

Learning to Slide a Magnetic Card Through a Card Reader

Vladimir Sukhoy, Veselin Georgiev, Todd Wegter, Ramy Sweidan, and Alexander Stoytchev
Developmental Robotics Laboratory
Iowa State University

{sukhoy, georgiev, twegter, rsweidan, alexs}@iastate.edu

Abstract—This paper describes a set of experiments in which an upper-torso humanoid robot learned to slide a card through a card reader. The small size and the flexibility of the card presented a number of manipulation challenges for the robot. First, because most of the card is occluded by the card reader and the robot’s hand during the sliding process, visual feedback is useless for this task. Second, because the card bends easily, it is difficult to distinguish between bending and hitting an obstacle in order to correct the sliding trajectory. To solve these manipulation challenges this paper proposes a method for constraint detection that uses only proprioceptive data. The method uses dynamic joint torque thresholds that are calibrated using the robot’s movements in free space. The experimental results show that using this method, the robot can detect when the movement of the card is constrained and modify the sliding trajectory in real time, which makes solving this task possible.

I. INTRODUCTION

Magnetic cards are widely used in human environments. They are small, thin, and light, which makes it possible for us to carry many of them in our wallets. In recent years they have become the preferred method of controlling door access in office buildings and hotel rooms. They are also used as the preferred method of payment at grocery stores, shopping malls, gas stations, parking lots, and vending machines. In some cases they are used to gain access to computer terminals and instrument panels.

Magnetic cards present a number of manipulation challenges. They are small, which makes them hard to grasp. They are flexible, which increases the difficulty of insertion and sliding tasks. They are symmetrical in shape, but they are not symmetrical in function, which often requires reorienting and re-grasping them. In other words, some of the same reasons why plastic cards are so popular are also the reasons why they are difficult to manipulate, especially for robots.

Card readers present a number of manipulation challenges as well. The location where the card must be inserted is usually not much thicker than the card itself. Once the card is inserted it must stay within the gap and must remain flush with the bottom of the card reader. It must travel a certain distance so that the entire magnetic strip passes over the reading head. Furthermore, it must be moved at a certain speed that is neither too slow nor too fast and without stopping before the entire movement is completed. The bending of the card should be minimized, otherwise it may not be read properly or it may pop out of the card reader. Finally, the card must be oriented properly with respect to

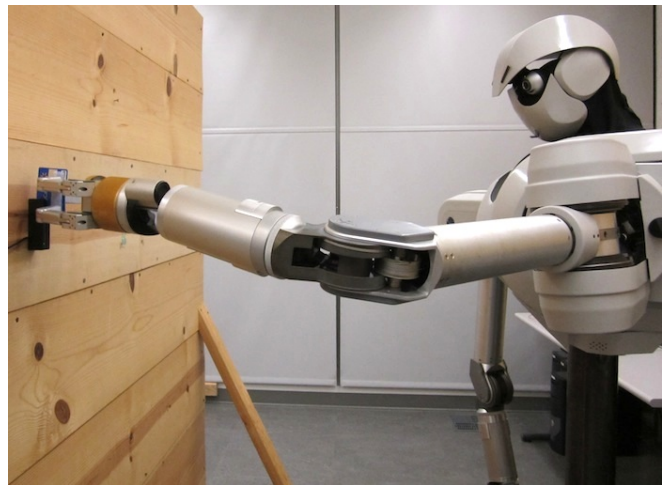


Fig. 1. The upper-torso humanoid robot that was used in the experiments, shown here sliding a card through a card reader.

the card reader as it can be read only if it is inserted in a certain orientation.

There are multiple reasons why robots should be able to use magnetic cards. The most pressing one is that there is currently a big push in the robotics community to design robots that can co-exist with humans in human environments. By default, the widespread use of magnetic cards in our society will dictate the need to train robots to use them as well. While it may be possible to design alternative cardless solutions for robots, this would not be feasible for all tasks as it would require a massive investment to change the existing infrastructure that relies on cards and card readers. A better approach would be to teach the robots how to use plastic cards in order to accomplish many practical tasks.

In our approach, we use proprioceptive feedback to detect when the movements of the robot become constrained. Because magnetic cards are flexible, the mechanism for detecting constrained movement must be sufficiently sensitive to prevent bending the card too much. Achieving this level of sensitivity using static joint torque thresholds can be very difficult. Instead, we implemented a learning approach that employs dynamic joint torque thresholds that the robot learns from its own experience. To the best of our knowledge, this is the first study that tackles the problem of autonomous sliding of a magnetic card by a robot.

II. RELATED WORK

The ability to manipulate objects in constrained spaces is crucial for solving insertion tasks, which are often called “peg-in-the-hole” tasks [1, Chapter 26]. Existing approaches for solving these tasks can be grouped into three categories: 1) visual servoing, 2) topological contact modeling, and 3) controllers based on proprioceptive feedback.

The approaches in the first category focus on detecting the holes using computer vision [2] [3] [4] [5] [6] [7]. In many cases the proposed application is plugging into a wall socket autonomously [3] [4] [5] [6] [7]. One characteristic feature of these approaches is that the manipulation task of actually inserting the peg into a hole is considered easy if the hole can be sufficiently well localized in the image. One of the approaches for autonomous plugging required installing a separate camera in the robot’s wrist and putting a checkerboard pattern on the plug to improve the performance of the visual algorithms [3]. In other cases, trial and error learning may be required to compensate for possible tracking inaccuracies [8]. Some approaches even propose to use new task-specific modalities to localize the holes better. For instance, an electromagnetic radiation sensor was proposed to detect the location of the electric socket [4].

Bruyninckx et al. [9] demonstrated that insertion tasks can be solved using two different types of approaches: 1) approaches based on kinematic models of the goal location and constraints; and 2) approaches based on geometrically precise alignment of the peg with respect to the hole.

This paper builds on our previous work [10] that showed that robots can use exploratory behaviors and proprioceptive feedback to solve insertion tasks. In particular, it was shown how the robot can solve a shape-sorter puzzle with three different blocks and holes [10]. It was also shown that joint torques recorded before and after insertions were significantly different [10]. The ability to detect these differences autonomously was left for future work.

This paper shows that differences in joint torques can be used not only to detect constrained movement of a peg in a hole, but also to detect other types of constrained movement, e.g., rubbing of a plastic card against the wall of a card reader. This paper shows that constrained movement can be detected in real time. This sets it apart from our prior work in which the difference in joint torques was detected offline [10].

Suarez et al. [11] proposed an approach for performing insertion tasks without the use of geometric models, as those are not always available in an accurate form. To do this, they modeled the many uncertainties in the robot’s environment and used force feedback information to guide the robot’s fine movements through several task states and eventually into the goal state. Paetsch and his colleagues [12] circumvented the problem of uncertainties by programming into their robot a number of strategies modeled after human strategies for inserting a peg into a hole. Their robot used the readings from the force feedback sensors to decide which strategies to use. Paetsch’s results showed that these strategies, when



Fig. 2. The card and the card reader used in the experiments: (a) front of the card; (b) back of the card; (c) the card reader mounted on the experimental fixture. A strip of Velcro was attached to both sides of the card to increase the friction between the card and the robot’s fingers.

used in tandem with each other, increased the success rate of the task.

Infants have the ability to consistently insert objects into holes at approximately 22 months of age [13]. Infant chimpanzees, which mirror the developmental cycle of humans, exhibit the insertion behaviors even earlier at 10 months [14]. These skills develop slowly over time and go through several stages before they can be fully useful. At 15 months, human infants are able to perform insertion tasks, but they perform at chance level [15]. This shows that as motor skills develop and cognitive faculties mature, insertion tasks become easier due to increased spatial reasoning and coordination.

III. EXPERIMENTAL SETUP

A. Robot

All experiments were performed using the upper-torso humanoid robot shown in Fig. 1. The robot’s arms are two Barrett Whole Arm Manipulators (WAMs), which are controlled at 500Hz from a real time Linux PC. Only the left WAM was used in the experiments, which has a three-finger Barrett Hand (model BH8-262) as its end effector.

B. Card and Card Reader

Figure 2 shows the card that was used in the experiments. The physical characteristics of the card are defined by the ISO 7810 standard, which fixes the dimensions of credit cards to $85.60\text{ mm} \times 53.98\text{ mm}$ [16]. The thickness of the card, which is defined by the ISO 7813 standard, is 0.76 mm [16].

A commercially available card reader (a generic 3 track USB MSR Magnetic Stripe Credit/Smart Card Swipe POS Programmable Mini Reader) was used in the experiments. It was purchased from amazon.com. The dimensions of the reader are $90\text{ mm} \times 26\text{ mm} \times 27\text{ mm}$. The reader supports bi-directional reading, but the magnetic strip can be read in only one orientation.

The card reader was mounted on a fixture in front of the robot, with the card slot in a vertical orientation (see Figure 2(c)). The card reader was connected to a computer through a USB cable so that successful slides of the card could be recorded. A slide was considered successful only if the entire message encoded on the magnetic strip was read.

C. Grasping the Card

The card is quite small relative to the robot’s fingers, which are twice as thick as human fingers. It slips out of grasp easily

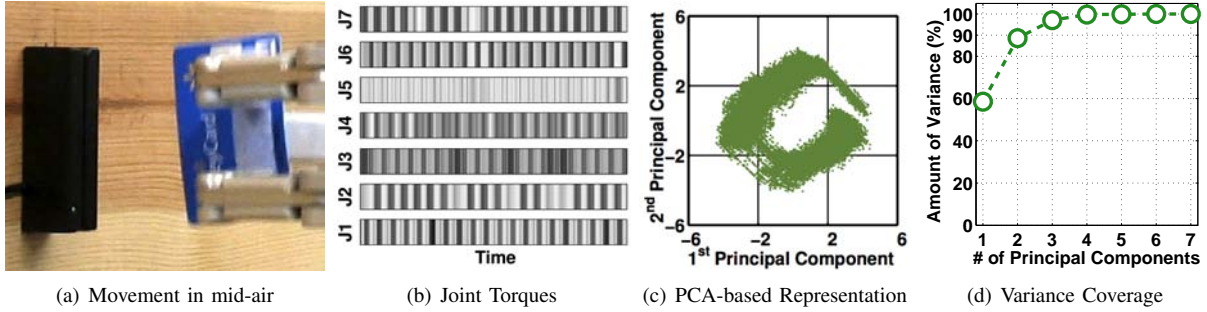


Fig. 3. Illustration of the process of building the background model for detecting constrained movement: (a) the card reader is moved away when the robot performs card swiping behaviors in mid-air; (b) joint torques recorded during these movements in free space; (c) a representation of these joint torques in the space spanned by the first two principal components; and (d) the amount of variance covered by different principal components.

because the fingers are made of brushed aluminum and are not well suited for this task. Furthermore, the morphology of the hand allows for only one viable grasp configuration as shown in Fig. 1. In addition to occluding most of the card, this configuration tends to bend the card into an U-shape, which makes it impossible to slide it through the card reader. To resolve these difficulties, while still focusing on the main goal of this paper (i.e., detecting movement constraints in constrained spaces while holding flexible objects) we added a layer of Velcro to each side of the card and to the fingertips of the robot (see Fig. 2(a) and Fig. 2(b)). This allowed the robot to hold the card firmly for long periods of time without bending it too much. All experiments started with the robot already grasping the card.

IV. DETECTING CONSTRAINTS

The key challenge in the card reading task is to avoid bending the card too much as it is swiped through the card reader. If the robot cannot correct the swiping movement to prevent excessive bending, then it can force the card out of the card reader or even permanently deform the card. To solve these problems the robot can try to detect when its movements are constrained. If the robot has this ability, then it can stop moving as soon as it detects the constraint. Furthermore, if the robot can detect which movement trajectories are constrained, then it can use this information to improve its future attempts at solving this task.

In this work, the robot detected constraints using the assumption that they make the proprioceptive sensations in the arm differ from the sensations of moving in free space. In other words, if the currently observed joint torques deviate from a background model of moving the arm in mid-air, then the current movement must be constrained. This allowed the robot to detect constraints exclusively through proprioceptive feedback.

There are multiple other ways in which constraint detection can be accomplished. We believe that our approach is more reliable than methods based on visual movement detectors. In this particular task, the card and the card reader are very small. The resolution of the cameras may not be sufficient to reliably detect the movements of the card. Furthermore, the bottom part of the card that has to be inside

the reader is not visible during sliding, so a decision would have to be made as to what to track. Some options include tracking the robot’s fingers and the top part of the card, if that part is not occluded by the hand. These options, however, may not be reliable due to the flexible nature of the card. As the card flexes, the part of the card inside the card reader can be fixed, but the part of the card grasped by the robot can be moving along with the robot’s hand perpendicularly to the sliding direction.

A. Building a Background Calibration Model

The background calibration model is the model of joint torques recorded during movements in free space. The model uses principal component analysis to reduce the dimensionality of the raw data. The joint torques are discretized in the space spanned by the first two principal components and the resulting point cloud is used to represent the distribution of the joint torques during unconstrained movement. When the robot is interacting with the card reader, it detects constrained movement if the new joint torques fall outside of this distribution.

To build the background model the robot performs a number of sliding movements in mid-air with the card in its hand. The movements are performed in the same spatial region where the card reader is placed, but the card reader is removed when the robot learns the model. The background model is built using joint torques recorded during these movements. The model includes: 1) a vector $M \in \mathbb{R}^7$ where each element is the mean torque for one of the seven WAM joints; 2) two vectors $P^{(1)} \in \mathbb{R}^7$ and $P^{(2)} \in \mathbb{R}^7$ with the coefficients for the first two principal components of the joint torques; and 3) a set of integer pairs S that represent observed joint torque values in the space spanned by the first two principal components.

More specifically, the background model is built from a matrix of joint torques $T \in \mathbb{R}^{n \times 7}$, where n is the number of joint torque vectors recorded while the robot was moving the card in mid-air (in the experiment $n = 37790$, which corresponds to $37790 \text{ WAM heartbeats} / 500 \text{ Hz} = 75.58 \text{ seconds}$). Thus, T_{ij} is the torque recorded during the i^{th} timestep for the j^{th} joint. The vector M contains the mean

torque for each joint, i.e.,

$$M_j = \sum_{i=1}^n \frac{T_{ij}}{n}, \quad j = 1, \dots, 7. \quad (1)$$

The vectors $P^{(1)} \in \mathbb{R}^7$ and $P^{(2)} \in \mathbb{R}^7$, which contain coefficients for the first two principal components of T , can be calculated using singular value decomposition (SVD) [17]:

$$\text{cov}(T) = U\Sigma U^T, \quad (\text{i.e., SVD of } \text{cov}(T)), \quad (2)$$

$$P^{(1)} = [U_{11}, U_{21}, U_{31}, U_{41}, U_{51}, U_{61}, U_{71}]^T, \quad (3)$$

$$P^{(2)} = [U_{12}, U_{22}, U_{32}, U_{42}, U_{52}, U_{62}, U_{72}]^T, \quad (4)$$

where $\text{cov}(T) \in \mathbb{R}^{7 \times 7}$ is the covariance matrix of T . Fig. 3(d) shows that the first two components account for 88.6% of the variance in the experiment, which justifies using only these two principal components to make the representation more compact.

A discretization of the point cloud formed by T in the space spanned by $P^{(1)}$ and $P^{(2)}$ can be represented by a set of integer pairs S . More formally,

$$S = \{(X_i, Y_i)\}, \quad i = 1, \dots, n, \quad (5)$$

where the integers $X_i \in \mathbb{Z}$ and $Y_i \in \mathbb{Z}$ are defined as shown below:

$$X_i = \text{round} \left(\delta \sum_{j=1}^7 (T_{ij} - M_j) \cdot P_j^{(1)} \right), \quad (6)$$

$$Y_i = \text{round} \left(\delta \sum_{j=1}^7 (T_{ij} - M_j) \cdot P_j^{(2)} \right), \quad (7)$$

for $i = 1, \dots, n$. The parameter δ , which controls the sensitivity of the discretization, was set to the empirically derived value of $1/4$. Figure 3(c) shows the point cloud and the discretization for one of the experiments in which the robot had to insert the card into the top end of the card reader slot and slide it down (see Section VI-B for more information about the results of this experiment).

Computational speed was one of the main reasons for choosing a model based on discretization in PCA space. Using this model, the calculations for constrained movement detection can be performed in real time. As the robot receives each new joint torque measurement it only needs to perform a few arithmetic and memory access operations to represent this measurement in PCA space. The robot also has to check if a similar set of joint torques were encountered during movements in mid-air. The total number of operations for completing both goals can be bounded by a constant.

B. Real-Time Detection of Constrained Movement

As the robot attempts to swipe the card, it can detect constrained movement for each joint torque vector $\tau \in \mathbb{R}^7$ that it gets from the torque sensors in the WAM. A pair of integers $(x(\tau), y(\tau))$ can be computed for the vector τ like in formulas (6) and (7) that were used to build the background model. If the pair $(x(\tau), y(\tau))$ is not in S , then the robot

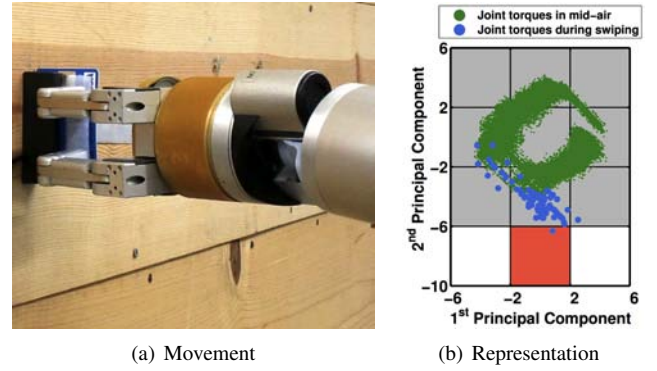


Fig. 4. Example of constrained movement detection: (a) picture taken during a movement performed by the robot when it became constrained, and (b) the state of the representation when the constraint was detected. One of the joint torque measurements fell into a region (red rectangle) that was not visited during movement in mid-air, which caused the detector to fire. See text for more details.

was constrained when τ was recorded. Otherwise, the robot was not constrained at that moment.

More formally, the integers $x(\tau)$ and $y(\tau)$ are computed as shown below:

$$x(\tau) = \text{round} \left(\delta \sum_{j=1}^7 (\tau_j - M_j) \cdot P_j^{(1)} \right), \quad (8)$$

$$y(\tau) = \text{round} \left(\delta \sum_{j=1}^7 (\tau_j - M_j) \cdot P_j^{(2)} \right). \quad (9)$$

The constrained movement detector is formally specified using the following expression:

$$\text{MoveConstrained}(\tau) = \begin{cases} \text{Yes,} & \text{if } (x(\tau), y(\tau)) \notin S; \\ \text{No,} & \text{if } (x(\tau), y(\tau)) \in S. \end{cases} \quad (10)$$

An example of constrained movement detection during card swiping is shown in Fig. 4. When the background model was built, only joint torques with coordinates in PCA space between -6 and 6 were observed. Because the discretization sensitivity δ was set to $1/4$, all integer pairs (X_i, Y_i) lie in the set $S = \{-1, 0, 1\} \times \{-1, 0, 1\}$. When the robot was later performing the card swiping behavior, it recorded one joint torque vector τ with coordinates in PCA space equal to $(0.83, -6.3)$. Thus, $x(\tau) = \text{round}(0.83/4) = 0$ and $y(\tau) = \text{round}(-6.3/4) = -2$. Constrained movement was detected because $(x(\tau), y(\tau)) \notin S$.

V. METHODOLOGY

This section describes how the robot performed the card swiping behavioral sequences. Each sequence had three phases: 1) move to a starting position near the card reader, 2) insert the card into the gap of the card reader, and 3) slide the card through the gap. For each of the three manipulation phases the WAM performed one movement. The destination for each of these movements was sampled from an area defined by a set of four points in joint space. The first phase used four points for which the card was about 10–15 cm

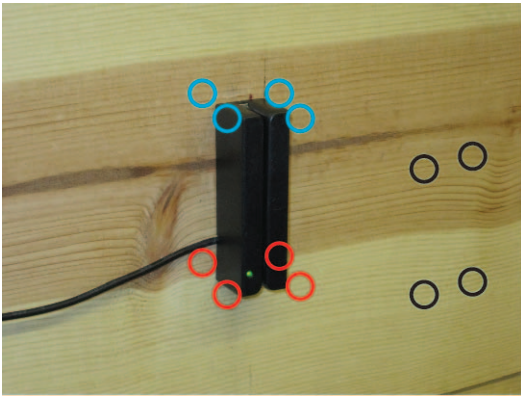


Fig. 5. The three sets of four anchor points that were used to sample the destination positions of the robot’s movements during a card sliding sequence. A human experimenter backdrove the robot’s arm until the middle of the bottom part of the card coincided with each point. The twelve vectors of joint angles recorded from the WAM in these locations were stored and used to generate candidate movements. See text for more details.

away from the board and formed a rectangle of roughly the same size as the card reader. The two remaining phases used two sets of four points for which the card was about 1 cm away from the four corners of the two opposite sides of the card reader. These locations were used to specify roughly where each movement in a manipulation sequence should finish, but the robot still had to select the exact destinations for each of the three movements in a sequence. The approximate locations of the twelve anchor points are shown in Fig. 5.

To avoid moving the card into obstacles, the constrained movement detection procedure described in Section IV was used. Because the constraint detector can be updated in real time, it was able to stop those movements immediately if formula (10) indicated a constraint. Because the robot records joint torques at 500 Hz, the reaction time of this approach is sufficient to prevent bending the card.

In order to achieve high success rates in this task, it is also necessary to use the feedback from the card reader. Even if the robot completes each of the three manipulation phases without being constrained, the entire sequence can fail because the card was inserted outside of the card reader, or because the magnetic strip did not traverse the reading head at a relatively constant speed, or because the magnetic strip disconnected from the reading head at some point. In other words, a card swiping sequence succeeds if and only if the robot receives the information stored on the magnetic strip of the card from the reader.

It can be very inefficient to restart the whole sequence from scratch if movement is constrained during the latter phases. For instance, if the card is constrained when the robot attempts to move it through the gap, then it can be more efficient to return to the point where the card was inserted into the gap and try to swipe it again in a slightly different direction rather than to remove the card from the reader and start the whole manipulation sequence from scratch.

To learn to swipe the card better, each of the three

Sequence	Start	Insert	Slide	Card Read
1				No
2				Yes
3				Yes

Fig. 6. Examples of behavioral sequences performed by the robot. Each sequence consists of three behaviors: start, insert, and slide. A behavior completes successfully only if the robot is not constrained during the corresponding movement (✓ symbol). When a behavior fails (× symbol), the robot backtracks to the position from which it started the movement and tries to complete the phase again (diagonal arrows). A sequence succeeds if and only if the card was read during the execution of its “slide” behavior.

manipulation phases uses a predictive model of its outcome. The predictive model uses the joint space coordinates of the movement destination as features. Each phase can have only two outcomes: success or failure. A phase succeeds if and only if the robot was not constrained *and* the whole sequence triggers a successful card read. More specifically, a behavioral phase fails in each of the following cases: 1) constrained movement is detected during this phase, 2) constrained movement is detected 10 times in a row for the next phase in a sequence (this ensures that the robot does not get stuck in an infinite loop if it cannot find a way to execute the next phases of the sequence without ever being constrained at the current phase), 3) the whole sequence completes but the card is not read. If a phase fails, then the robot returns to the point in joint space where the phase started and tries to perform the movement again. Thus, each phase can be attempted many times during a sequence. Whenever success or failure is recorded for a behavioral phase, the robot updates the predictive model of its outcome. The transitions between failed and successful phases for three sample sequences are shown in Fig. 6.

To implement the predictive model, a k -NN classifier

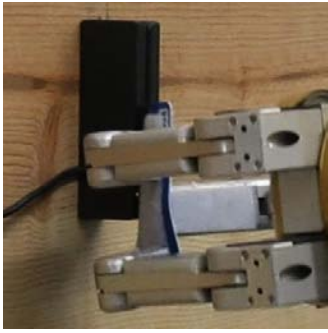


Fig. 7. Example of a failure that was recorded when the robot bent the card and forced it out of the gap. See text for more details.

with $k = 3$ was used. To select where to move, the robot generates 1000 candidate movement destinations using the four anchors points. The classifier predicts the probability of success for the candidates. The robot moves to the candidate destination with the highest probability of success.

VI. RESULTS

A. Card Swiping Behaviors Starting in the Gap

The goal of the first experiment was to quantify the utility of the constrained movement detector for successful card swiping. The behaviors started with the card already inserted into the top part of the card reader’s gap. In other words, the starting phase and the insertion phase of the card swiping sequence shown in Figure 6 were skipped and the robot only had to perform the final sliding phase.

Four experimental conditions were evaluated, which corresponded to the possible combinations of the following two binary variables: 1) learning using the card reader feedback was enabled or not, and 2) the constrained movement detector was enabled or not. For each of the four conditions the robot performed 100 card sliding behaviors. Figure 8 shows the number of successful card reads for each condition.

If learning from the card reader feedback was off and the constrained movement detector was also off, then the robot failed to swipe the card and forced it out of the gap despite its resistance. The resistance of the plastic card was insufficient to trigger the static torque limits of the WAM. For some other tasks these torque limits can help detect when the robot runs into constraints (e.g., static torque limits can detect when the WAM hits a wooden board as it tries to press a doorbell button [18]), but for the card reader task they were not sufficient. To prevent deformation of the card due to repeated bending, the human experimenter interrupted the robot and recorded a failure if the card was bent too much. This condition was attempted three times and each time the robot eventually bent and forced the card out of the gap. Figure 7 shows a typical example for the configuration of the card at the moment when the human experimenter would interrupt the program.

When the constrained movement detector was disabled, but feedback from the card reader was enabled, the robot was able to achieve several successful card reads before bending

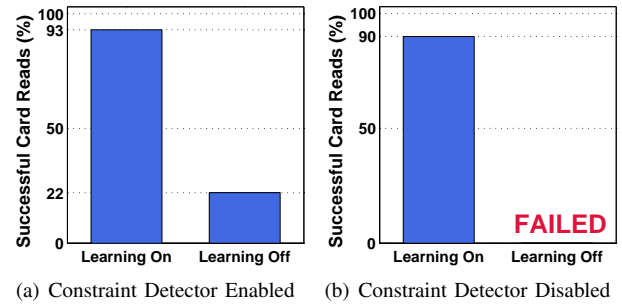


Fig. 8. Efficiency for card swiping behaviors that start with the card already in the gap of the card reader.

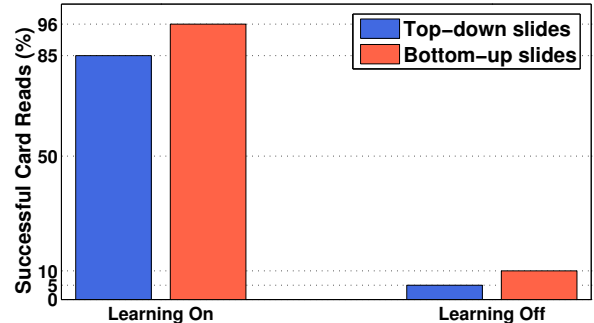


Fig. 9. Efficiency for complete card swiping behavioral sequences that consist of selecting the start position, inserting the card into the gap, and sliding the card through the gap. The constraint detector was enabled for each of the four conditions shown in this figure.

the card out of the gap. This focused subsequent behaviors and prevented future failures. Eventually, the robot achieved 90% card swiping efficiency in this condition. Even though the robot succeeded, there was a chance that it could force the card out of the gap before it could read it at least once, so this success may not be always reproducible.

When the constrained movement detector was on, but feedback from the reader was off, the robot kept the card in the gap and interrupted the behaviors that could bend it. Eventually it achieved 22% card swiping efficiency in this condition. In other words, the constrained movement detector allowed the robot to preserve the integrity of the experimental setup without relying on the card reader feedback.

Finally, when both learning from the card reader feedback and the constrained movement detector were on, the robot achieved 93% card swiping efficiency. This experiment showed that the constrained movement detector is very useful when learning to manipulate flexible objects. In the next set of experiments, where the robot had to learn to both insert and swipe the card through the gap, the constrained movement detector was always enabled.

B. Card Swiping Behaviors Starting Outside the Gap

The goal of the second experiment was to see if the robot can learn complete manipulation sequences that can successfully swipe a magnetic card through the card reader. This included selecting the start position outside the card reader, inserting the card into the gap of the card reader,

and sliding the card through the gap. The framework was evaluated for learning to slide the card in two directions: top-down and bottom-up. The swiping efficiency was also measured when the robot was not allowed to learn from the card reader feedback in order to estimate how often the robot can swipe the card simply by chance.

In each of the four experimental conditions, the robot performed 100 swiping sequences, which consisted of the three manipulation behaviors described in Section V. The results are shown in Fig. 9. The robot achieved 85% efficiency for top-down slides and 96% efficiency the bottom-up slides. The efficiency of the card swiping behaviors performed without learning was 5% for top-down swipes and 10% for bottom-up swipes.

These results show that using constrained movement detection and learning from card reader feedback can help achieve high card swiping efficiency. These results also show that the robot can learn complete card swiping behavioral sequences. Even though completing the sequence requires successful execution of each of the three phases, the robot was able to swipe the card efficiently in both directions.

VII. CONCLUSION AND FUTURE WORK

This work proposed a constraint detection algorithm based exclusively on proprioception. The algorithm was tested on the task of sliding a magnetic card through a card reader. The results of the experiments indicate that the algorithm can be used to associate unconstrained movement of the card with task completion. Manipulation strategies based on this concept were learned, which increased the robot's ability to complete the task successfully. Our algorithm can be used to enable robots to interact with environments in conditions where visual models cannot be created.

The method presented in this paper can be applied in many different ways. We have shown that it can be used to explore constraints and to learn from them. We foresee this method being used in other areas as well. One possible extension of this method is detecting that a key has been inserted into a keyhole and figuring out how to turn it. Using the exploration featured in our method, the robot may be able to open a lock. Furthermore, this method could allow robots to associate different keys with different locks. Observation of how the key turns in the lock would allow the robot to determine which key works for which lock.

Spatial semantic hierarchies could be also constructed using our method. For example, opening a door allows one to access the space behind it. Our method could detect that a closed door is a constraint that blocks the robot from accessing that space. Once open, however, the door no longer acts as a constraint.

Mobile robots could be able to use this method to extend the range of possible locations that they can access. Robots

that have to pass through doors may use our method to learn how to use door handles and how to open doors. For instance, exploring the directions of possible door movements would allow the robot to open a door regardless of whether it opens in or out. The robot may also learn to manipulate many different door handles by exploring how the unconstrained directions of movement change when the handle is grasped.

ACKNOWLEDGEMENTS

This work was funded in part by the NSF Research Experience for Undergraduates program (grant IIS-0851976).

REFERENCES

- [1] B. Siciliano and O. Khatib, *Handbook of Robotics*. New York: Springer-Verlag, 2008.
- [2] J. Morrow, B. Nelson, and P. Khosla, "Vision and force driven sensorimotor primitives for robotic assembly skills," in *Proc. of IEEE IROS*, vol. 3, 1995, pp. 234–240.
- [3] W. Meeussen *et al.*, "Autonomous door opening and plugging in with a personal robot," in *Proc. of the IEEE ICRA*, 2010, pp. 729 – 736.
- [4] B. Mayton, K. LeGrand, and J. Smith, "Robot, feed thyself: Plugging in to unmodified electrical outlets by sensing emitted ac electric fields," in *Proc. of IEEE ICRA*, 2010, pp. 715–722.
- [5] E. Torres-Jara, "A self-feeding robot," Master's thesis, MIT, 2002.
- [6] T. Yamada, K. Nagatani, and Y. Tanaka, "Autonomous insertion of a plug into real electrical outlet by a mobile manipulator," in *Proc. of the International Conference on Field and Service Robotics*, P. Corke and S. Sukkarieh, Eds., vol. 25. Springer, 2006, pp. 389–400.
- [7] L. Bustamante and J. Gu, "Localization of electrical outlet for a mobile robot using visual servoing," in *Proc. of the Canadian Conf. on Electrical and Computer Eng. (CCECE)*, 2007, pp. 1211–1214.
- [8] S. Chitta *et al.*, "Autonomous door opening and plugging in with a personal robot," in *Proc. of RSS Workshop on Mobile Manipulation*, 2009.
- [9] H. Bruyninckx, S. Dutre, and J. De Schutter, "Peg-on-hole: a model based solution to peg and hole alignment," in *Proc. of IEEE ICRA*, vol. 2, May 1995, pp. 1919–1924.
- [10] P. Koonce, V. Dutell, J. Farrington, V. Sukhoy, and A. Stoytchev, "Toward learning to solve insertion tasks: A developmental approach using exploratory behaviors and proprioception," in *Proceedings of the 25th National Conference on Artificial Intelligence (AAAI)*, 2011, pp. 1798–99.
- [11] R. Suarez, L. Basanez, and J. Rosell, "Using configuration and force sensing in assembly task planning and execution," *IEEE International Symposium on Assembly and Task Planning*, pp. 273– 279, 1995.
- [12] W. Paetsch and G. von Wichert, "Solving insertion tasks with a multifingered gripper by fumbling," in *Proc. of IEEE ICRA*, vol. 3, May 1993, pp. 173 –179.
- [13] K. Shutts *et al.*, "Young children's representations of spatial and functional relations between objects," *Child Development*, vol. 80, no. 6, pp. 1612–27, 2009.
- [14] M. Hayashi and T. Matsuzawa, "Cognitive development in object manipulation by infant chimpanzees," *Animal Cognition*, vol. 6, pp. 225–233, 2003.
- [15] H. Ornkloo and C. von Hofsten, "Fitting objects into holes: On the development of spatial cognition skills," *Developmental Psychology*, vol. 43, no. 2, pp. 404–416, 2007.
- [16] Wikipedia, "ISO/IEC 7810 — Wikipedia, The Free Encyclopedia," 2011, 17-September-2011. [Online]. Available: <http://en.wikipedia.org/w/index.php?title=ISO/IEC.7810&oldid=443534260>
- [17] G. Golub and C. Van Loan, *Matrix computations*. Johns Hopkins University Press, 1996, vol. 3.
- [18] V. Sukhoy and A. Stoytchev, "Learning to detect the functional components of doorbell buttons using active exploration and multimodal correlation," in *Proc. of IEEE Humanoids*, 2010, pp. 572–579.