

Proprioceptive Localization for Mobile Manipulators

Mehmet Dogar, Vishal Hemrajani,
Daniel Leeds, Breelyn Kane, and Siddhartha Srinivasa

CMU-RI-TR-10-05

Robotics Institute
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213

February 2010

© 2010 Carnegie Mellon University

Abstract

We use a combination of laser data, measurements of joint angles and torques, and stall information to improve localization on a household robotic platform. Our system executes trajectories to collide with its environment and performs probabilistic updates on a distribution of possible robot positions, ordinarily provided by a laser range finder. We find encouraging results both in simulations and in a real-world kitchen environment. Our analysis also suggests further steps in localization through proprioception.

Acknowledgements

This material is based upon work partially supported by the National Science Foundation under Grant No. EEC-0540865. Mehmet Dogar is partially supported by the Fulbright Science & Technology Program.

Contents

1	Introduction	1
1.1	Background	1
1.2	Related Work	2
2	Approach Overview	2
3	Localization Using Joint Angle Information	3
4	Localization Using Contact Point Information	6
4.1	Using perfect and noisy contact point information	6
4.2	Using joint torques to estimate contact point	7
4.3	Using both joint angles and contact point information	9
5	Robot Experiments	9
6	CONCLUSIONS AND FUTURE WORKS	11
6.1	Conclusions	11
6.2	Future Works	11

List of Figures

1	Proprioceptive localization problem.	1
2	A high level description of our localization process.	3
3	Joint angle aliasing during collision.	4
4	Example result of particle update.	4
5	The position error after using joint angle information to localize.	5
6	The heading error after using joint angle information to localize.	5
7	Contact point aliasing during collision.	6
8	The position error after using contact point information to localize.	7
9	The heading error after using contact point information to localize.	7
10	Example distribution of costs associated with each sampled point on the robot arm.	8
11	The position error after using torque information to localize.	9
12	The heading error after using torque information to localize.	9
13	The position error after using joint angle information, contact point information, and torque information to localize.	10
14	The heading error after using joint angle information, contact point information, and torque information to localize.	10
15	Initial pose estimate, true pose, and filtered pose estimate for two of the experimental runs.	11
16	Initial pose estimate, true pose, and filtered pose estimate with the arm configuration at the time of collision.	11

List of Tables

1	Localization on the Robot	10
---	-------------------------------------	----

1 Introduction

1.1 Background

The Home Exploring Robotic Butler (HERB) at Intel Labs Pittsburgh is a mobile robotic platform designed to give ordinary people assistance with everyday household tasks, such as loading a dishwasher, opening cabinet doors, and moving objects along a counter [1]. The necessary motions for such jobs, performed in a cluttered space, require accurate localization. The robot uses its position and orientation and a map of its environment to plan collision-free trajectories for its arm. Currently, HERB uses 2D laser and odometry localization. However, this approach can perform poorly in typical, highly-dynamic indoor environments (Fig. 1-a), causing unexpected collisions during task execution (Fig. 1-b).

In this paper, we explore the use of proprioceptive collision data to improve robot localization. In a dark room, a human will move her arms to locate nearby walls and furniture. She determines her true location by combining the knowledge of her position relative to these objects with her knowledge of the room's layout and her prior estimate of her absolute position. Similarly, HERB incorporates its inaccurate laser-based estimate from Fig. 1-a with collision information from Fig. 1-b to determine a more accurate position estimate in Fig. 1-c. In general, the laser estimates can be used to produce a cloud of potential robot locations, presented as particles. We use proprioceptive knowledge about HERB's joint angles and the location of contact along HERB's body at the time of each collision to contract the cloud. The location of contact must be estimated from measured torques, while joint angles can be read directly from sensors on the robot. Further collisions can be executed to sharpen our estimates. In this paper, we develop two aspects of localization: estimation of contact location and particle filter updates.



Figure 1: (a) Shows the actual pose of the robot along with a translucent overlay illustrating the estimated localized pose. (b) Due to inaccurate localization, a collision occurs at the illustrated point. (c) Real and estimated robot poses after correction. (d) Task execution with updated pose estimate.

1.2 Related Work

Proprioceptive localization was previously studied in [2]. In this paper, a quadrupedal robot used information about its gait and its terrain to determine its position. Particle filters were used to localize the robot in the terrain. Our approach also uses a particle filter in a constrained state space. However, movements of HERB’s arm to improve localization do not alter the robot’s position, which is determined by its segway base. Beyond the arm’s joint angles, we utilize the contact locations during collisions to increase certainty of HERB’s base position.

[3] addressed methods for determining a 2D contact point on a robotic hand using geometric joint configurations and torque values. A Kalman filter of the hand incorporated position and velocity estimates to predict where the fingertips, or fingerlinks, contacted a rigid object. Similarly, our approach used joint angle and torque values to predict collision contact locations, but in 3D. We incorporate these estimates into our particle filter’s sensor model to localize better the robot base.

The authors in [4] also investigated the use of joint torque information to determine contact location on a 4-DoF WAM arm. Given a set of joint torques, the contact location and the force vector were found analytically. However, this approach had several limitations. At least four torque measurements were needed for the analytical computation; therefore, these computations could not be performed for the first three links (one of which is the upper-arm of the robot). The approach also required the fourth link (the forearm of the robot) to be a line. Furthermore, unique solutions were attainable only for certain configurations of the arm. While we implemented the method in [4] and used it in our initial experiments, these limitations ultimately led us to adopt a different, but related, approach as detailed in Sec. 4.2.

2 Approach Overview

Our approach merges laser data and physical contact information to estimate robot location. HERB uses Adaptive Monte Carlo Localization (AMCL) [5] with laser and odometry measurements as input. A set of weighted random samples represents its belief about its location.

Our system performs further Bayesian updates [6] on these particles using proprioceptive data collected at the time of contact with the environment (Fig. 2). Contacts are detected on the robot when one or more of the joint torques exceeds a certain threshold. In simulation [7], they are detected by checking for spatial overlap between the arm and an object in the environment over the trajectory path.

Particle updates are performed by Bayesian inference. Given a hypothesized location H for a given particle and sensor input as evidence E , we find $P(H|E)$:

$$P(H|E) = \frac{P(E|H)P(H)}{P(E)} \quad (1)$$

where:

- $P(H)$ is the prior probability of H ;
- $P(E|H)$ is the conditional probability of seeing the evidence E given a value for the hypothesis H ;
- $P(E)$ is the marginal probability of E (*i.e.*, the a priori probability of witnessing the new evidence E under all possible hypotheses). The marginal can be computed using $P(E) = \sum P(E|H)P(H)$.

In the physical world, we assume that the recorded collision information (part of E) always is accurate—a binary choice between 1 (collision) and 0 (no collision)—and that HERB’s joint angles at the time of the collision are known, albeit with some error. However, the point of contact for collision may be uncertain. Below, we address two scenarios regarding this

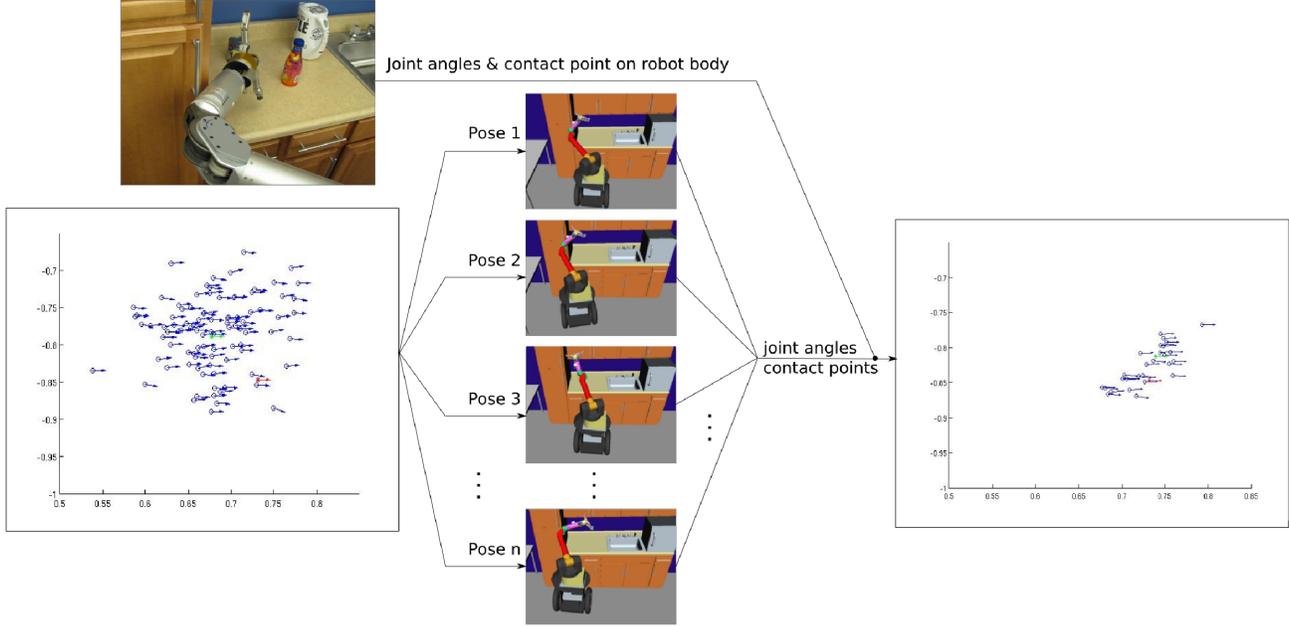


Figure 2: A high level description of our localization process. The particles in the initial distribution are assigned weights and resampled according to how closely they predict the joint angles and the contact point of the actual collision. In the image the blue particles are coming from the 2D laser localization module, the green particle is the mean of this distribution (the best guess), and the red particle shows the ground truth of actual robot pose.

information. The first assumes that the robot has tactile sensors distributed over the surface of the arm that provide the coordinates of the point of contact and magnitude of the force applied. The second calculates the probable values for the same data by using the joint torque values at the time of collision.

We use the observation data obtained from the simulation and that of the true arm when a collision occurs to compute $P(E|H)$. By simulating the trajectory for each particle in the sample set, we can determine the arm position for the hypothesized robot location—in simulation, as in the real world, the arm stops moving when a collision occurs. $P(E|H)$ for the joint angle observations is equal to the corresponding value of a Gaussian distribution about the true obtained joint angle position. $P(E|H)$ for the contact points are computed from a Gaussian distribution around the point of contact, if full tactile sensing is available. Otherwise, we obtain a discrete approximation to the probability distribution of the contact point on the surface of the arm using torque measurements, and use this to compute $P(E|H)$ (See Sec. 4.2). $P(H|E)$ updates can be performed using only the joint angle observation or only contact point observations, or using both observations by applying Bayes’ rule twice in succession.

3 Localization Using Joint Angle Information

We first use joint angles alone for robot localization. This approach successfully narrows the number of potential poses we consider viable, although fundamental geometric properties prevent decreasing the range of locations and headings even further without more intelligent trajectory planning.

Joint angles specify the configuration of the arm at the time of collision. However, it is important to note joint angles at the time of contact do not uniquely determine robot location. Fig. 3-b shows distinct robot base poses that produce identical arm configurations at collision time given the same trajectory. The results of a sample simulation run of particle filtering

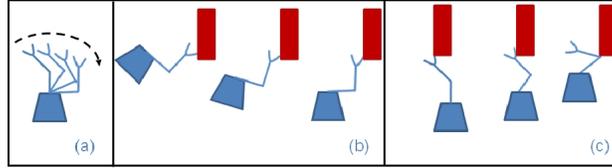


Figure 3: (a) A trajectory for the arm. (b) The joint angles at the moment of collision can be the same even when the robot pose is different and it is executing the same trajectory. (c) Assuming that the rotation of the robot is fixed when colliding with a plane, joint angles determine the position of the robot uniquely along the axis parallel to the plane normal.

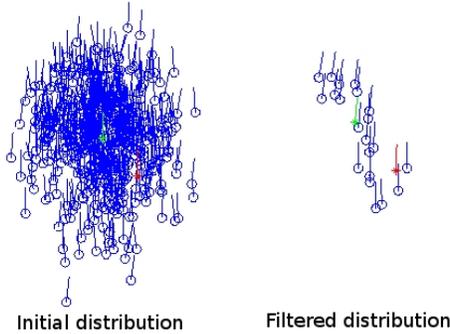


Figure 4: Example result of particle update. In this case the simulated robot arm collides with the corner of a kitchen cabinet, and the joint angle information is used to improve localization. The blue circles represent the position of the robot for each particle, while the lines represent its heading. The red particle represents the “ground truth”, *i.e.*, the robot’s true position. The green particle is the mean of the blue particles, representing the best guess for that distribution. After the particle update, a set of particles that all give (approximately) the same joint angle values remains. Fig. 3-b illustrates that the collision joint angles stay the same as the particles move up and left, and rotate to the right, which is the behavior shown by the filtered distribution here.

provide a similar perspective in Fig. 4. The initial distribution of particles is updated after executing a trajectory that leads to a collision with the corner of a kitchen cabinet, as in Fig. 3-b. The particles remaining after filtering are consistent with the recorded joint angles at collision time. However, the error between the estimated and the true position has not decreased.

We performed a set of simulation experiments in which we used joint angle information to localize our robot. For these experiments, we generated an initial Gaussian distribution of particles around a random point in front of the kitchen counter. In generating these particles we used the standard deviation of 3 cm in x and y , and 0.1 radians for the heading. We drew a random point from the same distribution as the “actual” position of our robot, found a trajectory for that pose that would bring the robot arm into contact with the environment, and simulated the trajectory for all particles. Figs. 5 and 6 present the localization errors after executing two such trajectories in twenty random runs. To narrow the particle cloud, we first assumed that we knew the joint angles at the time of collision without any noise, and then we repeated the localization by adding Gaussian noise of standard deviation 0.02 radians over each of the joint angle measurements. As expected, the updates lead to a significant decrease in the error, but even with noiseless knowledge of joint angles the robot still has a certain amount of uncertainty regarding its location. Noisy measurements leads to lower improvements than measurements without noise.

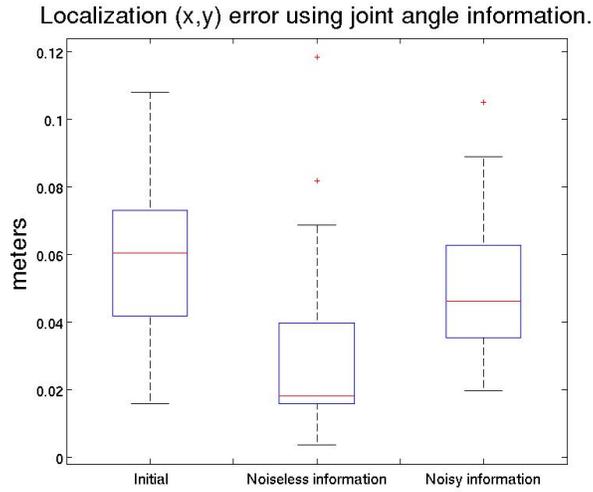


Figure 5: The position error after using joint angle information to localize. The error for a given distribution is found by taking the mean position of the particles in that distribution and computing the distance to the actual robot position. This figure shows the distribution of this error given the initial particle distribution, given the particles after filtering using joint angle information without noise, and given the particles after filtering using joint angle information with noise. These are the results of 20 random trials in simulation.

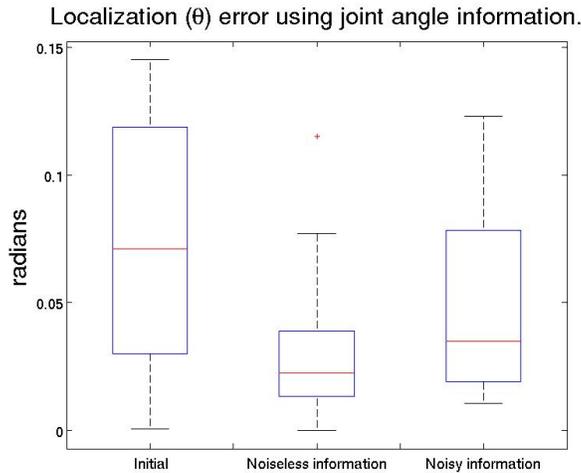


Figure 6: The heading error after using joint angle information to localize. The error for a given distribution is found by taking the mean heading direction of the particles in that distribution and computing its difference with the actual robot heading. This figure shows the distribution of this error given the initial particle distribution, given the particles after filtering using joint angle information without noise, and given the particles after filtering using joint angle information with noise. These are the results of 20 random trials in simulation.

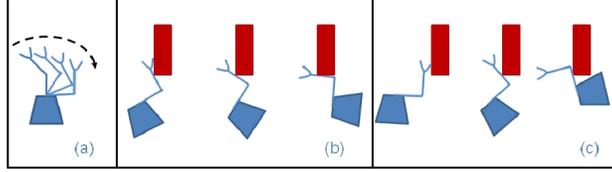


Figure 7: (a) A trajectory for the arm. (b) The contact point on the robot body at the moment of collision can be the same even when the robot pose is different and it is executing the same trajectory. (c) Example cases in which the contact point supplies some information to differentiate between different robot poses.

4 Localization Using Contact Point Information

We use the contact point on the robot body to further localize our robot. We consider three cases:

- Assuming a perfectly accurate tactile sensor giving the exact contact point on the body.
- Assuming a noisy tactile sensor giving an approximate contact point on the body.
- Assuming no tactile sensors and trying to infer the contact point from the joint torque values.

We study the case of the “perfect tactile sensor” to learn the limits of using contact point information to localize. We study noisy contact point sensing to account for real-world limits of any robot with tactile sensors. We study the case without tactile sensors to model most of today’s robots, including our robot, which do not have a continuous sensitive skin over its body to supply contact information. The joint torques in this case can supply some information about the location of the contact point.

Assuming that the robot can perfectly sense the contact point, we again note a given contact point on the body does not uniquely determine a pose for the robot. Fig. 7 illustrates that the same point on the body can make contact with the environment when executing the same trajectory even if the robot pose changes. Perfect knowledge of the contact point on a robot’s body is insufficient for perfect localization. Again, the use of intelligent planning of multiple trajectories may circumvent this problem.

4.1 Using perfect and noisy contact point information

Figs. 8 and 9 present the results of our simulated experiments for the case of assuming perfect contact point information and noisy contact point information. We again used an experimental setup where we generated an initial Gaussian distribution of particles around a random point in front of the kitchen counter (with the standard deviation of 3 cm in x and y , and 0.1 radians for the heading). We drew a random point from the same distribution as the “actual” position of our robot, found a trajectory for that pose that would bring the robot arm into contact with the environment, and simulated the trajectory for all particles. We then used the contact point on the robot arm to localize our robot. In assigning probabilities to our particles, we used the distance between the contact points predicted by each particle and the contact point of the “actual” robot. In the experiments where we used noisy contact point information, we added a Gaussian noise with a standard deviation of 1 cm in each direction. We see that the contact point localization results in a decrease in the error, but uncertainty about robot location remains. Again, noisy measurements leads to lower improvements than measurements without noise.

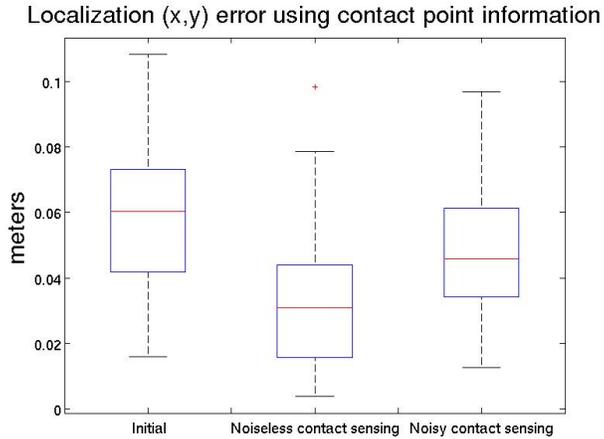


Figure 8: The position error after using contact point information to localize. The error for a given distribution is found by taking the mean position of the particles in that distribution and computing the distance to the actual robot position. This figure shows the distribution of this error given the initial particle distribution, given the particles after filtering using contact point information without noise, and given the particles after filtering using contact point information with noise. These are the results of 20 random trials in simulation.

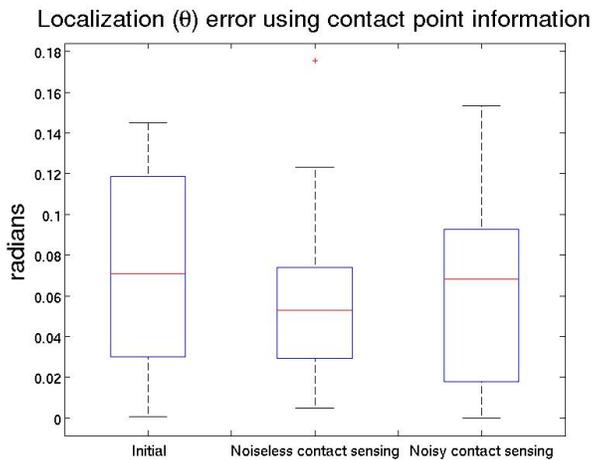


Figure 9: The heading error after using contact point information to localize. The error for a given distribution is found by taking the mean heading direction of the particles in that distribution and computing the difference to the actual robot heading. This figure shows the distribution of this error given the initial particle distribution, given the particles after filtering using contact point information without noise, and given the particles after filtering using contact point information with noise. These are the results of 20 random trials in simulation.

4.2 Using joint torques to estimate contact point

HERB and most robots lack a skin to supply exact contact point information. Therefore, we investigated approaches to estimate this information. Note we assume that our robot has a single point of contact with the environment; it is unlikely to hit two different points with two different parts of its arm at the same time while executing a trajectory.

One way to estimate the contact point is to use joint torques generated by the force acting at the contact point. We sample points over the robot arm and compute the torques we would see if the contact occurred at each of those points. We compare

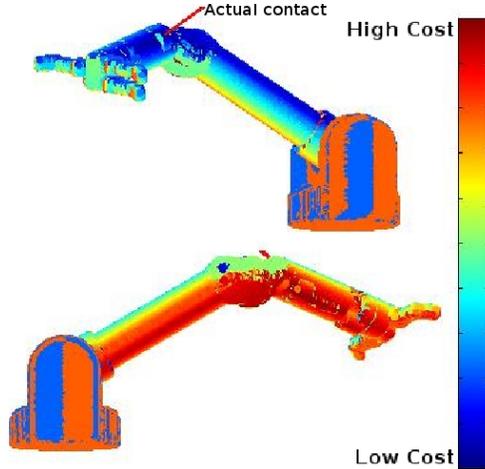


Figure 10: Example distribution of costs associated with each sampled point on the robot arm.

each of these torques with the actual torque values from the robot and assign a cost to the sampled points depending on the difference between the two torque vectors. This approach requires us to define appropriate force vectors to be applied at the contact location.

We set force directions equal to the normal to the surface at the contact point. This choice reflects the fact the robot has a low friction surface, making it unlikely to experience forces deviating significantly from the normal.

Unfortunately, we have no way to know the actual magnitude of the force applied to the contact. Therefore, we use a cost function invariant to the force magnitude. The set of torques for a given force is given by the Jacobian transpose [8]:

$$\tau = J^T * F \quad (2)$$

Any change in the magnitude of the vector F only scales the resulting T . Hence, if we use a magnitude of F on the sampled points that is different from the magnitude of the force at the actual contact point, the direction of the vector T is valid nonetheless. To compare two different sets of torques resulting from the applications of forces on different parts of the robot body, we use the directional difference of the torque vectors, ignoring the torque magnitudes. The directional difference between two torque vectors τ_1 and τ_2 is given by:

$$Cost(\tau_1, \tau_2) = \cos^{-1}\left(\frac{\tau_1 \cdot \tau_2}{|\tau_1||\tau_2|}\right) \quad (3)$$

We generate a cost map (such as the one in Fig. 10) by sampling points (and their normals) over the surface of the robot arm, simulating a unit force applied at these points, finding the corresponding torque vectors, and computing the cost associated with each point using Eqn. 3.

The cost distribution strongly discriminates between most of the sampled points on the arm and the actual contact point, assigning high costs to the former points. However, there are regions along the arm that the cost function cannot discriminate clearly, especially those that have a normal similar to that of the contact point and are on the same link with it.

In the previous section, we showed that a perfect knowledge of the contact point produces a range of valid robot poses. Using the torque information to estimate the contact point provides its own range of low-cost surfaces. Thus, robot localization using torque information will perform worse than localization assuming exact knowledge of the contact point. Figs. 11 and 12 show the results of our simulated runs. There remains minor improvement in localization given perfect sensing of joint torques. However, adding noise over the torque information further obscures this improvement.

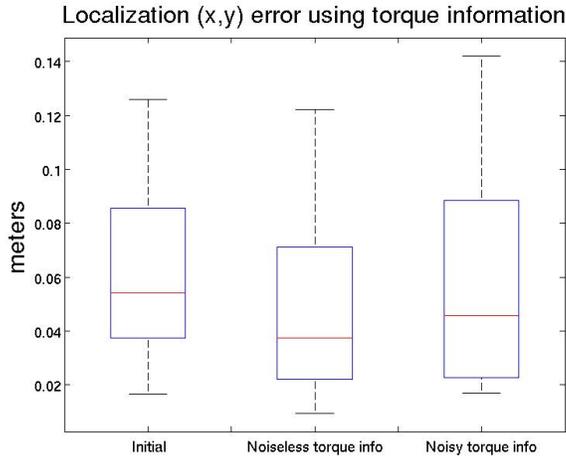


Figure 11: The position error after using torque information to localize. These are the results of 20 random trials in simulation.

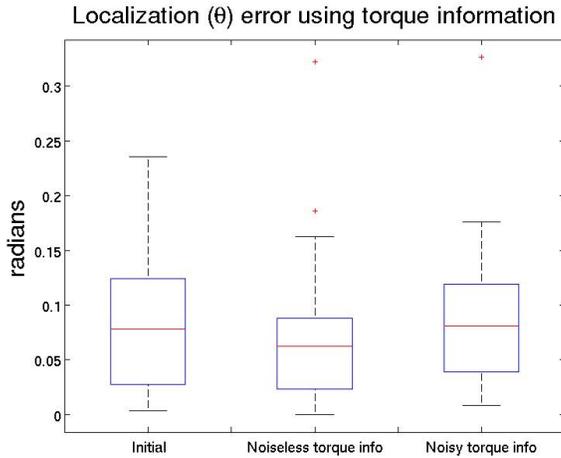


Figure 12: The heading error after using torque information to localize. These are the results of 20 random trials in simulation.

4.3 Using both joint angles and contact point information

Lastly, we use both the joint angle information and the contact point information to localize our robot. Figs. 13 and 14 show use of joint angle information and contact point information results in a better localization than does use of joint angles alone. However, combining joint angles and torque information provides little, if any, localization improvements.

5 Robot Experiments

Experiments were conducted on the robot using only joint angles without using joint torques. The joint torque values were too noisy to produce reasonable improvements in localization. These results are shown in Table. 1 , they show the initial error values in the position and the heading direction and the error values after filtering for four runs using different

Localization (x,y) error using joint angles and contact point

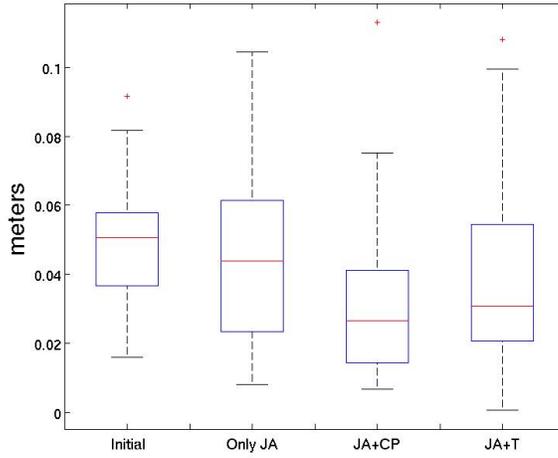


Figure 13: The position error after using joint angle information (JA), contact point information (CP), and torque information (T) to localize. These are the results of 20 random trials in simulation.

Localization (θ) error using joint angles and contact point

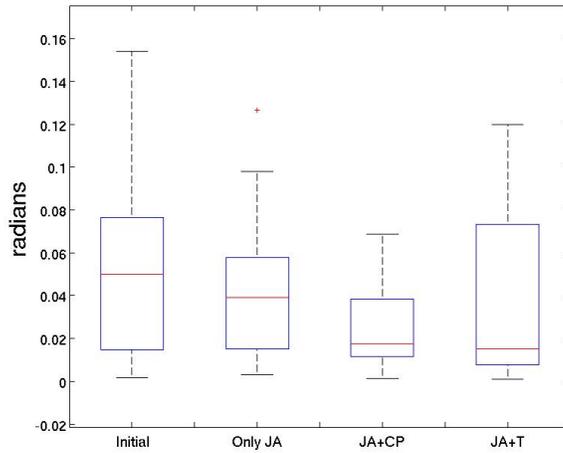


Figure 14: The heading error after using joint angle information (JA), contact point information (CP), and torque information (T) to localize. These are the results of 20 random trials in simulation.

Table 1: Localization on the Robot

	Initial pose mean error	Filtered pose mean error
Position (x,y)	0.0543m	0.0291m
Heading (θ)	0.1139 radians	0.0874



Figure 15: Initial pose estimate, true pose, and filtered pose estimate for two of the experimental runs.

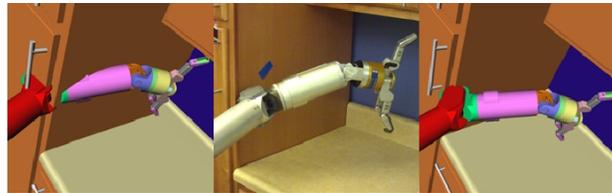


Figure 16: Initial pose estimate, true pose, and filtered pose estimate with the arm configuration at the time of collision.

trajectories. Fig. 15 shows the initial estimate of the robot’s pose, its actual pose at the time of collision and the pose estimate after filtering for two of the runs. We can see that the pose estimate improves after filtering and is closer to the true pose of the real robot in the world. Fig. 16 shows the robot arm configuration at the time of collision, for the initial, the final and the filtered pose.

6 CONCLUSIONS AND FUTURE WORKS

6.1 Conclusions

We demonstrated proprioceptive and collision data can sharpen robot localization estimates from a large particle cloud to a small set of likely positions. Different types of proprioceptive information are also useful in localizing the base’s position. The use of torque estimations combined with joint angles contribute to position certainty rather than solely utilizing one of these techniques for feedback.

6.2 Future Works

Although the use of proprioceptive data can increase a robot’s self awareness with the external environment, more information is necessary to improve localization. Our work can be expanded in several directions:

- Our approach could include a more thorough planning scheme. Paths to distinct surfaces (especially those with distinct orientations) would provide much richer localization information. We can perform future work on selecting

trajectories that best explore the robot's space. In this way, an active localization takes place similar to how a human might actively touch objects in a dark room upon initial entry.

- Our system utilizes geometric and force based stall information across the whole robot. We can apply similar particle update techniques to data coming from the pressure sensors on HERB's finger tips. These sensors are low resolution and tend to overflow; however, they may be useful for binary collision detection at the end effectors.

References

- [1] Siddhartha S. Srinivasa, Dave Ferguson, Casey J. Helfrich, Dmitry Berenson, Alvaro Collet, Rosen Diankov, Garratt Gallagher, Geoffrey Hollinger, James Kuffner, and Michael Vande Weghe. Herb: A home exploring robotic butler. *Autonomous Robots*, 28:5–20, 2010.
- [2] Sachin Chitta, Paul Vernaza, and Daniel Lee. Proprioceptive localization for a quadrupedal robot on known terrain. In *Proc. of IEEE International Conference on Robotics and Automation, 2007.*, 2007.
- [3] M. Huber and R. A. Grupen. 2D Contact Detection and Localization: Using Proprioceptive Information. *IEEE Transactions on Robotics and Automation*, 1(10):23–33, 1994.
- [4] Brian S. Eberman and John Kenneth Salisbury Jr. Determination of manipulator contact information from joint torque measurements. In *ISER*, pages 463–473, 1989.
- [5] C. Kwok, D. Fox, and M. Meila. Adaptive real-time particle filters for robot localization. In *IEEE International Conference on Robotics and Automation, 2003. Proceedings. ICRA'03*, volume 2, 2003.
- [6] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. The MIT Press, September 2005.
- [7] Rosen Diankov and James Kuffner. OpenRAVE: A Planning Architecture for Autonomous Robotics. Technical report, The Robotics Institute, Carnegie Mellon University, July 2008.
- [8] John J. Craig. *Introduction to Robotics: Mechanics and Control*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1989.