2013 IEEE/RSJ International Conference on
Intelligent Robots and Systems (IROS)
November 3-7, 2013. Tokyo, Japan

Enhanced Kinematic Model for Dexterous Manipulation with an Underactuated Hand

Yanyu Su[1], Yan Wu[2], Harold Soh[2], Zhijiang Du[3] and Yiannis Demiris[2]

Abstract— **Recent studies on underactuated manipulation usually describe the system with a *Kinematic Model (KM)*, which is built by adding external constraints to the standard manipulation analysis method. However, such external constraints are easily violated in a real-world dexterous manipulation task which results in significant control errors. In this work, the *Enhanced Kinematic Model (E-KM)*, an integrated model of the KM and the *Sparse Online Gaussian Process (SOGP)* is proposed. The E-KM can compensate the shortfalls of the KM by on-the-fly training the SOGP on the residual between the prediction of the KM and the ground truth data. Based on the E-KM, we further contribute an optimal controller for underactuated manipulations. This optimal E-KM controller is implemented and tested on the iCub, a humanoid robot with two anthropomorphic underactuated hands. Two sets of real-world experiments are carried out to verify our method. The results demonstrate that the controller using E-KM statistically can achieve higher control accuracy than using solely using the KM for a wide range of objects.**

I. Introduction

We address the problem of in-hand manipulation with an underactuated robot hand in the real-world environment, where a robot faces objects of different shapes and weights [1]. One main challenge of this problem is the difficulty of building kinematic model of the underactuated manipulation.

The underactuated hand, by definition, has fewer actuators than joints. This feature enables it to adapt to a wide variety of objects without a complex control structure [2]. However it also makes the standard analysis method of manipulation, such as [3], fail to describe the manipulation.

As discussed in [4], the standard analysis method of manipulation determines the motion of the actuators for realising a desired motion of the object by three constraint matrices, a *grasp matrix*, a *hand Jacobian* and an *actuator Jacobian*. But the *actuator Jacobian* of the underactuated hand, which maps the actuator space to the joint space, is usually rank-deficient and non-invertible since the number of actuators are smaller than that of joints. Therefore the motion of actuators to realise a desired motion of the object is indeterminable by the standard analysis method, unless additional constraints can be introduced, for example, forming a power grasp to gain more contact constraints.

Recent studies on underactuated manipulation usually use additional constraints to build the full-rank actuation Jacobian. For example, [4] uses the principle of elastic averaging, [5] uses the assumption of minimal contact force. However, applying these methods in the real-world environment faces a bottleneck - the additional constraints are easily violated due to the uncertainty of system, such as the change of the mechanics, the disturbance from the environment and so on. As a result, the errors of such kinematic model are often significant.

To compensate for these errors, there are at least two directions of research in robotics: 1) Use a learning technique to replace the entire analysis model [6], [7]; 2) Model the residual error of the analysis model [8], [9]. Although the prior approach produced some successful examples, however, its learning complexity scales drastically with the number of joints involved.

On the other hand, to our knowledge, no literature uses the latter approach to design a compensated model for underactuated manipulation. A similar approach has been applied on identifying the dynamic model of the autonomous blimp [10]. However a main difference is that our system continuously improves its internal model of familiar objects and create models for new objects. This feature is core for the underactuated manipulation, as the robot has to rapidly adapt to new objects in a real-life environment.

To improve the accuracy of the underactuated manipulation using the compensation approach, the selected model should have an online feature so it can adapt to system changes, such as replacing the driving tendons or adapting to a new object. It should also be capable of modelling the uncertainty of actuation errors which can be used as an input parameter to the controller; the computational complexity should be as minimal as possible so it can be used in a real-time controller.

In this work, we propose a *Enhanced Kinematics Model (E-KM)* for Underactuated Manipulation. To build the E-KM, we use the *Sparse Online Gaussian Process (SOGP)* to model the residual between the prediction of the *Kinematic Model (KM)* and the ground truth data. The output of the SOGP is used to compensate the KM on-the-fly.

To control the robot in real action, a controller has to be constructed for action execution. Most reported underactuated manipulation controllers follow the plan-execution control pattern, such as [4] and [11], which might not easily adapt to the change of the system for real-world application.

[1] Yanyu Su is jointly with the State Key Laboratory of Robot and System, Harbin Institute of Technology, China and the Electrical and Electronic & Engineering Department, Imperial College London, United Kingdom.suyanyu at hit.edu.cn

[2] Yan Wu, Harold Soh and Yiannis Demiris are with Department of Electrical & Electronic Engineering, Imperial College London, United Kingdom.yan.wu08, h.soh09, y.demiris at imperial.ac.uk

[3] Zhijiang Du is with the State Key Laboratory of Robot and System, Harbin Institute of Technology, China. zjdu01 at hit.edu.cn

978-1-4673-6357-0/13/$31.00 ©2013 IEEE

To overcome this shortage, we propose a dynamic image-based look-and-move controller of underactuated manipulation using the E-KM as the core model for action generation.

The rest of the paper is organised as follows: In Section II we will present the details of the proposed E-KM. This is followed by a description of the implemented controller on the iCub humanoid robot in Section III. Two sets of experiments are detailed in Section IV. The respective results discussed in Section V before we conclude for future work in Section VI.

II. The Enhanced Kinematic Model for Underactuated Manipulation

In this section, we present the Enhanced Kinematics Model for Underactuated Manipulation, an integrated model to improve the dexterous manipulability of an underactuated robot hand. This model compensates the shortfalls of the kinematic analysis approach using a machine-learning technique.

A. Problem Formulation

We consider the general problem of improving a Kinematic Model for underactuated manipulation using a machine-learning method. In this work, the following assumptions are made: the objects have been pinch-grasped before the manipulation; the manipulation follows all quasi-static assumptions; the displacements of the joints and the pose of the objects are fully observable; the pose of the hand remains static during manipulation to avoid changes in gravitational force; the actuators are driven by a position controller which takes the input of the incremental position signals.

Under the quasi-static assumptions, to fully describe the system of an underactuated manipulation, the state vector

$$s = [q^T \quad p^T]^T, \tag{1}$$

consists of the vector of the angular displacements of the joints $q = [q_1 ... q_{n_q}] \in \mathbb{R}^{n_q}$ and the vector describing the position and the orientation of the object $p = [p_1 ... p_{n_p}] \in \mathbb{R}^{n_p}$ with n_q and n_p denoting the number of joints and the number of degrees of freedom (DoFs) of the object.

In a KM, the prediction of differential increment of the current state can be expressed as

$$\delta \hat{s_0} = f(s, \delta a), \tag{2}$$

where $\delta \hat{s_0}$ is the vector of the predicted increment of the state, $\delta a = [\delta a_1 ... \delta a_{n_a}] \in \mathbb{R}^{n_a}$ is the vector of the differential motions of the actuators with n_a denoting the number of actuators.

As discussed in the introduction, the outputs of a KM usually have significant errors, because it cannot modelled the uncertainties. To tackle this problem, we propose compensating the output of a KM using a machine learning method

$$\delta \hat{s} = f(s, \delta a) + g(s, \delta a), \tag{3}$$

where $\delta \hat{s}$ is the compensated prediction of the increment of the state, $g(s, u)$ is the target function of a machine learning method that predict the output error of a KM.

The underlying machine-learning technique in $g(s, u)$ should possess the following properties: it should have an online feature so it can update when the system changes; it should also be capable of modelling the uncertainty of actuation errors; its computational complexity should be as minimal as possible so it can be used in a real-time controller.

In the following sections, we will detail our proposed E-KM which consists of two modules, the KM and the Sparse Online Gaussian Process (SOGP).

B. The Kinematic Model of Underactuated Manipulation

We adopt the quasi-static kinematic analysis approach for underactuated manipulation proposed by Odhner and Dollar [4]. Its result has the form

$$\delta \hat{s_0} = f(s, \delta a) = J_m \, \delta a, \tag{4}$$

where J_m is the Manipulation Jacobian

$$J_m = RK^{-1}R^T \begin{bmatrix} J_a^T \\ 0 \end{bmatrix} \left([J_a \quad 0] RK^{-1}R^T \begin{bmatrix} J_a^T \\ 0 \end{bmatrix} \right)^{-1},$$

where R is the nullspace basis of the Manipulation Constraint Matrix, J_a is the Actuation Jacobian, K is a matrix describing the stiffness of the system

$$K = R^T \begin{bmatrix} V_{qq} + S & 0 \\ T & V_{pp} \end{bmatrix} R,$$

where V_{qq} and V_{pp} are the second order partial derivatives of the potential energy of the system of underactuation manipulation. S and T are defined as

$$S_{ij} = \sum_k \frac{\delta J_{h,kj}}{\delta \theta_j} \mu_k + \sum_l \frac{\delta J_{a,li}}{\delta \theta_j} \lambda_l,$$
$$T_{ij} = \sum_k \frac{\delta G_{ik}}{\delta \theta_j} \mu_k,$$

where J_h is the Hand Jacobian, G is the Grasp Matrix, μ_k and λ_l are the elements of the contact force μ and actuator force λ.

Several components are needed to solve (**??**), such as the Manipulation Constraint Matrix, the Actuation Jacobian, the Contact Force and the Actuation Force. The methods for solving these components varies in literatures, but we adopt the standard methods in this paper for the generalizability.

1) Contact Constraint: The contact constraint, which describes the relative differential motion of the joints of the hand δq and the differential motion of the object δp, can be accounted using the standard manipulation analysis method descried in [3]. It can be expressed as

$$[H\tilde{J}_h \quad -H\tilde{G}^T] \begin{bmatrix} \delta q \\ \delta p \end{bmatrix} = 0, \tag{5}$$

where $H \in \mathbb{R}^{n_h \times 6n_c}$ is the matrix of contact constraints, $\tilde{G} \in \mathbb{R}^{n_p \times 6n_c}$ is the Full Grasp Matrix, $\tilde{J}_h \in \mathbb{R}^{6n_c \times n_q}$ is the Full Hand Jacobian, with n_c and n_h denoting the number of the contact points and the number of contact constraints. (**??**) is commonly denoted as

$$[J_h \quad -G^T]\begin{bmatrix} \delta q \\ \delta p \end{bmatrix} = 0,$$

where the first term on the left is the Manipulation Constraint Matrix, $J_h = H\tilde{J}_h$ is the Hand Jacobian, $G = \tilde{G}H^T$ is the Grasp matrix. It is practical to solve the Hand Jacobian J_h and the Grasp Matrix G with grasping simulators such as the GraspIt [12] and the Simox Grasp Studio [13].

2) Actuation Jacobian: The differential motion of joints δq is related to the differential motion of the actuators δa by

$$\delta a = J_a \delta q, \tag{6}$$

where $J_a \in \mathbb{R}^{n_a \times n_q}$ is the Actuation Jacobian, with n_a denoting the number of actuators. The Actuation Jacobian J_a is determined by the mechanical configuration of the underactuated joints of the robotic hands.

3) Contact Force and Actuation Force: The contact force $\mu \in \mathbb{R}^{n_h}$ and the actuation force $\lambda \in \mathbb{R}^{n_a}$ can be accounted by solving the equation derived from the principle of virtual works

$$\begin{bmatrix} V_q \\ V_p \end{bmatrix} + \begin{bmatrix} J_h^T \\ G \end{bmatrix}\mu + \begin{bmatrix} J_h^T \\ 0 \end{bmatrix}\lambda = 0, \tag{7}$$

where V_q and V_p are the partial derivatives of the potential energy of the system. The contact force $\mu \in \mathbb{R}^{n_h}$ are expressed in the contact frames consisting of at least the normal pressure force of the contact points.

For our problem, the potential energy of the system $V(q, p)$ consists of the elastic and the gravitational potential energies

$$V(q,p) = \frac{1}{2}q^T K_s q + mgp_g(p), \tag{8}$$

where K_s is the matrix of the spring stiffness, g is the gravitational acceleration, m and $p_g(p)$ are the mass and the centre of gravity of the object, which is a function of p.

Based on the minimal contact force principle presented in [5], we solve the contact force μ and the actuation force λ from (**??**) as an optimisation problem that minimises the contact force u under three constraints: the principle of virtual works, described in (**??**); the normal pressure force, that is the normal pressure should be no less than zero; and the maximum static friction, which limits the other elements of the contact force u besides the normal pressure.

C. Learning the Error of the Kinematic Model

We propose to model the residual error of the KM as a Gaussian Process (GP). For given elements $x \in \mathbb{X}$, a GP is specified by its mean function,

$$m(x) = \mathbb{E}[g(x)],$$

and its covariance function

$$k(x,x') = \mathbb{E}[(g(x) - m(x))(g(x') - m(x'))],$$

In contrast to other regression methods, GP regression (GPR) provides predictive distributions (instead of point predictions) and is able to learn the output noise from training data through maximum-likelihood maximisation. These features make GPR attractive for compensating the error of the kinematics analysis models [10].

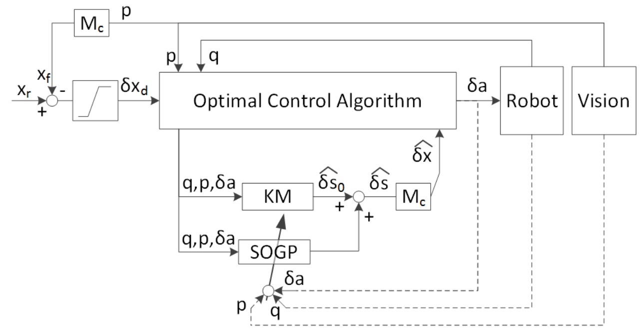

Fig. 1: The proposed optimal controller for underactuated manipulation using E-KM. The controller makes use of the dynamic image-based look-and-move strategy. The dotted lines shows the learning phasse.

Note that it is typical to the mean function of the GP to be zero, $m(x) = 0$, yielding a Gaussian Process of the form $GP(0, k(x,x'))$. Our approach of modelling the residuals of the KM can be interpreted as incorporating a fixed deterministic mean function $m(x) = f(x)$ into the GP.

One key drawback of the full GP implementation is the computational cost of the model ($O(n^3)$ for training). In this work, we have used the Sparse Online Gaussian Process (SOGP) [14]; an approximation of the full GP which has a lower computational complexity, $O(|B|)$ where B is the basis vector set (the retained training samples). Note that the maximum size of B can be constrained, effectively rendering the SOGP a constant time algorithm. Readers desiring more detail can find descriptions of the SOGP in [14], [15].

Here, we used the SOGP to learn and predict the residual error of the incremental object movement between the actual displacement and the KM output. The training data for our model is therefore:

$$\begin{aligned} x_t &= [s,u]^T = [q^T, p^T, \delta a^T]^T, \\ y_t &= \delta s - f(s,u) = \delta s - J_m \delta a, \end{aligned} \tag{9}$$

where x_t is the input and y_t is the output (or target).

III. Underactuated Manipulation Controller

In this section, we present an optimal controller for underactuated manipulation using the E-KM model with an optimal control algorithm.

A. Overview

Because of the limited workspace, most underactuated manipulation controllers do not take into account of all the possible DoFs of the objects [4], [11]. For this reason, we use a subset of the DoFs of the object $x \in \mathbb{R}^{n_x}$ with $n_x \leq n_p$ to describe the pose of an object, called the focused pose. The object pose p can be mapped to the focused pose x using a matrix $M_C \in \mathbb{R}^{n_x \times n_p}$

$$x = M_C p.$$

The framework of an underactuated manipulation controller generally follows the dynamic image-based look-and-move structure, as shown in Fig.1. Given the input of the desired pose of the object x_r, the controller sends the control signal, the differential movement of the actuators δa, to a position controller of the robot hand.

In the control loop, the desired differential movement of the object δx_d is gained by applying a limiting function for the control error

$$e = x_r - x_f,$$

where x_f is the current pose of the object obtained from the vision system.

The optimal control algorithm, which will be described in the Section III-B, takes the control step δx_d, the displacements of the joints q_f and the object pose x_f to account for the control signal δa, using a forward kinematic model to predict δx_d. The forward kinematic model used in our controller is the E-KM.

In the learning phase, the differential increment of the state δs is gained by

$$\delta s(t-1) = \begin{bmatrix} q_f(t) - q_f(t-1) \\ p_f(t) - p_f(t-1) \end{bmatrix},$$

where $t \geq 1$ is the index of the control loop, p_f is the pose of the object.

The differential increment of the state $\delta s(t-1)$ along with the actual command $\delta a_c(t-1)$, the manipulation Jacobian $J_m(t-1)$, the position of the joints $q_f(t-1)$ and the pose of the object $p_f(t-1)$ are substituted to (**??**) to generate the training data for SOGP.

B. Optimal Control Algorithm

The aim of the optimal control algorithm is to solve δa, by executing which the hand will move the object by δx_d. This is an inverse kinematics problem. However, since the number of DoFs of an underactuated manipulation system is usually greater than that of the focused ones, it can be formalised in the form of constrained optimisation problem

$$\arg\min_{\delta a} c(\delta a) \tag{10}$$

subject to

$$\begin{aligned} a_{min} \leq a + \delta a \leq a_{max} \\ x_r - [0 \quad M_c]\delta s = 0 \end{aligned}$$

where δs can be accounted by (**??**).

As determined by the definition of the differential motion, the δa should be as minimal as possible. Therefore we define the cost function $c(\delta a)$ as

$$c(\delta a) = \delta a^T W \delta a, \tag{11}$$

which is minimum when the δa keeps the best of differential motion best. The behaviour can be adjusted by the weight matrix $W \in \mathbb{R}^{n_a \times n_a}$.

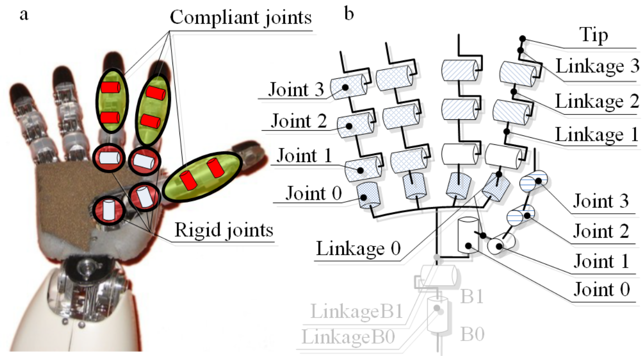

Fig. 2: The robot plantform iCub hand. A labelled picture of the right hand of the iCub (a) and the corresponding schematics of the hand (b), where the coupled joints are marked with the same patterns.

C. The Implementation

This controller is implemented and tested on the iCub humanoid robot developed by the RobotCub Consortium. The iCub has two 7-DoF arms, each is attached with an anthropomorphic underactuated hand with 9 actuators ("iCub-hand", shown in Fig. 2a).

The iCub-hand has 20 joints, some of which are coupled and underactuated including the distal joints of all fingers. This design, shown in Fig. 2b, enables the phalanges of these fingers to possess compliant characteristics. An onboard PID position controller is provided to drive the actuators. All the fingers are driven by tendons, which may not afford hundreds of manipulations. The transmission dynamics of the fingers change after each time a tendon is stretched or replaced. This makes the iCub-hand a very suitable application of the proposed controller with online learning capabilities.

1) E-KM: A manipulation simulator is built to generate the contact information (i.e. contact points, normal vectors of contact faces). The simulator takes the object pose and the displacements of the joints to calculate the contact information with the collision detection algorithm proposed in [16]. The contact information is then used to calculate the manipulation Jacobian (**??**). The compensation method discussed in Section II-A is then applied.

2) Vision Tracking System: The poses of the hand and the objects are estimated using the marker tracking method presented in [17] with a VGA camera operating at $20Hz$. The markers are attached to the palm of iCub-hand and the objects, as shown in Fig. 3b & 3e. Calibration of the camera is carried out before experiments to reduce tracking error along the principal axis of the camera.

3) Joints State Observation: The displacement of underactuated joints could be estimated using the Hall-effect sensors. We calibrate the sensor readings (ranged from 0 to 255) against the actual rotation angles (with the range of $0-90$ *deg*) using the parametric fitting method with cubic polynomial regression.

4) Position Controller: All the grasping and manipulations are executed using our previously developed controller

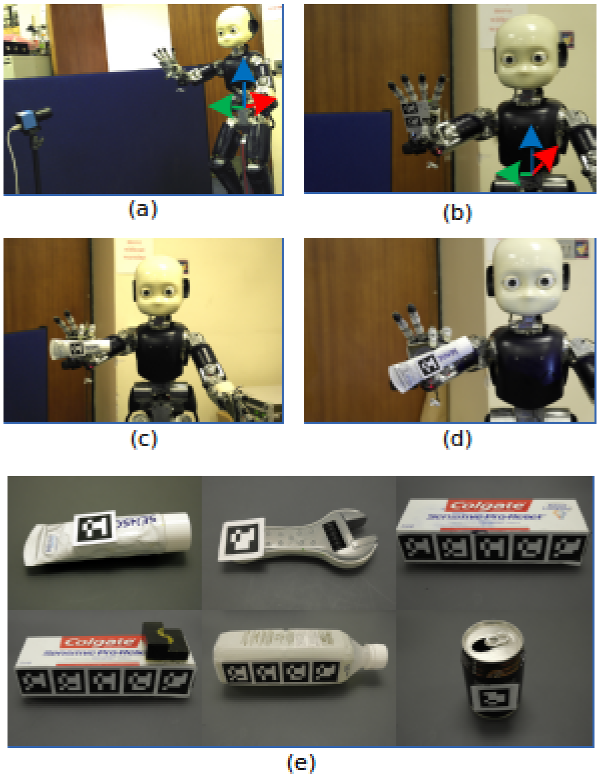

Fig. 3: The experiment setup. (a) A VGA camera is positioned in front of the iCub to obtain the pose of the objects and iCub-hand. (b) A marker is attached to the palm of the iCub-hand for calibration before the start of experiments. (c) shows the initial position of the object after it is pinched grasped. (d) shows an instance of the object after the manipulation. (e) shows the objects used in the experiments: toothpaste tube, toy applier, rectangular box, box with counter-weight, plastic bottle and can. The root coordinate system of the iCub is indicated in (a) and (b) with the red, green and blue axis correspond the x, y and z axis.

[18]. The controller sends position command to the interface of the iCub position controller. The iCub position controller drives the actuators with a control error of ± 1 *deg*.

IV. Experiments

Two different set of experiments are conducted to investigate how the E-KM model improves the underactuated manipulation (Online Manipulation Investigation Experiment), and evaluate the generalisability of the control method (Generalizability Investigation Experiment).

A. Experiment Setup

The experiment setup is shown in Fig. 3. Six different objects are used in our experiments (Fig. 3e). We used the previously implemented grasp controller [18] to execute a pinch grasp of the object as illustrated in Fig. 3c. The pose of the iCub-hand remains static throughout all experiments.

The in-hand manipulation is performed using our proposed controller to adjust the object to the required pose (Fig. 3d). Limited by the workspace of the iCub-hand, all manipulations are rotations along the x-axis of the robot frame of reference as shown in Fig. 3a.

TABLE I: The rotation manipulations for testing the controller (unit: *deg*)

Trial	Box	Weighted box	Can	Bottle	Applier
1(20%)	9	6	9	7	3
2(40%)	14	11	13	13	8
3(60%)	19	16	23	18	13
4(80%)	24	21	31	23	18
5(100%)	29	26	38	28	23

B. Online Manipulation Investigation Experiment

In this experiment, we carry out 45 rotation trials to manipulate a toothpaste tube (Fig.3e) to investigate how the SOGP can enhance the output of the KM in an online fashion. The grasp points and rotation angles are randomly chosen within the workspace.

C. Generalizability Investigation Experiment

To verify the generalisability of the E-KM model, we use a further set of five daily objects to perform the rotation manipulations (Fig. 3e). These objects are selected to represent the objects of different geometries and weight distributions.

In this experiment, all objects are pinch grasped at their geometric centres. A set of rotation manipulations are carried with the full controller. We then manually turn off the SOGP update and perform another set of rotation manipulations to benchmark the performance difference between KM and E-KM. The set of rotation manipulations for benchmarking the controller, tabulated in TABLE I, consists of the upper bound of the workspace and 4 other positions randomly selected at approximately 20%, 40%, 60% and 80% of the upper bound with a variance of ± 2 deg.

D. Performance Evaluation Metrics

We introduce three performance metrics to evaluate the performance of the controller quantitatively.

1) Prediction Error: The prediction Error of the E-KM e_{ekm} and the KM e_{km} can be measured using the difference between their predictions and the actual pose of the objects

$$e_{ekm}(t) = x_f(t+1) - J_m(t)\delta a(t),$$
$$e_{km}(t) = x_f(t+1) - J_m(t)\delta a(t) - g(q(t), p(t), \delta a(t)),$$

where t and $t+1$ indicate the current and next time step.

2) Static Error: The static error of the controller is a commonly used metric for controllers. In this paper, we calculate the static error e_s by calculating the difference between the desired pose and the actual pose after the control terminates.

$$e_s = x_r - x_f,$$

3) Hypothesis Testing: To statistically evaluate the hypothesis that E-KM improves the performance of underactuated manipulation over KM, we applied the standard two-sample t-test on the static error of the E-KM controller against that of the KM controller.

Furthermore, to statistically investigate how SOGP temporally improves the performance of the controller, we apply

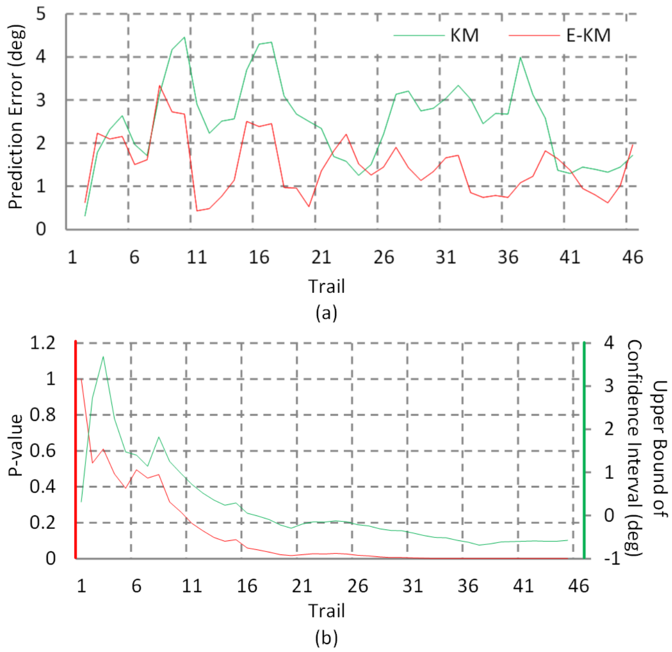

Fig. 5: The results of the online manipulation investigation experiment.

the t-test to the prediction errors of E-KM and KM

$$p(k) = p_{ttest}([e_{ekm}(1), ...e_{ekm}(k)], [e_{gm}(1), ...e_{gm}(k)]),$$
$$b(k) = b_{ttest}([e_{ekm}(1), ...e_{ekm}(k)], [e_{gm}(1), ...e_{gm}(k)]),$$

where $p(k)$ and $b(k)$ are the p-value and the upper boundary of the confidence interval gained by applying t-test on the data from 1 to k (k is smaller then the length of the data). We take the *Last Index of Negative Result* as a suggestion of the training time needed by the SOGP and the *Last Upper Boundary of the Confidence Interval* as a suggestion of the accuracy improvement of the E-KM over the KM.

V. Results and Discussions

Fig. 4 shows an illustration of the iCub manipulating a toy applier to its maximum workspace.

A. Online Manipulation Investigation Experiment

The prediction errors of the E-KM and the KM are shown in Fig. 5a, which indicates the learning process of the E-KM. Three immediate observations can be made from this graph: 1) The general trend of E-KM is lower than that of KM; 2) E-KM takes less than 10 trials to gain significant performance; 3) Although the E-KM model accepted failed training inputs (such as dislocations of the object at trials 20 and 44) along the sequence, it regained performance within very few trials afterwards.

We further applied t-test described in Section IV-D on the prediction error data, as shown in Fig. 5b. At 5% significance level, the test result shows the *Last Index of Negative Result* is 16 (out of 45), which indicates the prediction error of the E-KM is consistently smaller than the KM after 16 times of learning; and the *Last Upper Boundary of the Confidence Interval* is -0.57 *deg*, which indicates that the prediction error of the E-KM is 0.57 *deg* smaller than the KM at the confidence interval of 95%.

TABLE II: The mean static errors of the E-KM and KM controllers.

	W-box	Box	Bottle	Applier	Can
KM(*deg*)	1.9	2.8	1.3	1.0	1.1
E-KM(*deg*)	0.6	0.8	0.5	0.7	0.6
E-KM:KM (%)	31.5	28.5	38.5	70.0	54.5

B. Generalisability Investigation Experiment

The static error of both E-KM and KM controllers manipulating the different objects are shown in Fig. 6. We can see that the E-KM controller outperformed the KM in nearly all trials with different objects. The t-test indicates that the static error of the E-KM controller is significantly smaller than that of the KM controller with significance level of $\alpha = 0.05$ (P-value $p = 3.8 \times 10^{-4}$). The only exception is the "Can" in Trial 2. However, both KM and E-KM errors in this trial are less than 0.5 *deg*, while the control error of iCub position controller is ± 1 *deg*. Thus, this falls into the category of unavoidable error.

The mean static errors of the controllers are calculated and shown in TABLE II. All the mean static errors of the controller using the E-KM are less than the KM controller for all objects.

The above statistics and observations suggest that the E-KM is capable of generalising to a variety of object shapes, friction coefficients, weight distributions and sizes while maintaining the performance against the traditional KM controller.

VI. Conclusions And Future Work

In this paper, we contribute the Enhanced Kinematic Model of underactuated manipulation (E-KM), an integrated model of the Kinematic Model (KM) and the Sparse Online Gaussian Process (SOGP). The E-KM can compensate the shortfalls of the KM by on-the-fly training the SOGP on the residual between the KM and the ground truth data. Our experimental results show that the E-KM produces more accurate predictions over the KM and improves the controller performance. Another contribution is an underactuated manipulation controller which makes use of the dynamic image-based look-and-move strategy. Our experimental results show that the controller is capable to robustly manipulate the objects in real-life environments.

Looking forward, we envision several improvements that would increase the utility of our method. In this work, the controller only predicts the state one step ahead, which does not optimise the overall performance due to the static error, the settling time and etc. We will investigate on an improved controller which takes global optimisation into account. A possible approach for achieving this goal could be replacing the limiting function in Fig.1 with a reinforcement learning method to translate target poses to overall goals. Further

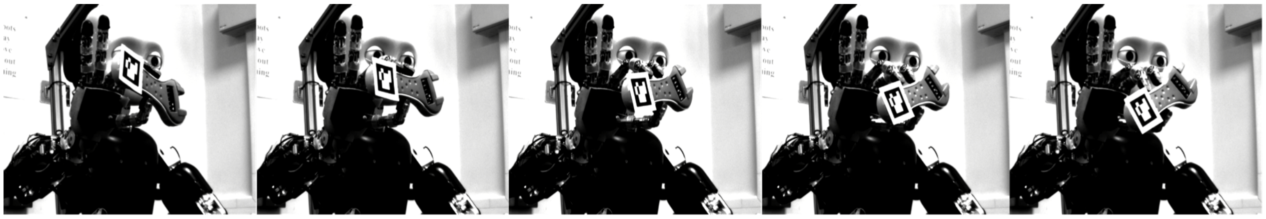

Fig. 4: Snapshots of the iCub rotating the toy applier from 0 to 28 *deg*.

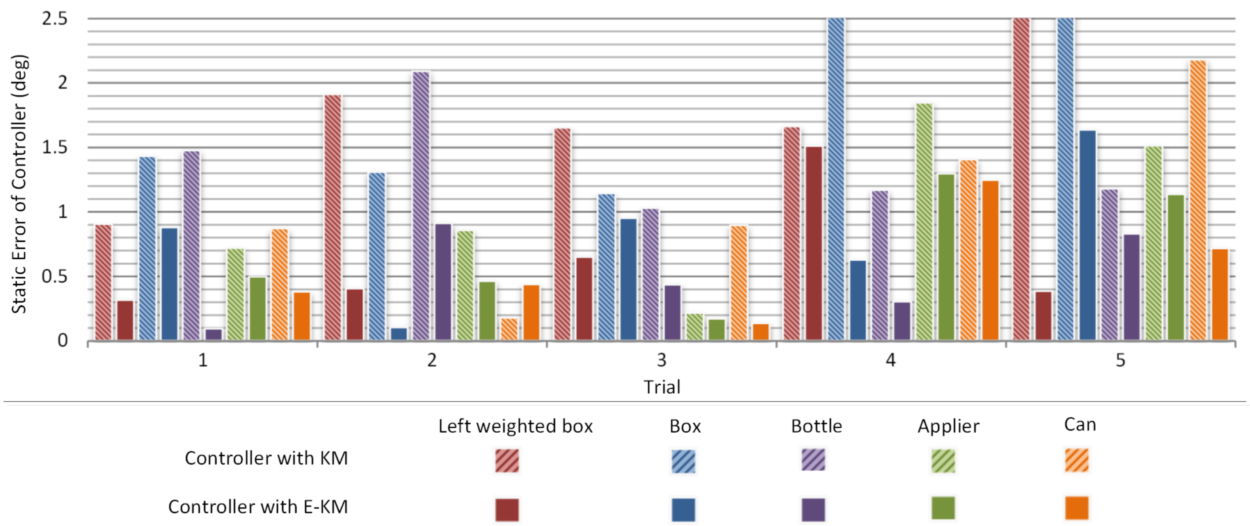

Fig. 6: The static error of the KM and E-KM controllers.

experiments will be carried out to test the prediction validity of the E-KM when visual input is switched off after the error prediction stablises.

ACKNOWLEDGMENT

The research leading to these results has received funding from the European Union Seventh Framework Programme FP7/2007-2013 Challenge 2 Cognitive Systems, Interaction, Robotics under grant agreement No [270490]- [EFAA].

REFERENCES

[1] C. Kemp, A. Edsinger, E. Torres-Jara, and D. Robots, "Challenges for robot manipulation in human environments," *IEEE Robotics & Automation Magazine*, vol. 14, pp. 20–29, Mar. 2007.

[2] A. M. Dollar, R. D. Howe, and S. Member, "Simple, Robust Autonomous Grasping in Unstructured Environments," in *International Conference on Robotics and Automation*, no. April, pp. 4693–4700, Apr. 2007.

[3] D. Prattichizzo and J. C. Trinkle, "Grasping," in *Springer Handbook of Robotics*, pp. 671–700, Spinrger, 2008.

[4] L. U. Odhner and A. M. Dollar, "Dexterous manipulation with underactuated elastic hands," in *International Conference on Robotics and Automation*, pp. 5254–5260, May 2011.

[5] M. Ciocarlie and P. Allen, "A design and analysis tool for underactuated compliant hands," in *International Conference on Intelligent Robots and Systems*, pp. 5234–5239, Oct. 2009.

[6] D. Caligiore, T. Ferrauto, D. Parisi, N. Accornero, M. Capozza, and G. Baldassarre, "Using motor babbling and Hebb rules for modeling the development of reaching with obstacles and grasping," in *International Conference on Cognitive Systems*, pp. E1–8, 2008.

[7] A. Dearden and Y. Demiris, "Learning forward models for robots," in *International Joint Conference on Artificial Intelligence*, vol. 5, pp. 1440–1446, 2005.

[8] J. Z. Kolter, *Learning and control with inaccurate models*. PhD thesis, Stanford University, 2010.

[9] P. Hemakumara and S. Sukkarieh, "Non-parametric UAV system identification with dependent Gaussian processes," 2011.

[10] J. Ko, D. J. Klein, D. Fox, and D. Haehnel, "Gaussian Processes and Reinforcement Learning for Identification and Control of an Autonomous Blimp," *International Conference on Robotics and Automation*, pp. 742–747, Apr. 2007.

[11] L. U. Odhner, R. R. Ma, and A. M. Dollar, "Precision grasping and manipulation of small objects from flat surfaces using underactuated fingers," in *International Conference on Robotics and Automation*, pp. 2830–2835, May 2012.

[12] A. Miller and H. Christensen, "Implementation of multi-rigid-body dynamics within a robotic grasping simulator," in *International Conference on Robotcis and Automation*, 2003.

[13] N. Vahrenkamp, M. Kröhnert, and S. Ulbrich, "Simox: A Robotics Toolbox for Simulation, Motion and Grasp Planning," *Robotics and Automation Systems*, pp. 585–594, 2013.

[14] L. Csató and M. Opper, "Sparse on-line gaussian processes.," *Neural computation*, vol. 14, pp. 641–68, Mar. 2002.

[15] H. Soh and Y. Demiris, "Iterative temporal learning and prediction with the sparse online echo state gaussian process," *International Joint Conference on Neural Networks*, pp. 1–8, June 2012.

[16] N. Vahrenkamp, T. Asfour, and R. Dillmann, "Efficient motion planning for humanoid robots using lazy collision checking and enlarged robot models," in *International Conference on Intelligent Robots and Systems*, pp. 3062–3067, Oct. 2007.

[17] X. Zhang, S. Fronz, and N. Navab, "Visual marker detection and decoding in AR systems: A comparative study," in *International Symposium on Mixed and Augmented Reality*, pp. 97–106, 2002.

[18] Y. Su, Y. Wu, K. Lee, Z. Du, Y. Demiris, and Y. Su, Yanyu; Wu, Yan; Lee, Kyuhwa; Du, Zhijiang;Demiris, "Robust Grasping for an Underactuated Anthropomorphic Hand under Object Position Uncertainty," in *International Conference on Humanoid Robots*, pp. 719–725, 2012.