_tactile_sensor.shtml

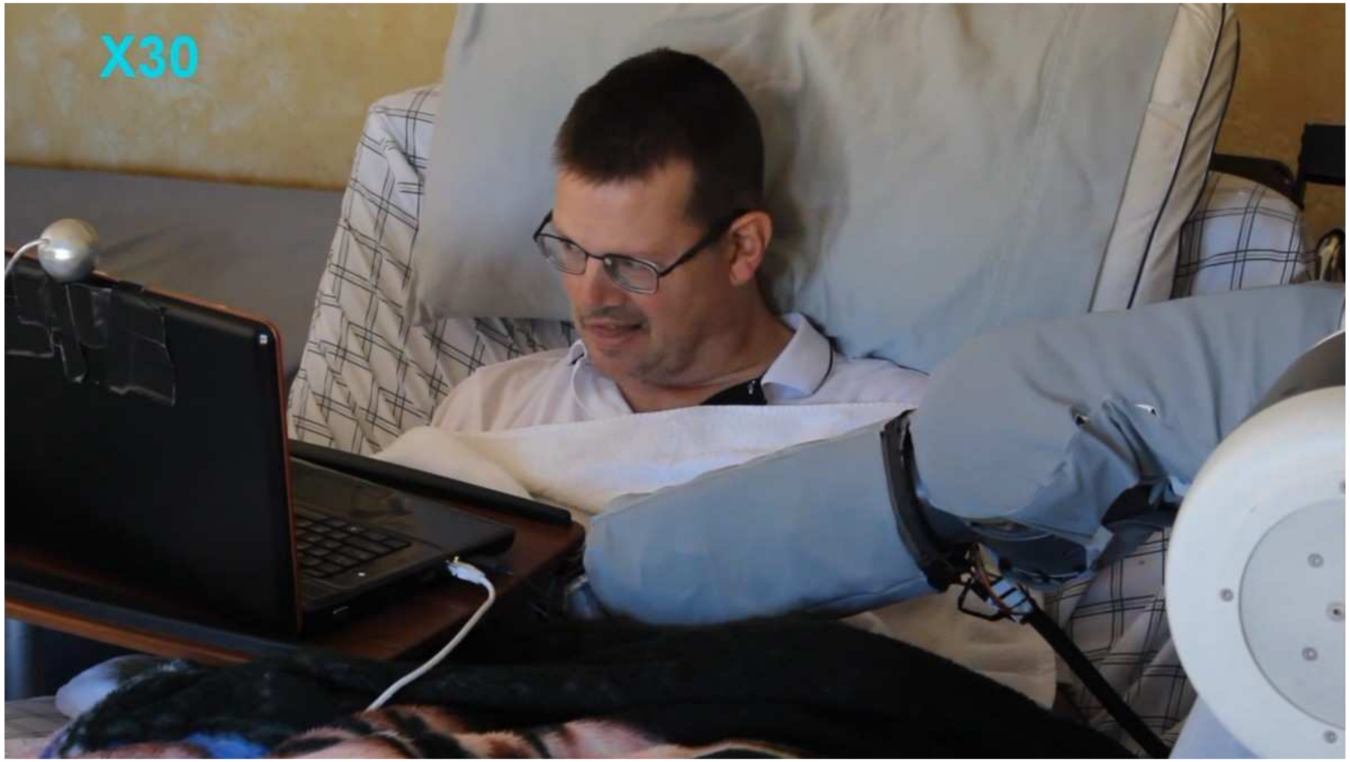
Specify Goal Pose with Interface





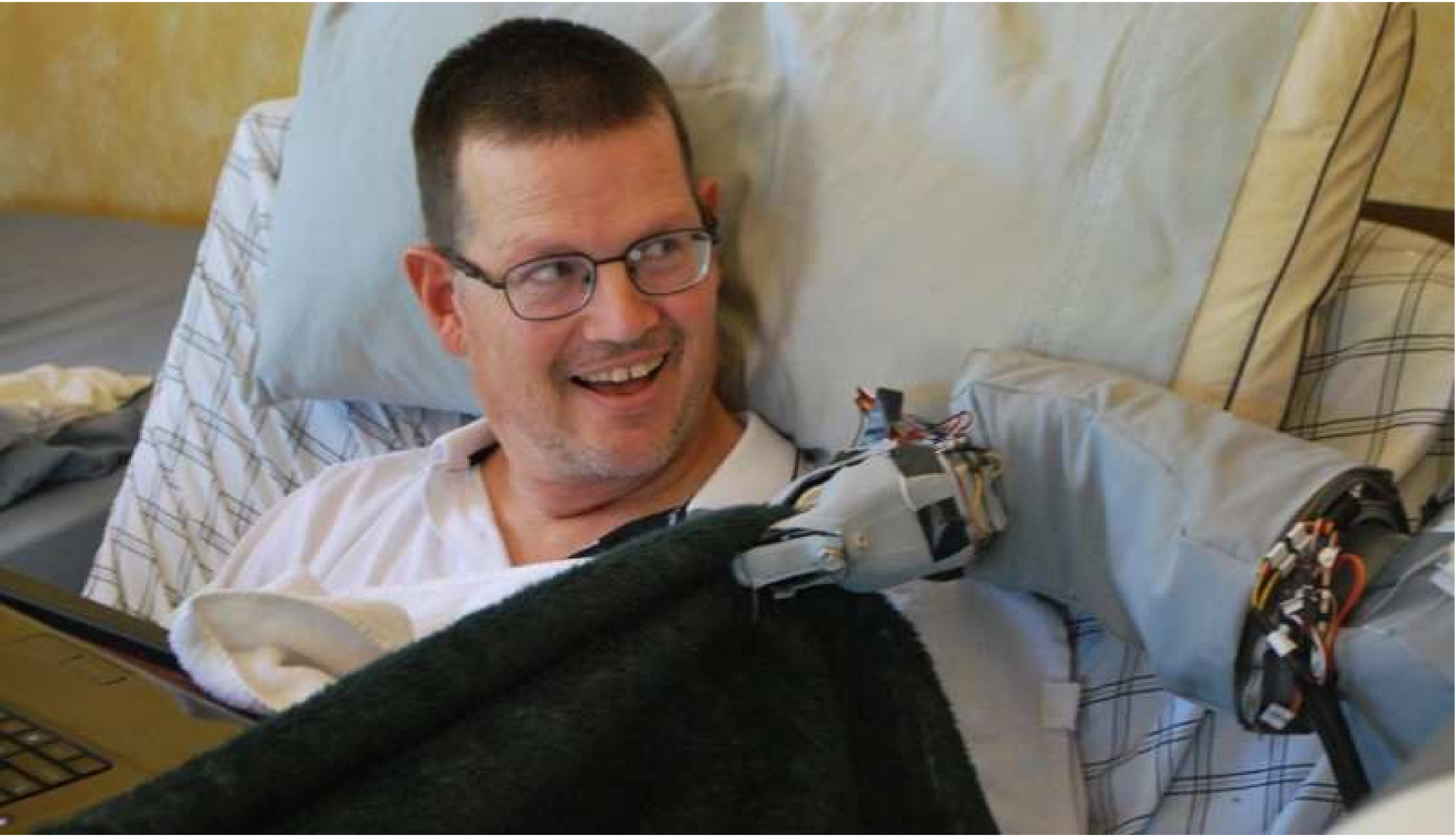


First Use of the System from Wheelchair



Picking Up a Cloth and Wiping Face in Bed

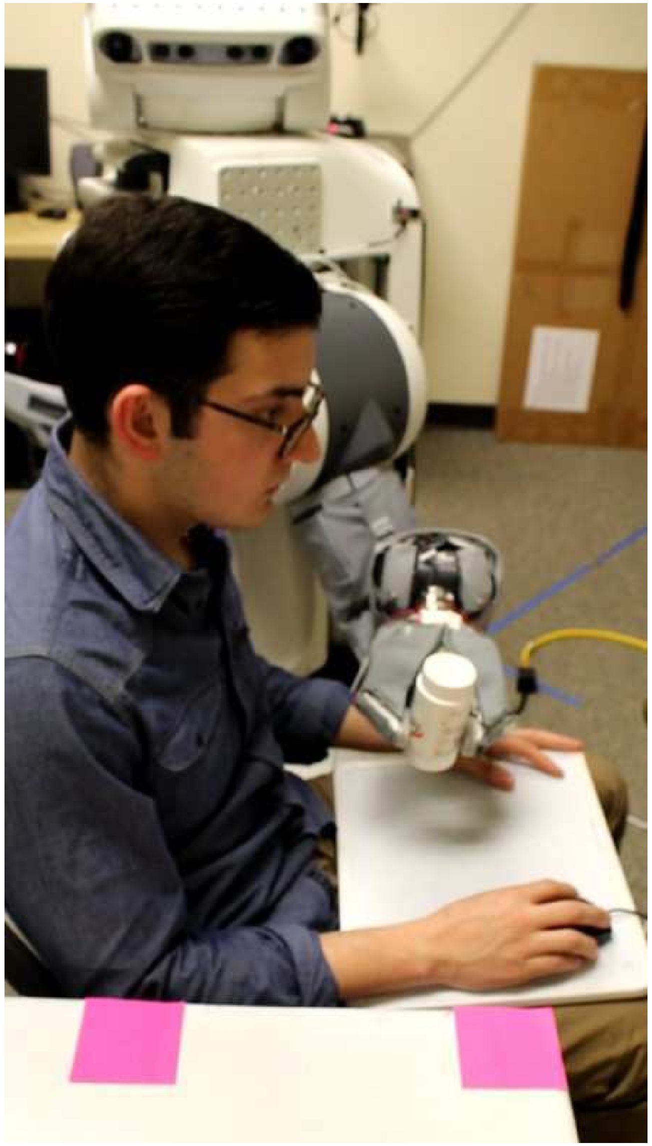
Grasping and Pulling up a Blanket in Bed

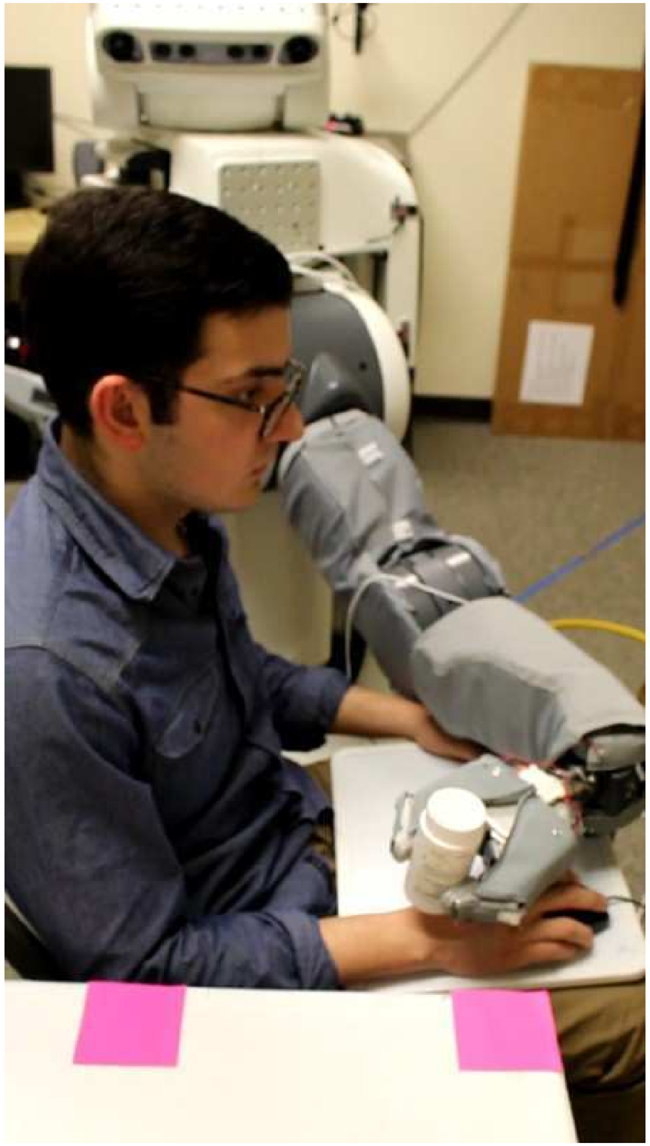


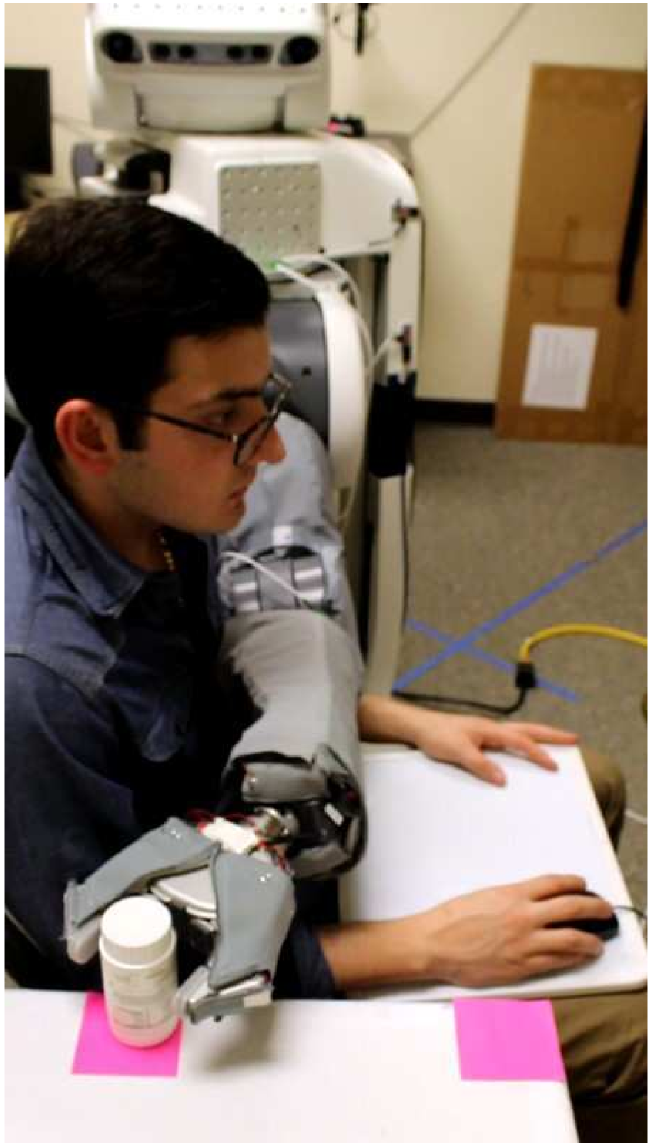
Henry Evans's Comments

- During the tests:
  - "It is very compliant"
  - "**I like it.**"
  - "I think it's a good safety feature because **it hardly presses against me** even when I tell it to."
  - **"It really feels safe to be close to the robot."**
- A week after the tests:
  - "Skin
  - **Overall awesome**
  - **Feels VERY safe**
  - Faster than motion planning
  - **It just wriggles around obstacles**"
  - **"DEFINITELY keep developing this !"**

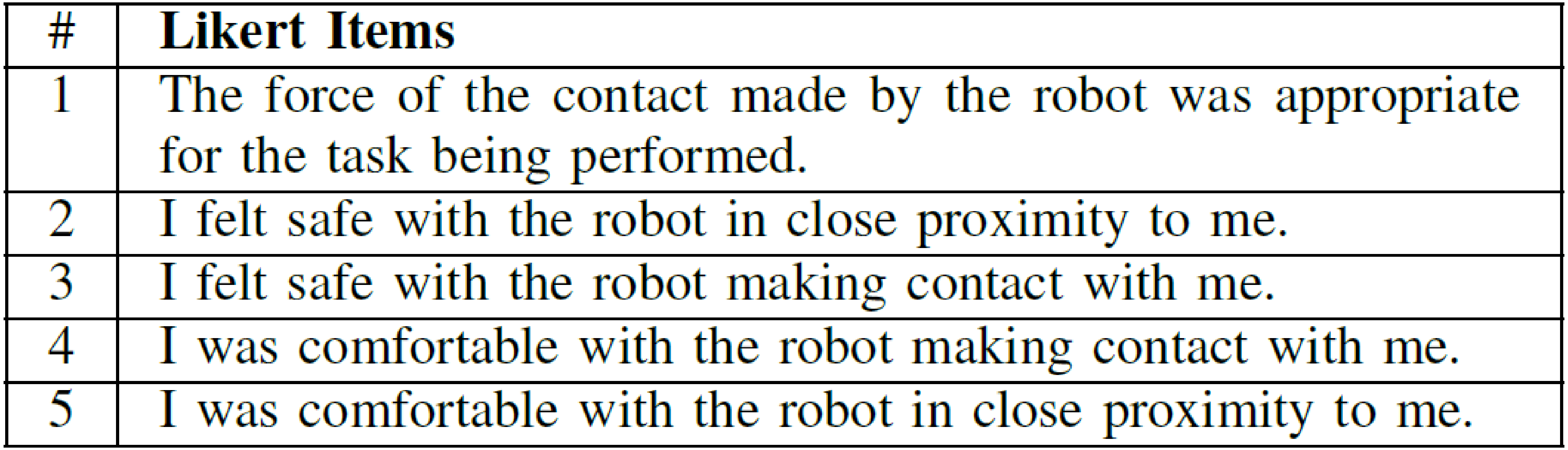
Will contact be acceptable to others?



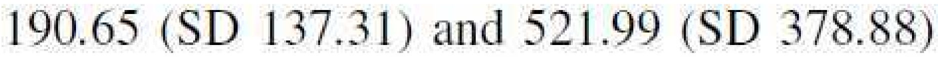




Questions

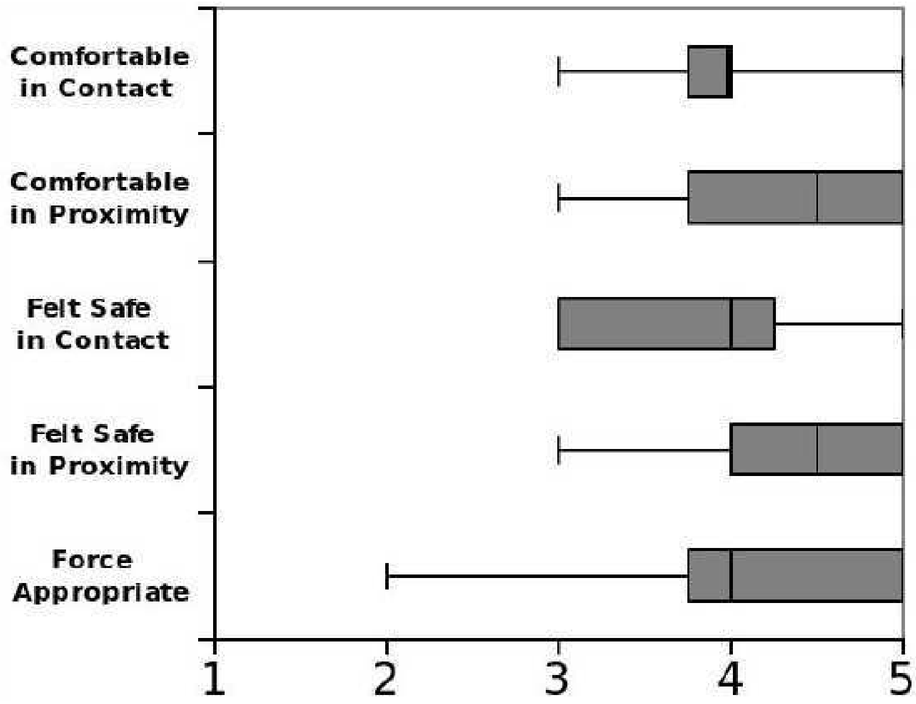


#	**Likert Items**
1	The force of the contact made by the robot was appropriate for the task being performed.
2	I felt safe with the robot in close proximity to me.
3	I felt safe with the robot making contact with me.
4	I was comfortable with the robot making contact with me.
5	I was comfortable with the robot in close proximity to me.

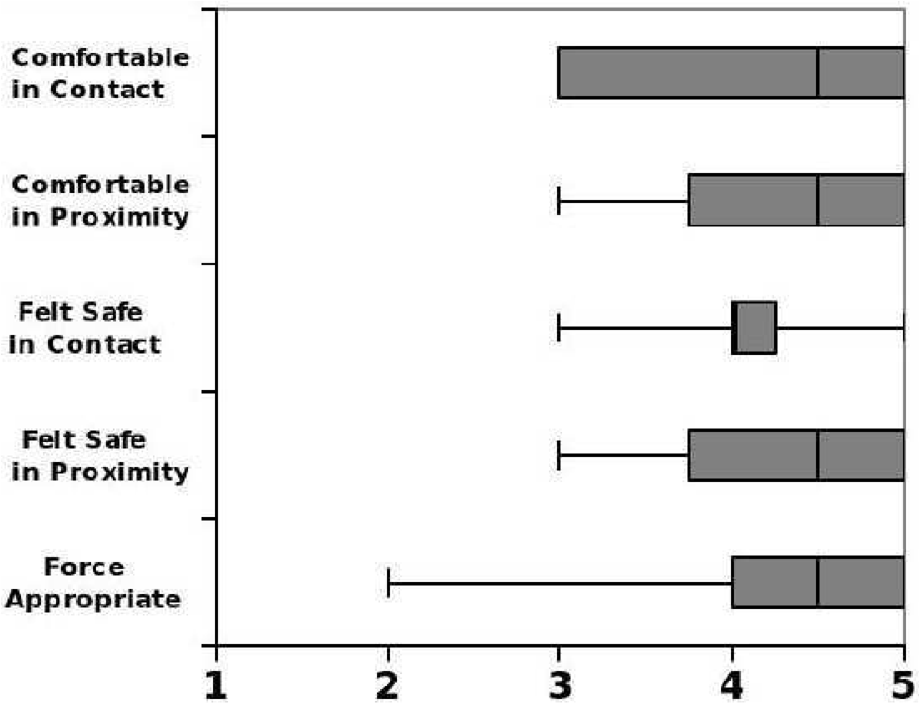


190.65 (SD 137.31) and 521.99 (SD 378.88)

8 Able-bodied Participants



Away Condition



Near Condition

1: Strongly Disagree, 2: Disagree, 3: Neither Agree nor Disagree, 4: Agree, 5: Strongly Agree

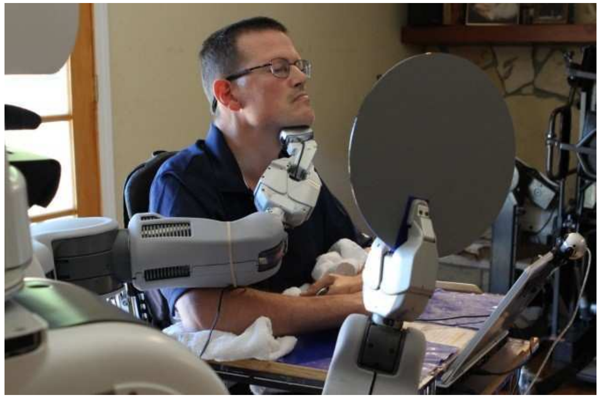
The Healthcare Robotics Lab at Georgia Tech

PI: Charlie Kemp, PhD

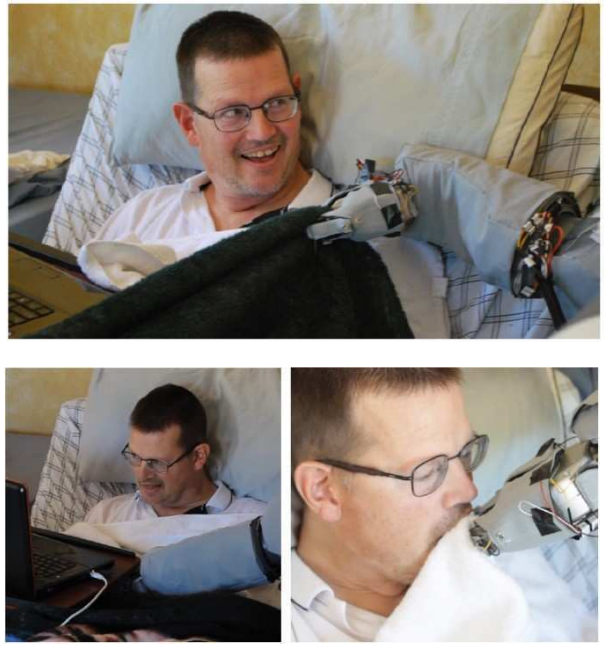
http://healthcare-robotics.com

Mobile robots have the potential to give motor-impaired users greater independence and serve as general-purpose assistive devices that deliver affordable 24/7 personalized care.

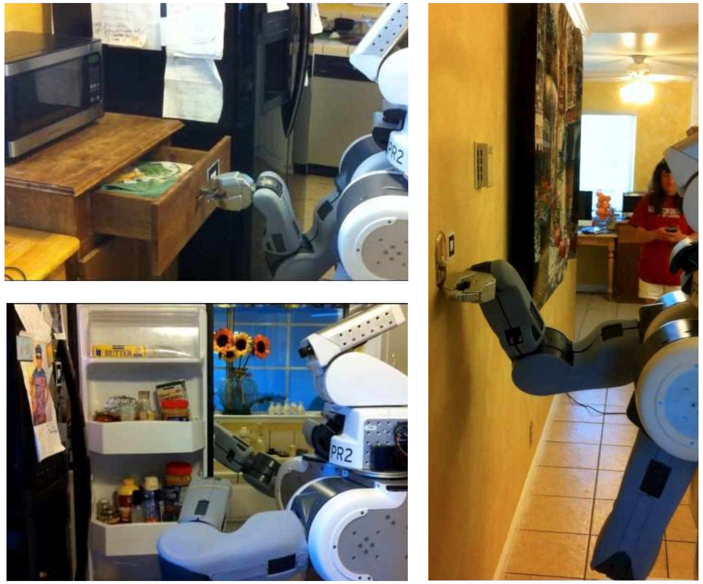
General purpose robot from Willow Garage used in this research.



Henry shaves himself at home using a web-based application for shaving.



Henry pulls up a blanket and wipes his face for himself while in bed at home using a robot with intelligent tactile sensing.



Henry operates devices in his house for himself with autonomous robot actions.

Henry Evans is severely impaired due to a brainstem stroke. He operates the robot using a mouse pointer that he controls using motion of his head and his fingers via an off-the-shelf head tracker and mouse buttons.

(Research performed as part of the collaborative Robots for Humanity project.)