In: Proc. 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems, San Diego, CA, USA (IROS 2007).

Load estimation and control using learned dynamics models

Georgios Petkos and Sethu Vijayakumar

***Abstract*— Classic adaptive control methods for handling varying loads rely on an analytically derived model of the robot's dynamics. However, in many situations, it is not feasible or easy to obtain an accurate analytic model of the robot's dynamics. An alternative to analytically deriving the dynamics is learning the dynamics from movement data. This paper describes a load estimation technique that uses the learned instead of analytically derived dynamics. We study examples where the various inertial parameters of the load are estimated from the learned models, their effectiveness in control is evaluated along with their robustness in light of imperfect, intermediate dynamic models.**

I. INTRODUCTION

Adaptive control for the problem of manipulation of unknown loads is a mature area that has given rise to theoretically well established and effective techniques [10] [11] [1]. Existing methods rely on an analytically derived model of dynamics. In many cases though, exact analytical derivation of the robot dynamics is not feasible. For example, there may be hard to model joint elasticity and friction or there may be uncertainties in the physical parameters of the robot. When an accurate dynamics model is difficult to derive, an attractive alternative is to learn the dynamics model using movement data [2]. Unfortunately though, a learned dynamic model cannot be directly used for load identification as it does not have an appropriate form for load estimation.

In this paper, we show how *a set of learned models* can be used to estimate the dynamics model in the form required for load identification. First, we show how this can be achieved from a set of learned models corresponding to manipulation of loads with *known* inertial parameters. We also show that the load estimates and the inferred model can be used for control. We, then, extend this to the case that learned models do not come labelled with inertial parameters. In that case, the estimated parameters of the load are a linear transformation of the actual ones but can be used for control. Furthermore, the issue of identifiability of parameters is addressed. The contribution of this work is that it shows how a robot can *generalize* the knowledge obtained from *learning* to manipulate a set of loads to other loads.

The rest of the paper is organised as follows. In Section II, we review the main methods in load estimation. In Section III we discuss learning the dynamics of a robot that manipulates a stationary load and using the learned dynamics for control. In Section IV, we see how from a set of learned models with known inertial parameters we can obtain the dynamics model in the form that is required for load estimation and show that accurate load estimation and control can be achieved using this method. In Section V, we extend this to the case that learned dynamics are not labelled with inertial parameters of the different loads.

Both authors are with the School of Informatics, University of Edinburgh, g.petkos@sms.ed.ac.uk

II. LOAD ESTIMATION

There are two main classes of load estimation methods. One requires torque sensing at all joints of the robot [7] [11], while the other requires force and torque sensing only at the wrist of the manipulator [1] [8]. Both classes of methods, are based on a linear relationship of the inertial parameters of the load and links to the dynamics of the robot.

We will first discuss the case where torque sensing in all joints is used. This approach treats the problem of load identification as a special case of link parameter estimation. The fundamental relationship that is exploited is [9]:

$$\tau = Y(q, \dot{q}, \ddot{q})\pi, \quad (1)$$

where τ is the vector of torques applied at the joints, q, $\dot{q}$ and $\ddot{q}$ are the joint angles,velocities and accelerations respectively and π is the vector of inertial parameters of all the links of the robot:

$$\pi = [m_1, m_1 l_{1x}, m_1 l_{1y}, m_1 l_{1z}, I_{1xx} ... m_n, m_n l_{nx} ... I_{nzz}]^T \quad (2)$$

Here, m_i is the mass of link i, l_i is the position vector of the center of mass of the link in the attached reference frame and I_i is the inertia tensor of the link measured around the origin of the link's reference frame. Given that the inertia tensor is a symmetric 3×3 matrix, this sums up to a total of 10 inertial parameters per link. Thus, with n links, π is a $10n$-dimensional vector. Y is called the regressor matrix and is in general a complicated function of kinematic parameters such as link lengths and joint orientations. Its dimensionality is $n \times 10n$. It is presumed that it can be accurately computed analytically. The full dynamic parameter identification problem entails inferring π. The scenario for load identification is a bit simpler though. If the load is assumed to be rigidly attached to the last link of the robot, manipulating different loads affects the dynamics only in changing the inertial parameters of the last link of the arm: m_n, l_n and I_n. Load estimation is then estimating the inertial parameters of the compound last link / load. Assuming that the inertial parameters of the first $n-1$ links are known and only the inertial parameters of the link n are unknown, the model of eq. (1) becomes:

$$\tau = c(q, \dot{q}, \ddot{q}) + Y_n(q, \dot{q}, \ddot{q})\pi_n \quad (3)$$

where the constant and known inertial parameters of the first $n-1$ links have been premultiplied by the respective first

$(n-1)\times 10$ columns of Y to give the vector c. Y_n consists of the last 10 columns of the matrix Y and π_n is the vector consisting of the inertial parameters of the compound last link / load only. Alternatively, this can be written as:

$$\tau = \tilde{Y}(q,\dot{q},\ddot{q})\tilde{\pi}_n \tag{4}$$

Here, $\tilde{Y}$ is the matrix $[c\, Y_n]$ and $\tilde{\pi}_n$ is the vector $[1\ \pi_n^T]^T$.

A sequence of measured torques $\tau^1, \tau^2 ... \tau^T$, joint angles $q^1, q^2 ... q^t$, velocities $\dot{q}^1, \dot{q}^2 ... \dot{q}^t$ and accelerations $\ddot{q}^1, \ddot{q}^2 ... \ddot{q}^t$ is collected and a sequence of $c^1, c^2 ... c^t$ and $Y_n^1, Y_n^2 ... Y_n^t$ is analytically computed. From these, the following equation is formed:

$$\begin{bmatrix} \tau^1 \\ \tau^2 \\ \vdots \\ \tau^t \end{bmatrix} = \begin{bmatrix} c^1 \\ c^2 \\ \vdots \\ c^t \end{bmatrix} + \begin{bmatrix} Y_n^1 \\ Y_n^2 \\ \vdots \\ Y_n^t \end{bmatrix} \pi_n \tag{5}$$

or more compactly:

$$\bar{\tau} = \bar{c} + \bar{Y}_n \pi_n \tag{6}$$

At this point any linear estimation technique can be used to estimate π_n. The simplest scenario is to assume stationarity of π_n and do simple least squares estimation of π_n:

$$\hat{\pi_n} = (\bar{Y}_n^T \bar{Y}_n)^{-1} \bar{Y}_n^T (\bar{\tau} - \bar{c}) \tag{7}$$

Another option is to assume non-stationarity of the load, and do recursive least squares estimation with a forgetting factor.

The other approach to load estimation uses a similar relationship to eq. (1) which relates the torque and force applied by the load to the last link of the robot. This has a linear relationship to π_{n+1}, where $n+1$ denotes the load only. A force / torque sensor at the wrist of the robot (between the last link and load) provides a set of measurements and a set of new regressor matrices (we improperly use the term regressor for this case as well due to lack of terminology) is computed and a least squares problem is solved to estimate π_{n+1}. In our discussion, we will focus on the first class of methods, because the model required for load estimation can be used for control as well. However, our arguments will be valid for the wrist force / torque sensor scenario as well and the relevance will be discussed.

A. Identifiability

There is an important issue in the described procedure that wasn't discussed. In general not all inertial parameters can be identified. There are two reasons for that. The first is that some parameters do not contribute at all in the dynamics. For example, consider a single link robot that revolves only around the y axis, then the moments of inertia around the x and z axes do not contribute at all in the dynamics and the corresponding columns of $\bar{Y}$ are 0. The second reason is that due to the structure of the manipulator, some columns of $\bar{Y}$ are linearly dependent. In general, the inertial parameters can be grouped in three categories:

- Identifiable parameters, that correspond to linearly independent columns of $\bar{Y}$.
- Partially identifiable parameters, that correspond to linearly dependent columns of $\bar{Y}$.
- Unidentifiable parameters, that do not contribute at all in the dynamics and the corresponding columns of $\bar{Y}$ are zero.

Existence of partially identifiable and unidentifiable parameters means that $\bar{Y}$ is not full rank and thus the least squares problem cannot be solved. A solution is to do ridge regression instead of least squares estimation. The accuracy of the estimates of identifiable parameters will depend on the selection of the penalty term λ, whereas non-identifiable parameters' estimates will be significantly wrong. A better solution is to remove the columns of $\bar{Y}$ corresponding to unidentifiable parameters and replace the columns corresponding to partially identifiable parameters by a proper linear combination. There has been some work on either symbolically [6] [5] or numerically [3] [4] characterizing the identifiability of inertial parameters. The most popular numerical approach involves doing Singular Value Decomposition on the $\bar{Y}$ matrix, for more details please see [4].

Classification of the inertial parameters of each link depends on the structure of the manipulator. However, some parameters may appear to be unidentifiable for specific movements of the manipulator although they aren't for others. For this purpose, a sufficiently rich movement has to be generated and there has been some work on finding rich movements.

In what follows, we will not focus on identifiability issues and generation of rich movements. We will assume that we know to which class each of the inertial parameters of the compound last link / load belongs. However, we will suggest some ways that the problem of identifiability could be dealt with using a set of learned models.

III. LEARNING DYNAMICS

The load estimation methods discussed presume that it is possible to compute the matrix that multiplies the inertial parameters vector analytically. However, this matrix may not be easy to derive, or there may be significant uncertainties in the parameters that it depends on.

An alternative to analytically deriving the dynamics model is to learn it. In this section we will discuss learning a dynamics model of the arm with a stationary load. In order to learn the dynamics model, a non-linear regression algorithm and movement data are needed. Movement data can be collected by controlling the arm with a non-model based controller, say a pure PID controller.

The form of the dynamics model that is learned is the inverse dynamics model, which maps joint angles, velocities and accelerations to torques that are required to achieve this acceleration from that set of joint angles and velocities:

$$\tau = g(q,\dot{q},\ddot{q}) \tag{8}$$

The inverse model can be used for control and as we will see can also be used for load estimation if we take the approach of torque sensing in all joints. If on the other hand, the wrist force / torque sensor approach is applied, then both an inverse model (for control) and a wrist sensor model (for load estimation) have to be learned.

A. Locally Weighted Projection Regression

An algorithm that has been shown to be robust in motor learning tasks is Locally Weighted Projection Regression (LWPR) [12] [13]. An LWPR model consists of a set of local linear models that come paired with a kernel that defines the area of validity of the local model. For a given input x, the kernel of the k-th local model determines a weighting $w_k(x)$ while the local linear model predicts an output $\psi_k(x)$. The combined prediction of LWPR is

$$\phi(x) = \frac{1}{W}\sum_k w_k(x)\psi_k(x), W = \sum_k w_k(x) \qquad (9)$$

Each locality kernel $w_k(x)$ has a parametric Gaussian form and the shape of the kernel is adapted during learning in a data driven manner. The local models are trained using an online variant of Partial Least Squares using a set of collected sufficient statistics. LWPR is incremental and non-parametric in the sense that new local models are added online on an as-needed basis when training proceeds and new areas of the input domain are explored. Furthermore, LWPR provides statistically sound confidence bounds. For more details on LWPR please see [13].

B. Using a dynamics model for control

An inverse model can be used in many control settings. However, we will use it as part of a composite controller. Given a desired trajectory in joint angle space $q^*_1, q^*_2 ... q^*_T$, the corresponding desired velocities $\dot{q}^*_1, \dot{q}^*_2 ... \dot{q}^*_T$, accelerations $\ddot{q}^*_1, \ddot{q}^*_2 ... \ddot{q}^*_T$ and the current joint angle at time t, the composite control command at time t is:

$$\tau_t = g(q^*_t, \dot{q}^*_t, \ddot{q}^*_t) + P(q^*_t - q_t) + D(\dot{q}^*_t - \dot{q}_t) \qquad (10)$$

This consists of a feedforward command given by the inverse model and a feedback command provided by a Proportional Derivative (PD) controller. A schematic of the composite controller can be seen in Fig. 1. If an accurate inverse model is used, the composite controller can achieve compliant, fast and accurate movement. One effect of the composite control approach is that the more accurate the inverse model g, the smaller are the errors and the error-correcting PD signals. Thus, the total amount of feedback control is a measure of the accuracy of the inverse predictive model.

C. Learning and control experiments

The ability of LWPR to learn non-linear dynamics that can be used for control was verified. A simulated [1] 3 DoF arm was used (see Fig. 2). The first joint allows up and down movements and the next two allow left and right movements. The task of the arm was to follow a smooth trajectory planned in joint angle space. The trajectory was a superposition of different phase-shifted sinusoidal trajectories for each joint. Twenty iterations of the trajectory were

[1] Simulations performed using ODE and OpenGL

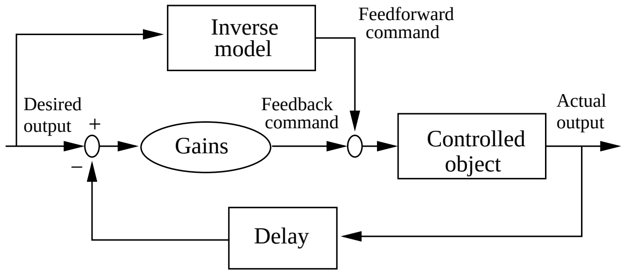

Fig. 1. The composite controller

executed. The arm was controlled by a PD controller in the first 3 iterations of the trajectory and then switched to a composite controller using the model being learned. Half of the observed data was used for training the model and half for testing and estimating the accuracy of the learned model. As discussed, the accuracy of the inverse model can also be judged by the amount of error-correcting feedback generated by the feedback part of the controller. If the inverse model is accurate then the feedback component of the composite controller will be small. Learning was repeated for the same task six times, while manipulating six different loads. The loads were selected in such a way that the learned dynamics can be used in our later experiments (see Section IV).

Results, averaged over the six trials can be seen in Fig. 3.

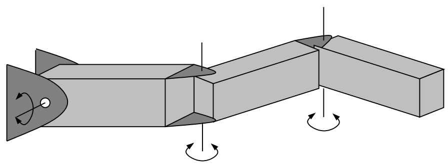

Fig. 2. Simulated 3DOF arm

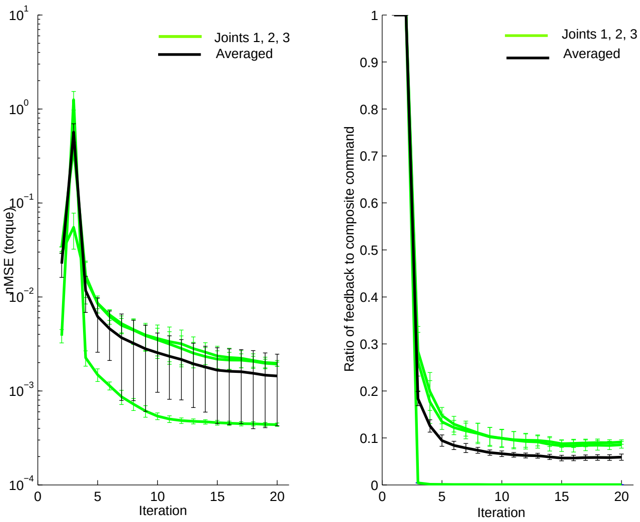

Fig. 3. Learning performance of LWPR over 6 different loads and 20 iterations. Left: normalized MSE on the test data. Right: average contribution of the error-correcting feedback PD control.

On the left we can see that the normalized MSE (nMSE) drops as the iterations proceed, reaching a very low value after only a few learning iterations. On the right we can see the ratio of the feedback to the composite command as training proceeds (we switch to composite control at the third iteration). A very low value is obtained, indicating that the learned models are quite accurate. Error bars are obtained by averaging between the six trials.

IV. LOAD ESTIMATION WITH LEARNED DYNAMICS MODELS

The models learned with any learning algorithm cannot be directly used for a load estimation task, since they do not provide the regressor matrix that is needed (we use the term regressor, although strictly it should only be used for the original Y matrix). An option is to use a set of learned dynamics, g^1, g^2 ... g^m corresponding to models with *reference* loads with known inertial parameters $\tilde{\pi}_n^1$, $\tilde{\pi}_n^2$ ... $\tilde{\pi}_n^m$. Since the regressor matrix $\tilde{Y}$ is the shared between different models, we can use eq. (4) to obtain:

$$\left[\begin{array}{c} g^1 g^2 ... g^m \end{array}\right] = \tilde{Y} \left[\begin{array}{c} \tilde{\pi}_n^1 \tilde{\pi}_n^2 ... \tilde{\pi}_n^m \end{array}\right] \quad (11)$$

or more compactly:

$$G = \tilde{Y}\tilde{\Pi} \quad (12)$$

Then, we can again solve a least squares problem to obtain an estimate of $\tilde{Y}$ like:

$$\tilde{Y} = G\tilde{\Pi}^T(\tilde{\Pi}\tilde{\Pi}^T)^{-1} \quad (13)$$

In an ideal scenario, with perfectly learned inverse models, we can use 11 models corresponding to manipulation of objects with linearly independent inertial parameters to obtain a good estimate of $\tilde{Y}$. More realistically though, since it is unlikely that the learned models are perfect, more accurate results can be expected while using more *reference* models. Having estimates of $\tilde{Y}$, we can do parameter estimation for novel loads using the method that we prefer (e.g. least squares). Subsequently, we can use the estimated $\tilde{Y}$ and the inferred estimates of the inertial parameter for control purposes. Substituting the learned model g with eq. 3, and using the estimated $\tilde{\pi}_n$, instead of the actual π_n, the control law of eq. (10) becomes:

$$\tau_t = c(q_t^*, \dot{q}_t^*, \ddot{q}_t^*) + Y_n(q_t^*, \dot{q}_t^*, \ddot{q}_t^*)\tilde{\pi}_n + P(q_t^* - q_t) + D(\dot{q}_t^* - \dot{q}_t) \quad (14)$$

Furthermore, $\tilde{Y}$ could be used with a numerical procedure like SVD for classifying the identifiability of the inertial parameters of the last link / load.

The above discussion considered the use of torque sensing at all joints. If the wrist force / torque sensor method is used, then one could still learn a model that maps from joint angles, velocities and accelerations to forces and torques measured at the wrist. Given a sufficient number of learned models, the regressor matrix can again be estimated and be used for load estimation. However, for control purposes, the inverse model of the robot will also have to be learned and another regressor matrix will have to be estimated.

A. Experiments

The six models learned in the previous section were used to obtain the regressor matrix and for performing load estimation and control under changing loads. Varying loads were randomly chosen with the constraint that out of the ten inertial parameters of the last link / load, only five were not zero. This was achieved by constraining both the center of mass of the link and the load to lie on the y axis of the link's reference frame (see Fig. 4) such that $m_n l_{nx}$ and $m_n l_{nz}$ are zero. Furthermore, the off-diagonal elements of the inertia tensor are zero. Out of the five non-zero inertial parameters, three were identifiable and inferred: m_n (mass), $m_n l_{ny}$ (product of mass and the y-position of center of mass) and I_{nxx} (moment of inertia around the x axis).

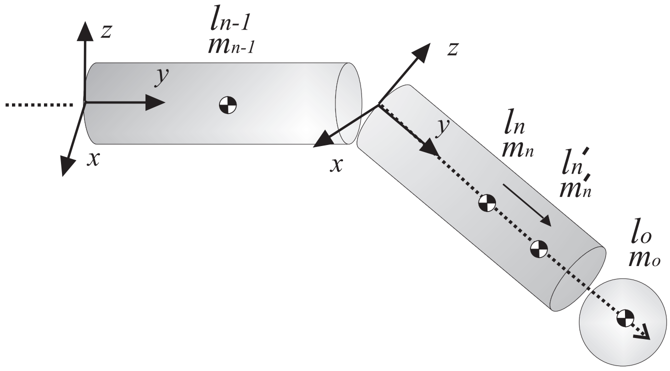

Fig. 4. In our experiments, both the center of mass of the last link l_n and the load l_o are constrained to lie on the y axis of the last link's reference frame, so that the center of mass of their union $\acute{l}_n$ also lies on the y axis.

We ran the simulation for 50 different trials (with random loads) and tried to estimate the inertial parameters of the compound last link / load. At the same time, the estimates were used for applying control as in eq.(14). The nMSE of the load parameter estimates, the average ratio of feedback to composite command and the tracking error averaged over the 50 trials (incl. std.dev.), can be seen in Table I.

	m_n	$m_n l_{ny}$	I_{nxx}
nMSE	**0.0006**±0.0024	**0.0010**±0.0037	**0.3278**±1.0031
	Joint 1	Joint 2	Joint 3
Ratio	**0.0016**±0.0009	**0.101**±0.0469	**0.1095**±0.05
	Joint 1	Joint 2	Joint 3
Tr.err.	**0.000009**±0.000009	**0.00036**±0.00035	**0.000209**±0.00019

TABLE I

ACCURACY OF LOAD ESTIMATION, RATIO OF FEEDBACK TO COMPOSITE COMMAND AND TRACKING ERROR

The accuracy of the estimates of m_n and $m_n l_{ny}$ is very high, whereas I_{nxx} is less accurately estimated. The ratio of feedback to composite command as well as the tracking error is as low as in the case where a single learned model of the arm dynamics with a single load was used for control (cf. Fig. 3, verifying that the estimated parameters are accurate enough for control purposes.

The same task was repeated for the case of randomly changing, ***non-stationary loads***. This is a more difficult scenario to handle than the one with stationary loads, since there may be large errors during switches between loads, especially when using the estimated parameters for composite control. The bottom part of Fig. 5 shows a typical behavior for the non-stationary mass estimation. The top half plots the corresponding feedback and feedforward commands using the estimated parameters, nicely showing that, with the exception of the spikes at the transition, the feedback command constantly stays close to zero. Statistics for the accuracy of the estimates, contribution of feedback command and tracking error averaged over the multiple trials are given in Table II. The accuracy of the estimates is lower than in the case of stationary loads, since our estimates are inevitably inaccurate for some short period after the transition. However, they are still quite accurate as indicated by the low feedback to composite command ratios.

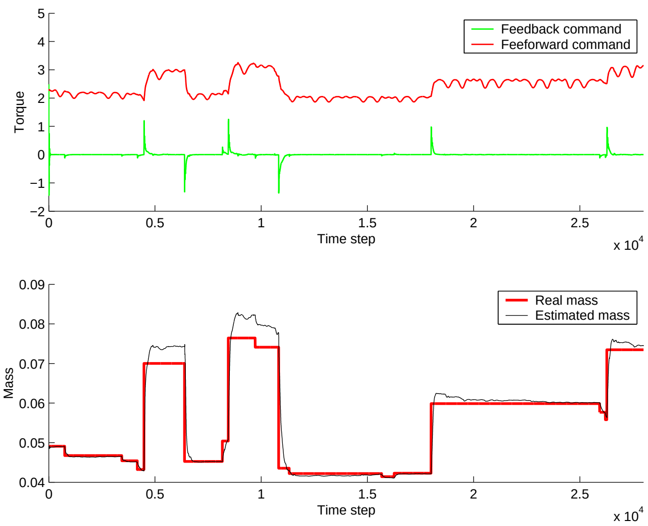

Fig. 5. Load estimation and control of non-stationary loads. Top: contribution of feedforward and feedback command. Bottom: real and estimated mass.

	m_n	$m_n l_{ny}$	I_{nxx}
nMSE	**0.0701**±0.3871	**0.1568**±0.3980	**1.1963**±2.4451
	Joint 1	Joint 2	Joint 3
Ratio	**0.0066**±0.0065	**0.1343**±0.0693	**0.1643**±0.0824
	Joint 1	Joint 2	Joint 3
Tr.err.	**0.000033**±0.000034	**0.00049**±0.00044	**0.00033**±0.00036

TABLE II
LOAD ESTIMATION AND CONTROL IN A NON-STATIONARY LOAD TASK.

Furthermore, the effect of using poorly learned dynamic models was investigated. The same load estimation and control task with 50 random stationary loads was executed using *reference* learned dynamics models with nMSE of around 0.05 as well as around 0.2, significantly higher than the average nMSE of 0.001 used in the previous experiments. This effect was produced by training the models with less data.

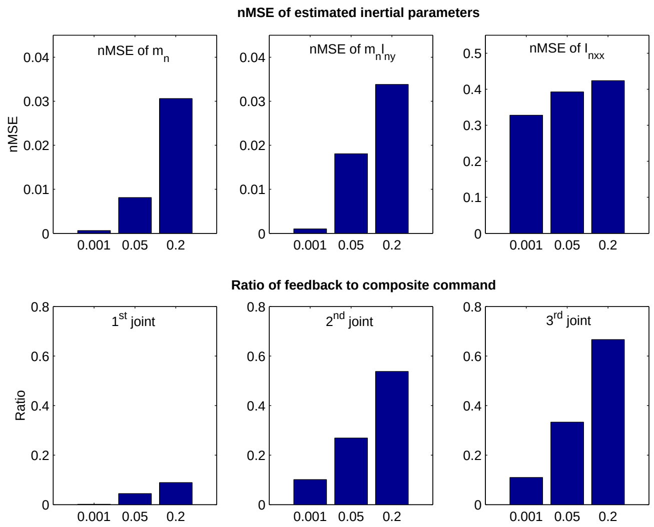

Fig. 6. Effect of imperfect dynamic models

Fig. 6 shows the results of the experiment and as expected, load estimation is less accurate as the accuracy of the learned models decreases. Also, the ratio of feedback to composite command becomes higher. However, there is graceful degradation of the performance, providing hope that such estimation techniques maybe be used to bootstrap learning, control and parameter identification in an online, incremental setting.

V. NOT USING LABELED DYNAMICS

Previous evaluations and experiments relied on the fact that adequate number of reference learned dynamic models with *known* inertial parameters exists. In many cases, an accurate estimate of the parameters, even for reference loads may not be forthcoming. Can we avoid using the real inertial parameters altogether? Consider introducing a linear transformation A and it's inverse between the regressor matrix and the set of inertial parameters in eq.(4) like:

$$\tau = \tilde{Y}(q, \dot{q}, \ddot{q}) A^{-1} A \tilde{\pi}_n \tag{15}$$

Then eq. (12) becomes:

$$G = \tilde{Y} A^{-1} A \tilde{\Pi} \tag{16}$$

and we can group the terms as:

$$G = (\tilde{Y} A^{-1})(A \tilde{\Pi}) \tag{17}$$

In the case that we infer l inertial parameters, if we have $m = l + 1$ learned dynamic models and the vectors of the inertial parameters for the different loads of the learned models are linearly independent, then one can set $A\tilde{\Pi}$ arbitrarily, to any full rank matrix. Hence, instead of estimating $\tilde{Y}$, one can estimate $\tilde{Y} A^{-1}$ as:

$$\tilde{Y} A^{-1} = G \tilde{\Pi}^T A^T (A \tilde{\Pi} \tilde{\Pi}^T A^T)^{-1} \tag{18}$$

This estimate can be used for estimating the current $A\pi_n$ which can be used with $\tilde{Y} A^{-1}$ for control. In this setup, the estimated load is not an estimate of the actual load but an estimate of a linear transformation of the load. However, when multiplied by the automatically appropriately transformed regressor matrix, it gives the right model of dynamics for the current load. In the case that there are more models m than inertial parameters, we cannot initialize the $A\tilde{\Pi}$ matrix arbitrarily since there may be no exact mapping A from the actual parameters $\tilde{\Pi}$ to the arbitrarily set matrix. Improper setting of $A\tilde{\Pi}$ would destroy the least squares solution. A solution would be to keep l base models and infer the inertial parameters for the rest $m - l$ inertial parameters and put them together with the arbitrarily set n parameters in the matrix $A\Pi$. However, since these estimates will not be accurate and since they will in turn be used to estimate another quantity, $\tilde{Y}A^{-1}$, it does not make much sense to use more that $l + 1$ models.

A. Experiments

The described procedure was empirically evaluated with the same task as in the case that we knew the inertial parameters of the load / last link of the arm. However, the accuracy of our estimates cannot be judged by comparing them directly with the correct values. Instead, we used the estimates for control and accumulated statistics on the contribution of the feedback command. This gives a measure about how accurate the feedforward command is and consequently how accurate the load estimates are. Again, we tried to infer three inertial parameters. The matrix $A\tilde{\Pi}_n$ was initialized as an upper triangular matrix, with all elements above the diagonal set to 1. The ratio of feedback to composite command and tracking error averaged over the 50 trials for the three different joints can be seen in Table III. The performance is similar to the case that labeled models are used, as in Table I.

	Joint 1	Joint 2	Joint 3
Ratio	**0.0016**$\pm$0.0014	**0.0971**$\pm$0.0342	**0.1216**$\pm$0.0505
	Joint 1	Joint 2	Joint 3
Tr.err.	**0.00001**$\pm$0.00001	**0.00032**$\pm$0.00026	**0.00022**$\pm$0.00021

TABLE III

LOAD ESTIMATION AND CONTROL WITHOUT LABELED LEARNED MODELS.

VI. DISCUSSION

We have shown that it is possible to do load estimation without using an analytically derived model. Experiments illustrate that using a set of learned models labelled with the appropriate inertial parameters (only of the load or of the load and last link of the arm) it is possible to do accurate load estimation and use the derived model and load estimates for control purposes. We have also shown that if the inertial parameters are not available, a linear transformation of the actual inertial parameters can be estimated and that this estimate can be used for control purposes. This nicely achieves *generalizing* the knowledge obtained from *learning* to manipulate a set of loads to manipulating other loads, without using any prior knowledge about the dynamics or the loads. Most importantly, there seems to be a graceful degradation of the performance of the load estimates and the control performance in relation to the inaccuracies of the learned dynamic models.

An interesting extension would be to do automatic classification of the identifiability of the inertial parameters of the load or last link. Since a learned set of models with their respective inertial parameters can give an estimate of the actual regressor matrix, a sequence of estimates can be used with a numerical algorithm like SVD to achieve this. However, although in the case that analytically derived dynamics are used, the zero singular values can be directly incorporated using the methods described in [4], in the case that learned dynamics are used, the regressor matrix will almost always not be singular (due to approximation inaccuracies). In the case that the labels are not known and the matrix $\tilde{Y}A^{-1}$ is inferred, SVD cannot be used anymore to determine the number of identifiable inertial parameters since the unknown linear transformation A^{-1} changes the singular values of $\tilde{Y}$. A possibility in that case would be to do an incremental search, adding one model and one variable at a time until performance decreases.

VII. ACKNOWLEDGEMENTS

GP is supported by the Greek State Scholarships Foundation. The work is partially funded through an EU-FP6 Integrated Project (SENSOPAC) grant to SV.

REFERENCES

[1] C G Atkeson, C H An, and J M Hollerbach. Estimation of inertial parameters of manipulator loads and links. *Int. J. Rob. Res.*, 5(3):101–119, 1986.

[2] C. G. Atkeson, A. W. Moore, and S. Schaal. Locally weighted learning for control. (1-5):75–113, 1997.

[3] F. Caccavale G. Antonelli and P. Chiacchio. A systematic procedure for the identification of dynamic parameters of robot manipulators. *Robotics*, 17:427–435, 1999.

[4] Maxime Gautier. Numerical calculation of the base inertial parameters of robots. In *Proceedings of IEEE International Conference on Robotics and Automation*, volume 2, pages 1020–1025. IEEE, 1990.

[5] W. Huo. New formulas for complete determining base parameters of robots. In *Robotics and Automation, 1995. Proceedings., 1995 IEEE International Conference on*, volume 3, pages 3021–3026, 1995.

[6] Wisama Khalil and Fouad Bennis. Symbolic calculation of the base inertial parameters of closed-loop robots. *I. J. Robotic Res.*, 14(2):112–128, 1995.

[7] Howard Olsen and George Bekey. Identification of robot dynamics. *Proceedings of the IEEE International Conference on Robotics and Automation*, 1986.

[8] Krzysztof Kozlowski Piotr Dutkiewicz and Waldemar Wroblewski. Experimental identification of robot and load dynamic parameters. *Proceedings of the IEEE Conference on Control Applications*, 1993.

[9] L. Sciavicco and B. Siciliano. *Modelling and Control of Robot Manipulators*. Springer, 2000.

[10] J.E. Slotine and W. Li. On the adaptive control of robot manipulators. *International Journal of Robotics Research*, 6(49–59), 1987.

[11] J. Swevers, C. Ganseman, D. B. Tukel, J. de Schutter, and H. Van Brussel. Optimal robot excitation and identification. *Robotics and Automation, IEEE Transactions on*, 13(5):730–740, 1997.

[12] S. Vijayakumar and S. Schaal. Locally weighted projection regression: An o(n) algorithm for incremental real time learning in high dimensional spaces. *Proceedings of the Seventeenth International Conference on Machine Learning (ICML 2000)*, 1:288–293, 2000.

[13] Sethu Vijayakumar, Aaron D'Souza, and Stefan Schaal. Incremental online learning in high dimensions. *Neural Computation*, 17:2602–2634, 2005.