# Tools for Data-driven Modeling of Within-Hand Manipulation with Underactuated Adaptive Hands

**Avishai Sintov** SINTOV1@TAUEX.TAU.AC.IL
*School of Mechanical Engineering, Tel-Aviv University, Tel-Aviv, Israel.*

**Andrew Kimmel** ANDREW.KIMMEL@RUTGERS.EDU
**Bowen Wen** WENBOWENXJTU@GMAIL.COM
**Abdeslam Boularias** ABDESLAM.BOULARIAS@RUTGERS.EDU
**Kostas E. Bekris** KOSTAS.BEKRIS@RUTGERS.EDU
*Computer Science Department, Rutgers, the State University of New Jersey, Piscataway, NJ, USA.*

## Abstract

Precise in-hand manipulation is an important skill for a robot to perform tasks in human environments. Practical robotic hands must be low-cost, easy to control and capable. 3D-printed underactuated adaptive hands provide such properties as they are cheap to fabricate and adapt to objects of uncertain geometry with stable grasps. Challenges still remain, however, before such hands can attain human-like performance due to complex dynamics and contacts. In particular, useful models for planning, control or model-based reinforcement learning are still lacking. Recently, data-driven approaches for such models have shown promise. This work provides the first large public dataset of real within-hand manipulation that facilitates building such models, along with baseline data-driven modeling results. Furthermore, it contributes ROS-based physics-engine model of such hands for independent data collection, experimentation and sim-to-reality transfer work.

**Keywords:** Data-driven models; Within-hand manipulation; Underactuated hands.

## 1. Introduction

Underactuated hands with compliant fingers offer an appealing grasping solution due to their adaptability to objects of uncertain size and shape. They enable stable and robust grasps via open-loop control without tactile sensing (Odhner and Dollar, 2015; Deimel and Brock, 2016). Furthermore, the open-source underactuated robotic hands considered in this work (Ma and Dollar, 2017) are easy to fabricate using 3D-printing and to modify. However, due to uncertainties in the manufacturing process, the hands can easily differ in size, weight, friction and inertia. On top of that, passively elastic joints in underactuated hands are difficult to model. Consequently, precise analytical models for such hands are often unavailable as they are hard to derive (van Hoof et al., 2015). A relatively new direction in this context, is the development of data-driven models proposed by Sintov et al. (2019), which brings the promise of enabling accurate enough predictions and adaptiveness to individual systems that can be used for planning and control (Kimmel et al., 2019).

To facilitate further efforts on data-driven research of adaptive hands, this paper provides an experimental setup using both real data and simulations. The dataset could be used for different tasks, including: *identifying a model* of the hand from data to acquire a mapping of states and actions to new states (Sintov et al., 2019); *failure identification* to predict whether the hand drops an object given a state and an action (Calli et al., 2018); and *model transfer learning* to minimize the data needs for updating an existing model for a new hand, a new object, or even environment changes (Bocsi et al., 2013; Makondo et al., 2015). This allows researchers to propose and evaluate

algorithms for modeling, reinforcement learning, transfer learning and failure detection. In addition, this work provides a framework for sim-to-real transfer learning.

Datasets in robotic manipulation are an efficient platform for advancing research and comparing to the state-of-the-art (Yu et al., 2016). As such, we provide the Rutgers Underactuated-hand Manipulation (RUM)[1] dataset. RUM includes data for several hands with various objects using an autonomous data collection process involving an industrial robot that decreases data uncertainty (Fig. 1). To the best of the authors' knowledge, this is the first large dataset for within-hand manipulation with underactuated hands. In previous work by Sintov et al. (2019), modeling for adaptive hands was presented using a small scale dataset collected manually and carefully in an exhausting procedure. In this large scale automated collection, data variance is much larger which presents challenges in prediction accuracy.

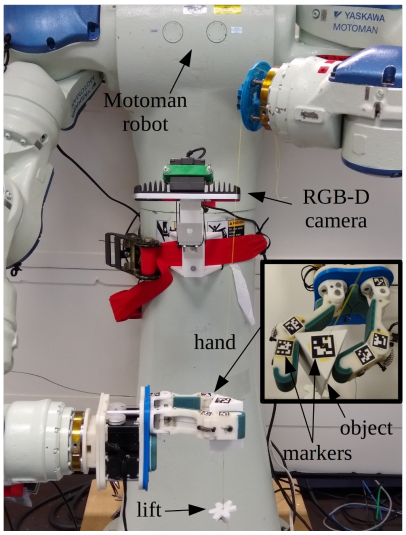



Figure 1: Adaptive hand during data gathering. The dual-arm robot is used to automate the collection process and acquire a wide dataset.

Furthermore, we present an easily accessible *ROS-Gazebo*-based model of adaptive hands (Fig. 4) along with open-source code to control the hand, collect and process data. The simulation enables the use of two and three finger hands, Model-T42 and Model-O (Ma and Dollar, 2017), respectively. While the simulation is based on some linear assumptions and only approximates the behaviour of a real hand, it can be used for prototyping and bootstrapping. Users are able to vary hand properties, such as geometry, friction, spring stiffness, and maximum actuator load, so that different hands can be simulated. In addition, a user can choose the set of features (e.g., object position and velocity, joint angles and angular velocities, etc.) to include in the state as well as the allowed actions (e.g., discrete or continuous) for learning purposes. Users can alternatively build new models to independently collect data using the provided open-source code.

## 2. Related Work

Compliance in underactuated hands makes the derivation of models challenging due to the complex responses of the passive joints and uncertainties in the internal (e.g., joint friction) and external (e.g., between fingers and object) friction (van Hoof et al., 2015). A gripper configuration is directly dependent on the forces being applied to each of the fingers. Without tactile sensing, a precise frictional model is required to estimate such forces, which is difficult to estimate at every point on the contact surface. Typical frictional models assume uniformity of surfaces (Yu et al., 2016). For this reason, along with the inability to control individual joint positions of an underactuated system, physically accurate models are difficult to derive. Modeling tools for underactuated manipulation have been introduced in prior work (Laliberté and Gosselin, 1998; Odhner and Dollar, 2011; Rocchi et al., 2016). These efforts examine joint configurations, torques, and energy with simplified frictional models. A popular modeling technique applies a hybrid parallel/serial approach using screw theory, which further simplifies derivation (Borras and Dollar, 2013). These techniques tend

1. Rutgers Underactuated-hand Manipulation (RUM) dataset - Webpage and source code, https://github.com/avishais/underactuated_hand_tools

to be sensitive to assumptions in external constraints and are mostly suitable for simulation. The lack of good models for adaptive hands has led to the data-based modeling approach presented by Sintov et al. (2019). Consequently, in this paper we provide a framework that could facilitate further advancement and benchmarking of modeling approaches for adaptive hands (Calli et al., 2015; Cruciani et al., 2020).

A data-based model is commonly formed as a *transition model* that maps a given state and action to the next state (Punjani and Abbeel, 2015). Such models are important for model-based RL (Polydoros and Nalpantidis, 2017) and planning (Kimmel et al., 2019). They are often obtained through non-linear regression in a high-dimensional space. Stochastic models, such as Dynamic Bayesian Networks (DBN) (Nguyen et al., 2013) and Locally Weighted Bayesian Regression (LWBR) (Bagnell and Schneider, 2001), provide a probability distribution over predictions. Another common approach for stochastic modeling of dynamical systems is Gaussian Processes (GP) (Rasmussen and Williams, 2005; Ko et al., 2007; Deisenroth et al., 2014). They have achieved efficient learning, but rely on good coverage of the underlying space during training. Data-driven models have been used to learn the probability distribution of an object after a grasp (Paolini et al., 2014) or during regrasping (Paolini and Mason, 2016). A hybrid modeling approach combining analytical and data-based models showed improved accuracy in feed-forward control (Reinhart et al., 2017). The tools we provide in this work would enable to build data-based models for adaptive hand and enable benchmarking with the state-of-the-art.

## 3. The RUM dataset

We consider the 2-finger Model-T42 hand seen in Figures 1 and 7. All hands used are fabricated through 3D printing and are open-sourced (Ma and Dollar, 2017). Each finger of the hand has two compliant joints with springs. In addition, an actuator provides flexion to the finger through a tendon running along the finger length. The fingers also have high friction pads to avoid slipping. Specifications for the dataset are as follows:

- *State:* We measure different features including object pose, finger locations and actuator loads, which can be used to represent the state of the hand-object system.
- *Actions:* The hand is controlled through changes of actuator angles, where an atomic action $\mathbf{u}$ corresponds to unit changes of the actuators' angles at each time step. That is, an action moves actuator $i$ with an angle of $\lambda\gamma_i$, where $\lambda$ is a predefined unit angle and $\gamma_i$ is in the range [-1, 1]. In the dataset, we distinguish between data with either discrete or continuous actions. In the discrete case and in each time step, $\gamma_i$ is equal to either 1, -1 or 0, yielding eight possible actions – while excluding the stall action $(0, 0)$.
- *Objects:* We have used a set of 14 3D-printed objects, all having a constant profile parallel to the motion plane. All objects are described in Table 1 and seen in Figure 2, while detailed drawings can be viewed in the project's web-page.
- *Hands:* We include data and validation paths for two different hands with the same design. Due to the fabrication process, large variations in the motion between both hands exist. Thus, users would be able to evaluate transfer learning algorithms to map a model of one hand to the other.
- *Validation:* For each object, we recorded 10 long validation paths within the workspace produced with the same action sequences (also included for independent collection). The paths have a substantial coverage and employ frequent changes of action.

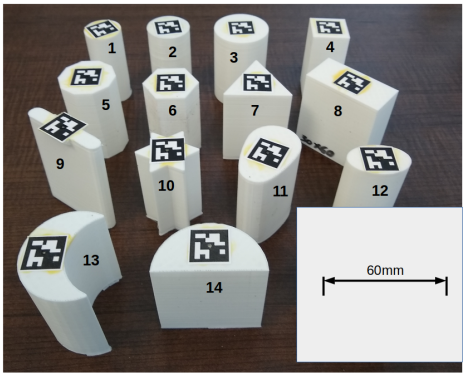


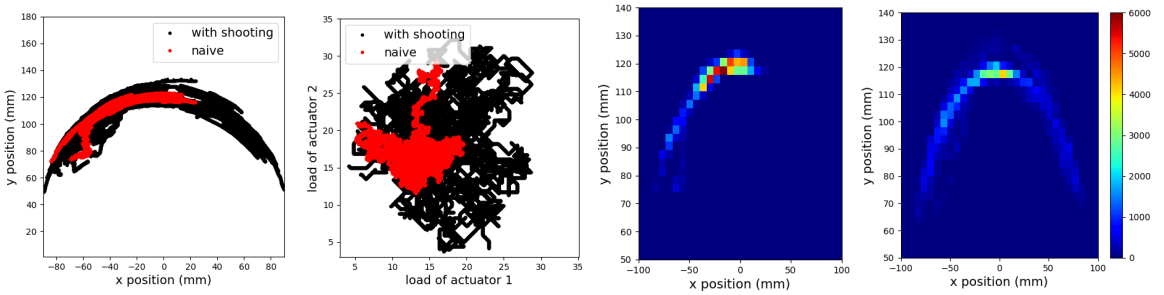

Figure 2: Test objects in the dataset.

Table 1: Description of test objects

| | Profile/Object | Description | Abbr. |
|---|---|---|---|
| 1 | Circular | $\varnothing 30mm$ | cyl30 |
| 2 | Circular | $\varnothing 35mm$ | cyl35 |
| 3 | Circular | $\varnothing 45mm$ | cyl45 |
| 4 | Square | $30mm \times 30mm$ | sqr30 |
| 5 | Reg. decagon | Circum. circle $\varnothing 42mm$ | poly10 |
| 6 | Reg. hexagon | Circum. circle $\varnothing 40mm$ | poly6 |
| 7 | Reg. triangle | Edge length $50mm$ | tri50 |
| 8 | Rectangle | $30mm \times 60mm$ | rec60 |
| 9 | Rectangle | $10mm \times 60mm$ | rec10 |
| 10 | Hexagram | Circum. circle $\varnothing 40mm$ | str40 |
| 11 | Egg | H. $52mm$, Mx. W. $45mm$ | egg50 |
| 12 | Ellipse | $25mm \times 40\ mm$ | elp40 |
| 13 | Crescent | Out $\varnothing 55mm$, in $\varnothing 45mm$ | cre55 |
| 14 | Semi-circular | $\varnothing 60mm$ | sem60 |

We have built an automated collection system seen in Figure 1. The adaptive hand is mounted on an arm of a Motoman SDA10F dual-arm robot. At each episode of the collection, the robot grasps the object, performs within-hand manipulation until it drops, and then repeats. To do so, a thin string runs through a hole in the center of the object and is connected to the floor (through a spring) and to the second arm. When the object drops, it is stopped by the lift on the string. Then, the upper arm pulls the string up along with the object into the reach of the fingers toward a new grasp. During manipulation, we record object pose (geometric center and orientation), finger positions and actuator loads. Poses of object and fingers are measured through fiducial markers. A base marker is fixed on the hand for reference. During in-hand manipulation, the pose (position and orientation) of the object relative to the hand, along with the position of fingers, is recorded. The loads of the actuators are measured directly from the actuators' feedback. Data points are logged along with a time stamp and the corresponding action.

Data collection can be performed naively by randomly choosing actions. However, the collected data would be concentrated near the initial grasping state and be sparse on farther regions. Therefore, a shooting approach can be employed where at each episode and with some probability, the hand will go towards an arbitrary direction with a large number of steps, and then apply random actions to reduce uncertainty in the region. As seen in Figure 3, sparsity is reduced within the state-space and exploration is increased using the shooting approach compared to the naive one.




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

Figure 3: (a) Object position and (b) actuator loads when collecting 80,000 data points (in the simulation environment) with the naive (red) and shooting (black) approaches. (c) and (d) are position density plots for the naive and shooting approaches, respectively. With the naive, the data is seen to be concentrated near the start state with less exploration while with the shooting, the data is more dispersed.

The acquired data comprises of observable state-action trajectories, with no additional labeling required. Thus, the resulting data is a set of state-action points $\mathcal{P} = \{\bar{\mathbf{x}}_0, \ldots, \bar{\mathbf{x}}_k\}$, where $\bar{\mathbf{x}}_i = (\mathbf{x}_i^T,\ \mathbf{u}_i^T)^T$. The distributed data is recorded in 10 $Hz$. The trajectories in $\mathcal{P}$ are preprocessed to a set of training inputs $\bar{\mathbf{x}}_i$ and corresponding output labels of the next state $\mathbf{x}_{i+1}$ to define $\mathcal{T} = \{(\bar{\mathbf{x}}_i), (\mathbf{x}_{i+1})\}_{i=1}^{N}$. For each $\bar{\mathbf{x}}_i$, we also label $\mathbf{d}_i = \{\texttt{success}, \texttt{fail}\}$ indicating whether the transition from $\mathbf{x}_i$ with action $\mathbf{u}_i$ failed. The final outcome is the RUM dataset, which comprises of approximately 300,000 transition points for each object, both for discrete and continuous actions.

## 4. Physics Engine Simulation

We also provide a physics engine simulation of the Model T-42 and Model-O hands (Ma and Dollar, 2017) in the ROS-Gazebo environment, as seen in Figure 4. Similar to the real hand, we control the actuators angles $\phi \in \mathbb{R}^d$ ($d = 2$ and $d = 3$ for the 2- and 3-finger hands, respectively) which, when rotated, increase or release tension $\mathbf{f}_t$ on the tendons. We assume a linear model, such that $\mathbf{f}_t = Q\phi$ where $Q > 0$ is a diagonal matrix. When modeled in Gazebo, the two joints of a finger are actually fully-actuated and can be controlled individually. To simulate the compliance through the tendons and similar to prior work (Rocchi et al., 2016), we enforce underactuation by coupling joint torques $\tau \in \mathbb{R}^{2d}$ according to

$$\tau = R\mathbf{f}_t - K\mathbf{q} - \tau_d(\dot{\mathbf{q}}) \tag{1}$$

where $R$ is a $2d \times d$ matrix determining the distribution of the tendon forces on the joints. Matrix $K$ is a $2d \times 2d$ diagonal stiffness matrix where its values are defined by the coefficients of the springs mounted on each joint. Vectors $\mathbf{q} \in \mathbb{R}^{2d}$ and $\dot{\mathbf{q}} \in \mathbb{R}^{2d}$ are the joint angles and angular velocities, respectively, measured within the simulator. $\tau_d(\dot{\mathbf{q}})$ is the damping vector and can also include torsional friction at the joints. In this formulation, the $d$ input values of the actuators angles $\phi$ determine the $2d$ finger joint torques $\tau$ and by that, simulate underactuation. Information on additional properties and settings is provided in the dataset web-page.

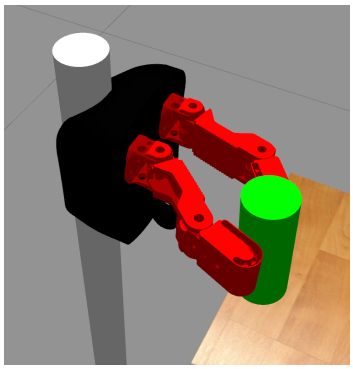


Figure 4: Physics engine simulation of the Model T42 adaptive hand in ROS-Gazebo.

## 5. Baseline results

This section provides baseline analysis with the data and simulations to identify some insights and open problems. The analysis is focused on data collected with discrete actions. The data-collection process and the baseline experiments can be viewed in the supplementary video[2].

### 5.1. Data-based Modeling

In previous work (Sintov et al., 2019; Sintov et al., 2020), a data-based modeling approach was presented using Gaussian Process (GP) and Artificial Neural Networks (ANN). The position of the manipulated object and the actuator loads were shown to be sufficient in describing the state of a real hand under a quasi-static motion assumption. Nevertheless, the data collection for that work was performed manually and, therefore, the variance of the initial state was rather low. In the large scale autonomous data collection here, the variance is quite high and thus, a similar setup yields inaccurate predictions. The addition of the state (object pose and actuator loads) at the initial grasp of the object

2. Supplementary video: https://youtu.be/7dtjIIzFpb0

to the action, however, significantly improves the accuracy. That is, the model requires some knowledge of past experience. This is not required, however, for the simulated hand model. We also note that for non-circular object profiles, the orientation angle must be included in the state to allow the model to consider the shape. Nonetheless, profile symmetry in some objects allows trimming of the angle range.

For the RUM dataset, a Manifold-Learning GP proposed by Sintov et al. (2019) was applied for local regression. Given a query state-action pair, $K_1 = 1,000$ nearest neighbors in the training data are found. Next, a diffusion map for the $K_1$ points is built acquiring a reduced representation $\mathcal{E}$ of the data in the lower dimensional subspace (dimension 3). Finally, the $K_2 = 100$ closest points to the query point, evaluated through geodesic distance across the subspace, are chosen for regression. For data collected in the simulated environment while manipulating a 19.2 mm diameter cylinder, we have also trained a Rectified Linear Unit (ReLU) NN with two hidden layers and 200 neurons each. Figure 5 presents the model error plots with regards to the predicted horizon length for both the RUM dataset and simulated hand. The figure shows that it is harder to predict the motion of non-circular objects. Figure 6 presents the RMSE with regards to the data size used for GP regression. The computation times averaged over 1,000 prediction trials with GP and NN are 440 milliseconds and 2.8 milliseconds, respectively. The computation time was evaluated on a single Intel Xeon E5-4650 processor with 8 GB of RAM. Evidently, NN is more computationally efficient than GP.

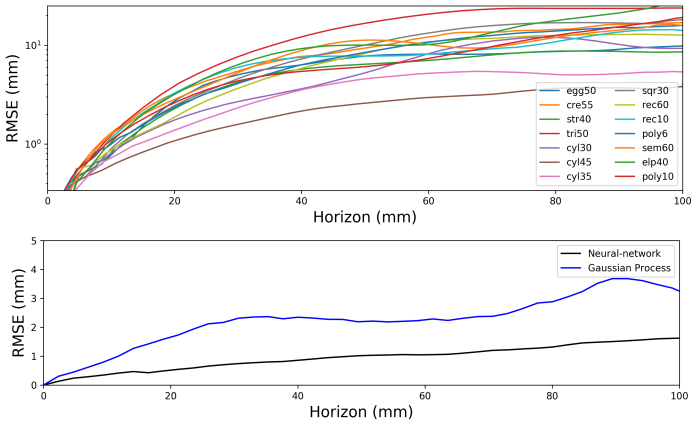



Figure 5: Root-mean-square error with regards to the length of the predicted horizon: (top) with GP for different objects and in a logarithmic scale, and (bottom) with NN and GP, for a cylinder of 19.2 $mm$ within the physics-engine environment.

We also demonstrate the implementation of a naive closed-loop controller with a simple "follow-the-carrot" scheme, where the intermediate goal is moved along the path as the object advances. The path was generated using a motion planner to a hand picked goal while using the NN model for propagation. We use the GP model trained over the data of the corresponding object. At each step and after acquiring feedback of the current state, the GP model predicts the next state for each of the eight possible actions. The action that is exerted is the one that predicts motion closest to the current intermediate goal position along the path. For each

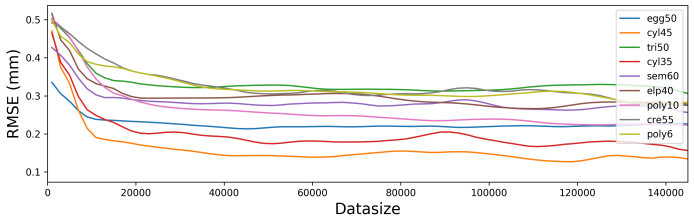



Figure 6: RMSE of GP predictions for different objects and with regards to the number of transition points in the data.

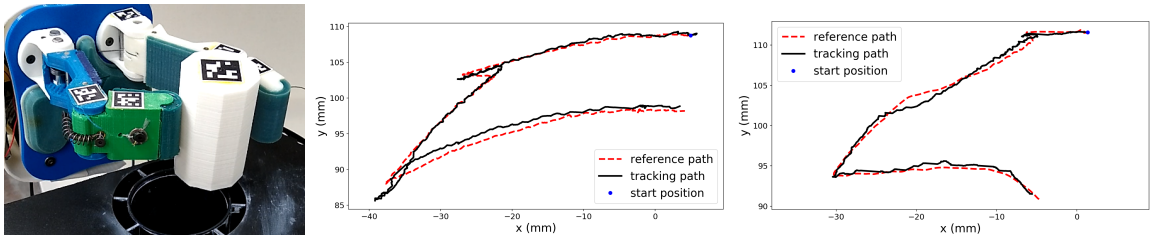



Figure 7: Model-based closed-loop tracking in a peg-in-a-hole problem: (left) poly10 object is manipulated on a path seen in the (middle) plot. Path tracking for cyl35 is shown on the (right) plot.

of the 10 test paths, 10 tracking trials were executed. Table 2 summarizes the tracking RMSE for all trials and for some of the objects, while Figure 7 presents two tracked paths.

### 5.2. Failure identification

We compare eight common classifiers (Singh et al., 2016) with all test objects: Gaussian Process, Decision Tree, Linear Support Vector Machine (SVM), Radial Basis Function (RBF) SVM, Adaptive Boosting (AdaBoost), Random Forest, Nearest Neighbors and Neural Network. As mentioned before, labeled state-action pairs were extracted from the recorded data while 100 points are used for validation. Table 3 presents results for the overall classification success rates with the RUM dataset and the simulator. We note that for cylinders, the orientation of the object has no effect on the probability of failure. However, for other non-cylindrical objects such as a polygonal profile, the orientation angle is critical for successful classification. Thus, the angle is included in the state.

Table 2: RMSE for model-based closed-loop tracking

| Object | RMSE (mm) | Object | RMSE (mm) |
|---|---|---|---|
| cyl30 | 1.03 | poly6 | 1.48 |
| cyl35 | 1.29 | poly10 | 1.16 |
| sqr30 | 1.37 | str40 | 0.91 |
| tri50 | 1.06 | elp40 | 1.14 |

### 5.3. Model transfer

Training transition models typically requires several hours of data collection. To alleviate this problem, it is commonly considered to transfer already learned models across different robots or tasks (Schramm et al., 2020; Makondo et al., 2018). To motivate this model transfer problem in the context of adaptive hands, we show results for prediction error with regards to the modified system. We have recorded data for one hand (termed *blue* hand) with several objects and the modeling results were presented previously. Later, we recorded data and the 10 test paths with a target hand (termed *red* hand). Figure 4 shows prediction results for three objects. The dashed curves are prediction errors for test paths recorded with the blue hand and evaluated with the blue hand model. In contrast, the continuous curves are test paths recorded with the red hand but predictions were also performed with the blue hand model. Visual motion comparisons between the two hands can be seen in the supplementary video. These results clearly show the inability to directly use an old model for a modified system, indicating that the model should be adjusted. To make the approach viable, the transfer should be done with the minimal amount of new data.

Table 3: Failure classification results for the test objects and with different classifiers

| Object | Gaussian Process | Decision Tree | Linear SVM | RBM SVM | AdaBoost | Random Forest | Nearest Neigh. | Neural Network |
|---|---|---|---|---|---|---|---|---|
| Real data | | | | | | | | |
| cyl30 | 96.8% | 84.6% | 66.6% | 95.3% | 86.4% | 83.9% | 93.8% | 83.9% |
| cyl35 | 96.9% | 86.3% | 78.1% | 93.1% | 86.6% | 83.2% | 93.5% | 81.1% |
| cyl45 | 96.6% | 92.6% | 70.0% | 94.4% | 93.5% | 88.2% | 92.9% | 88.9% |
| poly6 | 94.8% | 86.6% | 72.4% | 92.2% | 87.9% | 81.4% | 87.5% | 87.9% |
| poly10 | 95.7% | 91.8% | 71.3% | 96.5% | 90.7% | 87.6% | 91.1% | 91.1% |
| sqr30 | 96.3% | 88.2% | 70.9% | 91.9% | 86.7% | 86.4% | 93.0% | 85.3% |
| str40 | 94.5% | 80.0% | 62.2% | 94.2% | 82.9% | 77.4% | 90.0% | 81.9% |
| rec10 | 97.7% | 82.9% | 73.8% | 93.2% | 82.9% | 80.7% | 94.7% | 83.3% |
| rec60 | 95.1% | 86.7% | 65.3% | 93.9% | 87.9% | 84.2% | 93.9% | 84.6% |
| egg50 | 97.3% | 78.3% | 66.9% | 93.4% | 80.0% | 73.8% | 94.3% | 80.0% |
| elp40 | 95.9% | 83.1% | 72.4% | 93.7% | 87.8% | 85.6% | 90.4% | 84.5% |
| tri50 | 96.0% | 77.9% | 67.3% | 92.9% | 81.1% | 78.7% | 92.1% | 78.7% |
| cre55 | 96.8% | 83.7% | 77.1% | 95.1% | 87.5% | 85.0% | 95.8% | 84.7% |
| sem60 | 95.9% | 88.0% | 70.9% | 94.8% | 88.3% | 81.8% | 92.2% | 83.2% |
| Simulated data | | | | | | | | |
| cyl19 | 98.7% | 91.2% | 78.7% | 97.5% | 91.2% | 92.5% | 97.5% | 92.5% |

## 6. Conclusion

In this work, we have proposed a set of tools that can be used for evaluating algorithmic performance with adaptive hands. We focus on the evaluation of methods to process data collected from such hands to generate viable models. We include a large dataset of transition points recorded with real hands and different objects along with a physics engine based simulation platform. In addition, the provided source code enables users to independently collect data with different scenarios, objects, and hands. We have identified three possible usages of the data: modeling a hand to predict its motion, identifying probable failure modes, and model transfer. Baseline results demonstrated the feasibility of the data and motivated further research in the modeling and control of adaptive hands.

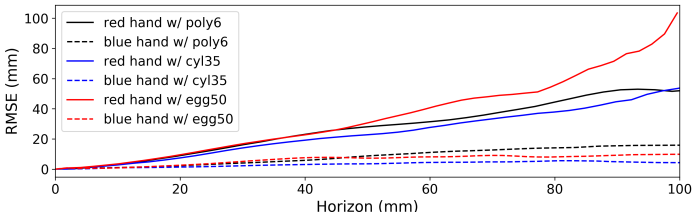



Table 4: Root-mean-square error with regards to the length of the GP predicted horizon and for three objects. Dashed curves refer to the blue hand model predicting paths of the blue hand. Continuous curves are prediction errors using the blue hand model on test paths recorded with the red hand.


## Acknowledgments

This work was supported by NSF awards 1734492, 1723869 and 1846043. Any opinions and conclusions expressed in this work are made by the authors and do not necessarily reflect the views of the sponsor.

## References


J. A. Bagnell and J. G. Schneider. Autonomous helicopter control using reinforcement learning policy search methods. In *IEEE Int. Conf. on Rob. and Aut.*, volume 2, pages 1615–1620, 2001.

Botond Bocsi, Lehel Csató, and Jan Peters. Alignment-based transfer learning for robot models. In *The International Joint Conference on Neural Networks*, pages 1–7, 2013.

J. Borras and A. M. Dollar. A parallel robots framework to study precision grasping and dexterous manipulation. In *IEEE Int. Conf. on Rob. and Aut.*, pages 1595–1601, May 2013.

B. Calli, A. Walsman, A. Singh, S. Srinivasa, P. Abbeel, and A. M. Dollar. Benchmarking in manipulation research: Using the yale-cmu-berkeley object and model set. *IEEE Robotics Automation Magazine*, 22(3):36–52, Sep. 2015.

B. Calli, K. Srinivasan, A. Morgan, and A. M. Dollar. Learning modes of within-hand manipulation. In *IEEE Int. Conf. on Rob. and Aut.*, pages 3145–3151, May 2018.

S. Cruciani, B. Sundaralingam, K. Hang, V. Kumar, T. Hermans, and D. Kragic. Benchmarking in-hand manipulation. *IEEE Robotics and Automation Letters*, 5(2):588–595, 2020.

Raphael Deimel and Oliver Brock. A novel type of compliant and underactuated robotic hand for dexterous grasping. *The International Journal of Robotics Research*, 35(1-3):161–185, 2016.

M. P. Deisenroth, P. Englert, J. Peters, and D. Fox. Multi-task policy search for robotics. In *IEEE Int. Conf. on Rob. and Aut.*, pages 3876–3881, May 2014.

A Kimmel, A Sintov, B Wen, A Boularias, and K Bekris. Belief-space planning using learned models with application to underactuated hands. In *Int. Symposium on Robotics Research*, 2019.

J. Ko, D. J. Klein, D. Fox, and D. Haehnel. Gaussian processes and reinforcement learning for identification and control of an autonomous blimp. In *IEEE Int. Conf. on Rob. and Aut.*, pages 742–747, April 2007.

T. Laliberté and C. M. Gosselin. Simulation and design of underactuated mechanical hands. *Mech. and Mach. The.*, 33(1-2):39–57, Jan 1998.

R. R. Ma and A. M. Dollar. Yale openhand project: Optimizing open-source hand designs for ease of fabrication and adoption. *IEEE Rob. & Aut. Mag.*, 24:32–40, 2017.

N. Makondo, B. Rosman, and O. Hasegawa. Accelerating model learning with inter-robot knowledge transfer. In *IEEE International Conference on Robotics and Automation*, pages 2417–2424, May 2018.

Ndivhuwo Makondo, Benjamin Rosman, and Osamu Hasegawa. Knowledge transfer for learning robot models via local procrustes analysis. In *IEEE-RAS International Conference on Humanoid Robots*, pages 1075–1082, 2015.

T. Nguyen, Z. Li, T. Silander, and T. Leong. Online feature selection for model-based reinforcement learning. In *Int. Conf. on Mach. Lea.*, pages 498–506, 2013.

Lael U. Odhner and Aaron M. Dollar. Dexterous manipulation with underactuated elastic hands. In *IEEE Int. Conf. on Rob. and Aut.*, pages 5254–5260, May 2011.

Lael U. Odhner and Aaron M. Dollar. Stable, open-loop precision manipulation with underactuated hands. *Int. J. of Rob. Res.*, 34(11):1347–1360, Sep 2015.

R. Paolini and M. T. Mason. Data-driven statistical modeling of a cube regrasp. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2554–2560, Oct 2016.

Robert Paolini, Alberto Rodriguez, Siddhartha S. Srinivasa, and Matthew T. Mason. A data-driven statistical framework for post-grasp manipulation. *Int. J. of Rob. Res.*, 33(4):600–615, 2014.

Athanasios S. Polydoros and Lazaros Nalpantidis. Survey of model-based reinforcement learning: Applications on robotics. *J. of Intel. & Rob. Sys.*, 86(2):153–173, May 2017.

A. Punjani and P. Abbeel. Deep learning helicopter dynamics models. In *IEEE Int. Conf. on Rob. and Aut.*, pages 3223–3230, May 2015.

Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning*. Cambridge, MA, The MIT Press, 2005.

Felix Reinhart, Zeeshan Shareef, and Jochen Steil. Hybrid analytical and data-driven modeling for feed-forward robot control. *Sensors*, 17, 02 2017.

A. Rocchi, B. Ames, Zhi Li, and K. Hauser. Stable simulation of underactuated compliant hands. In *ICRA*, pages 4938–4944, May 2016.

Liam Schramm, Avishai Sintov, and Abdeslam Boularias. Learning to transfer dynamic models of underactuated soft robotic hands. In *Proc. of the IEEE Int. Conf. on Rob. & Aut.*, Paris, France, 2020.

A. Singh, N. Thakur, and A. Sharma. A review of supervised machine learning algorithms. In *Int. Conf. on Comp for Sustainable Global Development*, pages 1310–1315, March 2016.

A. Sintov, A. Morgan, A. Kimmel, A. Dollar, K. Bekris, and A. Boularias. Learning a state transition model of an underactuated adaptive hand. *IEEE Rob. & Aut. Let.*, 4(2):1287–1294, 2019.

Avishai Sintov, Andrew Kimmel, Kostas Bekris, and Abdeslam Boularias. Motion planning with competency-aware transition models for underactuated adaptive hands. In *Proc. of the IEEE Int. Conf. on Rob. & Aut.*, France, 2020.

Herke van Hoof, Tucker Hermans, Gerhard Neumann, and Jan Peters. Learning robot in-hand manipulation with tactile features. In *IEEE-RAS Int. Conf. on Hum. Rob.*, pages 121–127, Nov 2015.

K. Yu, M. Bauzá, N. Fazeli, and A. Rodriguez. More than a million ways to be pushed. a high-fidelity experimental dataset of planar pushing. *IEEE/RSJ Int. Conf. on Int. Rob. & Sys.*, 2016.