Abstract—In this paper, we propose a hybrid methodology based on a combination of analytical, numerical and machine learning methods for performing dexterous, in-hand manipulation with simple, adaptive robot hands. A constrained optimization scheme utilizes analytical models that describe the kinematics of adaptive hands and classic conventions for modelling quasistatically the manipulation problem, providing intuition about the problem mechanics. A machine learning (ML) scheme is used in order to split the problem space, deriving task-specific models that account for difficult to model, dynamic phenomena (e.g., slipping). In this respect, the ML scheme: 1) employs the simulation module in order to explore the feasible manipulation paths for a specific hand-object system, 2) feeds the feasible paths to an experimental setup that collects manipulation data in an automated fashion, 3) uses clustering techniques in order to group together similar manipulation trajectories, 4) trains a set of task-specific manipulation models and 5) uses classification techniques in order to trigger a task-specific model based on the user provided task specifications. The efficacy of the proposed methodology is experimentally validated using various adaptive robot hands in 2D and 3D in-hand manipulation tasks.

I. INTRODUCTION

Dexterous, in-hand manipulation has received increased attention over the last decades, as robots are rapidly introduced in human-centric, dynamic environments where they are called to execute robustly various everyday life tasks. The most common interaction of robots with their surroundings is via grasping and/or manipulating objects or a specific part of the environment (e.g., a door handle, a button, a knob etc.). For many years, the field of robotic manipulation was mainly associated with multi-fingered, highly dexterous, fully actuated, expensive robot hands that were typically equipped with sophisticated sensing elements (e.g., tactile or force sensors) and required complicated control laws.

Over the last decade a new class of adaptive, compliant, underactuated robot hands has become increasingly popular, mainly for robust grasping [1]–[5]. The low cost, the lightweight design and the low complexity of these hands make them easy to acquire and develop, while the inherited compliance and the reduced number of actuators and constraints facilitate the execution of robust grasps even under significant object pose uncertainties, simplifying not only grasping but also dexterous in-hand manipulation.

In particular, a series of recent studies have demonstrated the efficacy of adaptive hands in performing dexterous, in-hand manipulation tasks [6]–[8] (e.g., equilibrium point manipulation, finger gaiting etc.). Although manipulation experiments with these hands can be easily conducted and the user can easily tune their behavior for specific tasks and objects by empirically choosing the required control gains / parameters, modelling of the manipulation problem still remains an extremely challenging task due to actuation constraints and hard to model phenomena.

In this paper, we present a methodology that combines learning, numerical and analytical methods in a synergistic fashion, in order to simplify in-hand manipulation with adaptive hands (e.g., underactuated and/or compliant). More precisely, a simulation module is synthesized using a combination of numerical and analytical methods that predicts the kinematics of adaptive hands and facilitates a quasistatic modelling of the manipulation problem. A learning scheme utilizes the simulation module in order to derive the feasible manipulation paths for each hand-object combination, feeds these paths to an experimental setup that extracts manipulation data for training that include dynamic, hard to model phenomena (e.g., slipping) and employs appropriate clustering, regression and classification techniques in order to: 1) group together similar manipulation paths, 2) train task-specific manipulation models for all possible tasks, and 3) trigger a task-specific manipulation model based on the user provided task specifications.

This work was partially supported by the National Science Foundation grant IIS-1317976.

M. V. Liarokapis and A. M. Dollar are with the Department of Mechanical Engineering and Materials Science, Yale University, New Haven, CT, USA (e-mail: {minas.liarokapis, aaron.dollar}@yale.edu).
The idea of using simple, underactuated robot mechanisms for executing dexterous, in-hand manipulation tasks (see Fig. 2 and 3) is definitely not new. In [9], Bicchi summarized the evolution of robot hands and made a critical evaluation of the core ideas and trends, distinguishing between robot hands designed for mimicking the human anatomy (hands with anthropomorphic designs) and robot hands designed to be as functional and dexterous as possible. Moreover, Bicchi also presented a series of arguments favoring a “minimalistic” approach in designing robot hands (e.g., hands with a limited number of actuators and / or sensors).

Simple, adaptive robot hands have received an increased attention over the last years, with most studies focusing on new hand designs [1], [3], [6], [10], [11]. The particular class of hands, uses either tendon based or linkage based transmission and generally requires a spring element that acts as an antagonist to resist gravitational forces [12]. Such an elasticity is typically implemented using spring loaded pin joints or flexure joints based on elastomer materials (e.g., silicone or polyurethane sheets). Despite their promising performance, adaptive hands have also certain limitations and disadvantages. The use of underactuation and the structural compliance may facilitate the extraction of robust grasps, but complicates also modelling of the grasping and manipulation problems, making the derivation of analytical models challenging. As a result, nowadays, most researchers control adaptive hands in an open-loop fashion without performing any kind of grasping and / or manipulation synthesis. A prerequisite for formulating sophisticated control schemes for these hands that will allow us to execute meaningful manipulation tasks, is to accurately model their behavior while reaching, grasping and manipulating an object.

Recently, Odhner et al. proposed the smooth curvature model [13], [14], a methodology that provides an efficient representation of complex flexure joints based on elastomer materials, approximating them as elastic beams that bend smoothly. The smooth curvature model uses low-order polynomials that describe the curvature of these beams even for significant deformations and derives - in a computationally efficient fashion - appropriate homogeneous transformation matrices between the links of the robot as well as their derivatives (e.g., Jacobians, Hessians matrices etc.).

Subsequently, Odhner et al assessed the efficiency of adaptive, compliant and underactuated robot hands for executing robustly a wide range of grasping and dexterous manipulation tasks. In [15], they demonstrated an open-loop manipulation methodology for efficiently picking up small objects from a table surface (e.g., executing a flip and pinch task). The particular method was inspired by a human manipulation strategy and the efficacy of the proposed scheme was experimentally validated through extensive experimental paradigms that involved the Barrett WAM 7 DoF robot arm and a two finger, compliant, underactuated robot hand. In [16], they discussed the efficiency of the examined, differential-type, adaptive, underactuated robot hands in executing dexterous in-hand manipulation tasks, providing also evidence that the reduction of the number of actuators and constraints, actually makes the manipulation tasks easier to implement and control.

Regarding in-hand manipulation, in [17] Mason et al presented a methodology for performing a bin picking task with a general-purpose, simple, three-fingered, compliant robot hand that facilitates the execution of stable grasps even with irregular objects. The proposed methodology uses machine learning techniques and sensor data to perform object recognition and localization or to reject the grasp, using a minimalistic approach that the authors call “grasp first – ask questions later”. In [18] Dafle et al, proposed an in-hand manipulation methodology that relies on resources extrinsic to the hand (e.g., gravity, external contacts etc.). The particular methodology takes advantage of the aforementioned extrinsic dexterity concept, in order to facilitate the execution of dexterous, in-hand manipulation tasks with simple hands, using a set of re-grasping actions. All re-grasp actions were completely open-loop and hand scripted, but they were also surprisingly robust. In [19], Hoof et al used a reinforcement learning methodology to acquire in-hand manipulation skills for adaptive hands equipped with tactile sensors. The proposed methodology does not require analytical dynamic or kinematic models and the learned skills generalize to novel objects with a small loss in performance.
All these studies focused either on analytical or on machine learning methods for performing meaningful tasks with simple, adaptive hands, requiring in many cases some kind of feedback or sensing. Moreover, none of these studies provides an intuitive model for the user to plan the in-hand manipulation task or provide the required task specifications directly in the object space, e.g., “the object should move from a specific pose \( A \) to a desired pose \( B \)” (see Fig. 2 for details).

The rest of the paper is organized as follows. Section II focuses on the methods used and the schemes proposed, Section III presents results with various adaptive hands performing planar and 3D manipulation tasks and Section IV discusses the underlying problems of the analytical and the learning methods. Finally, Section V concludes the paper.

II. METHODS

In this section we describe the analytical, numerical and machine learning methods used in order to formulate the proposed hybrid methodology.

A. Modelling Adaptive Hands

For the simple case of a hand with spring loaded pin joints, the hand configuration at equilibrium can be found by minimizing the energy function:

\[ V(q) = \frac{1}{2} q^T K q \]

where \( q \) is the joint angles vector and \( K \) is the stiffness matrix that represents the compliances of the pin joints.

For the more complex case of flexure joints based on elastomer materials, we can use the smooth curvature model [13], [14] that provides computationally efficient estimations of their kinematics. The smooth curvature (SC) model is based on the assumption that elastic beams bend smoothly and that their curvature can be approximated by low-order polynomials (e.g., Legendre polynomials). In order to model the flexure joints, the SC model uses three generalized coordinates instead of one that is used for the spring loaded pin joints and provides a set of homogeneous transformations between the robot links.

In order to compute the configuration of the hand object system at equilibrium (the analysis is quasistatic) we need to formulate the problem as a constrained minimization of the potential energy of the hand system \( V(q) \) [14]. The final scheme can efficiently derive the finger poses relatively to the tendon displacements or loads, as well as to accurately represent the effect of any external forces acting on the system (e.g., contact forces). The equilibrium configuration of the hand object system can be found by minimizing:

\[ E(\tau) = -\nabla_q V(q) + J^T F \]

where \( \tau \) is the vector of the generalized forces acting on the system, \( F \) is the vector of the forces applied at a specific point \( p \) on the robot (e.g., contact points), \( J^T \) is the Jacobian of the particular point coordinates and \( \nabla_q V \) is the gradient of the total internal energy of the robot. An analytical description of the constraints that should be incorporated in the formulation of the energy minimization problem, as well as a stability measure that can be used to interrupt the exploration of low-quality manipulation tasks, can be found in [16].

B. Deriving Optimal, Stable Grasps

The process of securing a grasp with an underactuated robot hand is decomposed in two different steps. The first step is to compute an appropriate wrist offset for prepositioning the robot arm and the second step is to apply appropriate motor loads in order to reach the desired contact points and grasp firmly by applying appropriate contact forces.

In order to compute the wrist offset, we start by applying an increasing equal load to both motors until the solution given by the forward kinematics (FK) of the smooth curvature model [14] dictates that the distance between the fingertips is equal to a dimension of the object (e.g., the grasping side of a rectangle). Then, the distance between the fingertips of the robot hand and the robot arm wrist is the required wrist offset from the object center of mass. Having prepositioned the arm, we only need to apply the computed motor loads, which will drive the fingers to the desired contact points, optimizing also a grasp quality metric [20]. The metric chosen in this study is the distance between the contact centroid \( (\mathbf{c}_c) \) and the object geometric centroid \( (\mathbf{c}_g) \). When this distance is zero the grasp is optimal. The distance is given by:

\[ Q_{DC} = \| \mathbf{c}_c - \mathbf{c}_g \| \]

The maximization of the particular grasp quality metric leads to a better grasp that tends to be more stable and a good candidate for initializing a manipulation task. More details regarding the usage of the grasp quality metrics and the applicability of a grasping force optimization procedure, can be found in [21], [22].

C. A Simulation Module

The proposed simulation module uses the constrained energy minimization scheme and predicts in a feed forward manner from arbitrary motor displacements or loads and hand object interactions (e.g., contact forces, contact rolling etc.) the new hand object configurations. The only problem is that most arbitrary tendon object configurations lead eventually to a loss of grasp. To confront this problem, we use a manipulation workspace exploration algorithm that exhaustively searches the hand object system configuration space by randomly perturbing the motor positions, eliminating the unstable grasps or those manipulation trajectories that are too short to be meaningful. For doing so, the particular algorithm takes as input two different thresholds, the stability and the trajectory length thresholds. In case a grasp is less stable that the provided threshold, the examined manipulation trajectory is terminated. If the length of the terminated trajectory is less than the length threshold, then the entire trajectory is omitted.

The simulation module allows us to test and explore the manipulation capabilities of any hand object system without relying on time-consuming experiments with the real robot. It must be noted that the simulation module requires extensive knowledge of the robot hand parameters (e.g., joint stiffness, link lengths, finger-pads compliances) as well as of the object parameters (e.g., object pose, object weight, object stiffness etc.). Moreover, the simulation module provides intuition about the problem mechanics, since by altering the different hand or object parameters we can examine what is their effect in the manipulation workspace or in the execution of meaningful manipulation tasks.
D. Features Selection and Trajectories Annotation

As it can be noticed in Fig. 2, the manipulation problem can be defined as a fingertips driven object pose perturbation from current pose A to a desired pose B. The definition of dexterous manipulation according to Bicchi [9] is: “(The) capability of changing the position and orientation of the manipulated object from a given reference configuration to a different one, arbitrarily chosen within the hand workspace”. For this reason, we annotate each manipulation trajectory with a vector containing the pose of the object and the motor positions of the hand at two time instances, at the beginning and the end of the manipulation process. Such a choice is supported by the fact that we deal with underactuated mechanisms that have a constrained and task-specific, feasible manipulation workspace (a sub-manifold of the entire hand workspace). For more complex robot hands, multiple, task-specific trajectories may exist between two different object poses, requiring the use of more time instances.

E. Formulating the Learning Scheme

In order to formulate the proposed learning scheme (see Fig. 4), we use different clustering, regression and classification techniques, as follows.

1) Classification and Regression Techniques

In order to train appropriate classifiers and regressors and since the examined problems are multi-class and multi-dimensional, we use the Random Forests classification and regression techniques.

The Random Forests is an ensemble classifier that consists of multiple decision trees (see Fig. 5) and was initially proposed by Tin Kam Ho [23] and Leo Breiman [24]. The final decision of the forest is always the most popular class among the individual classifiers (the different trees of the forest). Random forests are efficient and fast on large databases, provide high classification accuracy, can handle thousands of input variables without deletion and are particularly efficient for multi-class problems and multi-dimensional spaces. In case of regression, the models are trained with continuous response variables as outputs, instead of the categorical response variables (classes). The regressor is used to train all the identified, task-specific manipulation models and the classifier is used to trigger the appropriate task-specific manipulation model based on the user input.

2) Clustering Technique

In order to group together similar manipulation trajectories we perform clustering of the manipulation data using the annotated features that we presented in Subsection II-D. More specifically the clustering technique used in this paper is the k-means algorithm. The k-means is a vector quantization method that partitions n observations into k clusters, so as for each observation to belong to the cluster with the nearest mean. In our case, k-means is used in order to group together similar, feasible manipulation paths / trajectories by deriving appropriate clusters based on aforementioned annotated feature variables.

3) Outline of the Learning Scheme

The learning scheme consists of two separate modules (see Fig. 4), the offline training phase and the online execution phase. For the offline training phase, the learning steps are:

- The simulation module uses the manipulation workspace exploration algorithm in order to extract the feasible manipulation paths for a given hand object configuration.
- The feasible manipulation paths are fed to the automated data collection setup that provides real manipulation data for training that include also dynamic, difficult to model phenomena (e.g., slipping).
- The raw object and motor trajectories (manipulation data) are annotated with the aforementioned feature variables and are clustered to similar hand, object and grasp specific trajectories (task-specific trajectories).
The Random Forests regression technique is used in order to train manipulation models for all possible tasks. These models are equivalent to the inverse of the Hand Object Jacobians (in the motor positions space).

For the online execution phase the steps are:
- The user provides a desired hand-object configuration.
- The user input is classified to the most similar path.
- A task-specific manipulation model is triggered.

### III. RESULTS

In this section we present the apparatus as well as experimental results and paradigms with two-fingered and three-fingered adaptive hands, executing planar and 3D, in-hand manipulation tasks.

#### A. Apparatus

In this section we present the apparatus used in order to conduct the automated data collection experiments and extract the required manipulation data for the training of the task-specific manipulation models.

1) Adaptive, Underactuated and Compliant Robot Hands

In this work, we use four open-source, adaptive, underactuated and compliant robot hands based on flexure joints or spring loaded pin joints. The employed hands have been developed within the Yale Open Hand project and their designs are freely distributed through the project’s website (www.eng.yale.edu/grablab/openhand/).

It must be noted that each robot finger has a dedicated motor. Model O has an extra motor for the coupled abduction/adduction DoF of the two rotating fingers.

2) Robot Arm

In order to automate the data collection procedure, we use a redundant 7 DoF robot arm (WAM, Barrett Technology). The employed arm facilitates reaching and grasping of the different objects that are located in the experimental workspace (see Fig. 7), without requiring a human presence. Such a setup can conduct numerous experimental trials without supervision, letting the system explore the manipulation capabilities of each hand, in an exhaustive manner. The Barrett WAM robotic manipulator is depicted with the model T42FF attached at the end-effector, in Fig. 8.

![Fig. 6. The examined adaptive robot hands of the Yale Open Hand project. The depicted robot hands are fully open-source and instructions for their replication can be found at the project’s website (www.eng.yale.edu/grablab/openhand/).](image)

![Fig. 7. The experimental setup used for the automation of the data collection procedure. The different model objects used are depicted together with a frame that allows them to be positioned in a free floating pose in 3D space and return passively to the same pose, upon release of the grasp. The frame was built using a set of T-slotted profiles of the Industrial Erector Set (80/20).](image)

![Fig. 8. The Barrett WAM 7 DoF robot arm with the model T42FF (with 2 flexure joints per finger) attached.](image)
3) Motio...red manipulation tasks with a 3D printed cylinder (diameter of 20 mm). The different colors denote the different finger motors, while the continuous lines denote the actual positions and the dashed lines the estimated motor positions. Data for a single trial are presented.

4) Model Objects

The model objects used for the experiments are 3D printed hollow spheres and cylinders that have an appropriate sheaths for mounting the magnetic motion capture system sensors. More precisely, spheres and cylinders with diameters of 30, 35, 40, 45, 50, 55 and 60 mm were used. The weight of each object was in the range of 6 gr and 18 gr for the spheres and 12 gr and 25 gr for the cylinders. The sensors were fixed in the centers of the model objects using appropriate tape and hot glue. For the analysis we used a friction coefficient of \( \mu = 0.6 \) (friction between the finger-pads of the fingers - made using Vytaflex 40 Smooth-On urethane rubber - and the ABS of the 3D printed objects).

5) Automated Data Collection

The steps followed by the automated data collection procedure, were the following: 1) the robot arm reaches an object specific pre-grasp configuration, which has a certain wrist offset from the object center of mass, 2) hand closes the fingers so as to reach the desired contact points, 3) hand achieves a stable grasp and executes a random, feasible manipulation task, 4) hand releases the object, 5) the object returns passively to its initial pose and the robot arm to the starting configuration.

All the steps are repeated until all feasible manipulation tasks have been executed a predefined number of times. The extracted manipulation data, include also dynamic phenomena (e.g., slipping) that are hard to model and they are used by the proposed learning scheme for training.

B. Estimating Motor Positions for Planar Tasks

In this subsection we evaluate the efficiency of the task-specific manipulation models in predicting accurate motor positions from object trajectories, for the case of planar in-hand manipulation tasks executed with Models T42PP and T42PF. More precisely, the manipulation tasks where equilibrium point manipulations of a 3D printed cylinder and the task-specific manipulation model acts as the inverse of the Hand Object Jacobian (in the motor positions space).

Results for single trials that were not previously seen during training (validation data) are depicted in Fig. 9 for Model T42PP and in Fig. 10 for Model T42PF. As it can be noticed, the T42PP results are slightly better, as we use more accurate motors for the particular hand, which actuate the fingers in a smoother fashion (Dynamixels MX28 instead of Dynamixels RX28).

C. Estimating Motor Positions for 3D Tasks

In this subsection we evaluate the efficiency of the task-specific manipulation models in predicting accurate motor positions for executing 3D in-hand manipulation tasks with the Model O of the Yale Open Hand project [25]. The manipulation task was a random, multi-directional equilibrium point manipulation of a 3D printed sphere. Results for Model O are depicted in Fig. 11. It is certainly not a surprise that the estimation accuracy drops for Model O, since certain dynamic phenomena (e.g., uncontrolled slipping and rolling) are much more intense in the 3D case. Despite the deterioration of the performance, the estimations are quite accurate also in the 3D case.
D. The Effect of Slipping

As we have already noted, in this paper we feed the feasible manipulation paths that we extract from the simulation module to the actual experimental setup, in order to collect manipulation data that contain dynamic phenomena like slipping. The motivation behind this choice is that a manipulation model trained with such data will be able to account also for such difficult to model phenomena that have not been included in the simulation process.

But what happens when our method faces phenomena, that have not been “seen” during training? In order to investigate the possible effect of such a case, we excluded from the training data those trials that had uncontrolled slipping (in a supervised fashion) and we trained the manipulation models with the remaining data. Then, we provided as an input to the trained models, an object trajectory that included uncontrolled slipping (a trial of those previously omitted) and we compared the estimated motor position trajectories with the actual trajectories. Results are depicted in Fig. 12. Although, “unseen” slipping phenomena deteriorate the estimation accuracy, the trained models still manage to provide a reasonable performance.

E. Generalization to New Objects

In case that the system must manipulate a novel object not previously seen during training, then the proposed scheme can skip the experimental data step and use directly the data derived from the simulation module to train the task-specific manipulation models (as no experimental data are available for the new objects). Of course, these models will not account for dynamic phenomena like slipping and will be more prone to errors, providing a deteriorated motor positions estimation accuracy. It must be noted that in all cases we hypothesize that we have an analytic description of the object parameters (e.g., weight, stiffness etc.) and geometry (required by the simulation module).

F. Evaluation of the Motor Positions Estimation Accuracy

In order to evaluate the efficiency of the proposed task-specific manipulation models for planar and 3D, in-hand manipulation tasks executed with various adaptive hands, we used data not seen during training (validation data) and we compared the actual and the estimated motor trajectories for the same task. In order to represent the similarity between the actual and the predicted motor trajectories as a percentage, we used the following metric:

$$ S = 100(1 - \text{RMS}(p_a - p_o)/\text{RMS}(p_o)) \% $$  

where $p_a$ and $p_o$ are the actual and the estimated motor positions respectively. The trajectory similarities for planar manipulation tasks executed with Models T42PP, T42PF and T42FF as well as the for a 3D manipulation task executed with Model O are contained in Table I. It must be noted, that the similarity scores, are the average values over the multiple rounds of the 10-fold cross-validation method used. The training data contained 10 manipulation trials with the same object. More precisely, cylinders with a diameter of 20 mm and 60 mm were used for Model T42PF and Model T42PF respectively, while a sphere with 40 mm diameter was used for Model O. The motor positions for all trials are normalized.

<table>
<thead>
<tr>
<th>Model</th>
<th>Similarity Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>T42PF</td>
<td>98.84%</td>
</tr>
<tr>
<td>T42PF</td>
<td>99.44%</td>
</tr>
<tr>
<td>O</td>
<td>96.01%</td>
</tr>
<tr>
<td>SD</td>
<td>0.15%</td>
</tr>
<tr>
<td>SD</td>
<td>0.06%</td>
</tr>
<tr>
<td>SD</td>
<td>1.35%</td>
</tr>
</tbody>
</table>

G. Experimental Paradigms

Various experimental paradigms with Models T42PP, T42PF, T42FF and Model O of Yale Open Hand project [25], performing planar and 3D in-hand manipulation tasks are included in the accompanying video.


IV. DISCUSSION

The major problems of the analytical methods are: 1) the difficulties we face modelling certain phenomena like the contact slipping and rolling, 2) the inaccuracies in deriving certain system parameters (e.g., friction coefficient, stiffness of flexure and pin joints, fingertips compliance etc.). The obvious solution for these problems is an exhaustive system identification to derive all missing terms.

The main source of errors for the learning methods is the data quality and availability. Experiments with the particular class of hands can be easily conducted, but their behavior is not always repeatable. As a result, the data used for training should be “big” enough to account for all possible scenarios and behaviors. Such errors can be easily resolved by pursuing more robust hand designs.

V. CONCLUSIONS

In this paper we presented a learning scheme for deriving task-specific models for dexterous in-hand manipulation with adaptive hands. In this respect, a simulation module was synthesized using a combination of numerical and analytical methods that predicts the kinematics of adaptive hands and performs a quasi-static modelling of the manipulation problem. The proposed machine learning scheme was used in order to split the problem space and derive task-specific manipulation models that account also for difficult to model, dynamic phenomena like slipping. The efficiency of the proposed method was experientially verified with various adaptive, underactuated and compliant robot hands.

REFERENCES