A Comparison of Classical and Learning Controllers ⋆

Joseph Sun de la Cruz * **Dana Kulić** * **William Owen** **

* *Department of Electrical and Computer Engineering*
University of Waterloo, Waterloo, ON, Canada
(e-mail: {jsundela, dkulic}@uwaterloo.ca
** *Department of Mechanical and Mechatronics Engineering*
University of Waterloo, Waterloo, ON, Canada
(e-mail: bowen@uwaterloo.ca)

Abstract: This paper focuses on evaluating Locally Weighted Projection Regression (LWPR) as an alternative control method to traditional model-based control schemes. LWPR is used to estimate the inverse dynamics function of a 6 degree of freedom (DOF) manipulator. The performance of the resulting controller is compared to that of the resolved acceleration and the adaptive computed torque (ACT) controller. Simulations are carried out in order to evaluate the position and orientation tracking performance of each controller while varying trajectory velocities, end effector loading and errors in the known parameters. Both the adaptive controller and LWPR controller have comparable performance in the presence of parametric uncertainty including friction. The ACT controller outperforms LWPR when the dynamic structure is accurately known and the trajectory is persistently exciting.

Keywords: Learning Control, Adaptive Control, Robot Dynamics

1. INTRODUCTION

The use of robotics worldwide is most prevalent in the industrial setting where the environment is highly controlled. Under these conditions, robot manipulation often consists of repetitive tasks such as pick-and-place motions, allowing the use of simple, computationally inexpensive decentralized Proportional-Integral-Derivative (PID) control which treat the nonlinearities and highly coupled nature of manipulators as disturbances. Unlike decentralized controllers, control strategies that are based on the dynamic model of the manipulator, known as model-based controllers, present numerous advantages such as increased performance during high-speed movements, reduced energy consumption, improved tracking accuracy and the possibility of compliance (Nguyen-Tuong et al., 2009). However, the performance of model-based control is highly dependent upon the accurate representation of the robot's dynamics, which includes precise knowledge of the inertial parameters of link mass, centre of mass and moments of inertia, and friction parameters (Craig et al., 1986). In practice, obtaining such a model is a challenging task which involves modeling physical processes that are not well understood or difficult to model, such as friction (Armstrong-Hélouvry et al., 1994) and backlash. Thus, assumptions concerning these effects are often made, leading to inaccuracies in the model. Furthermore, uncertainties in the physical parameters of a system may be introduced from discrepancies between the manufacturer data and the actual system (Ayusawa et al., 2008). Changes to operating conditions can also cause the structure of the system model to change, thus resulting in degraded performance.

Traditionally, adaptive control strategies have been used to estimate parameters of the dynamic model online (Craig et al., 1986), but with the requirements of knowing joint accelerations and good initial estimates of the system parameters. Since (Craig et al., 1986), subsequent work has been done to eliminate these constraints (Ortega and Spong, 1988; Burdet and Codourey, 1998; Yu and Lloyd, 1995). A number of adaptive laws have been proposed (Slotine and Li, 1987; Landau and Horowitz, 1988) based on the preservation of passivity properties of the robot. Although they differ from the class of controllers in (Craig et al., 1986), the motivation for these schemes is also to eliminate the need for joint acceleration measurement (Ortega and Spong, 1988). Despite these advancements, adaptive methods are still reliant upon adequate knowledge of the structure of the dynamic model and are thus susceptible to modeling errors and changes in the model structure. An alternative solution is sliding mode control (Slotine, 1985) which has been shown to be robust to system modeling errors, but is susceptible to control chattering due to its discontinuity across sliding surfaces (Yao and Tomizuka, 1996).

Whereas adaptive control strategies assume an underlying dynamic structure to the system, model learning controllers attempt to learn the dynamic model of the system. An early approach to this problem named MEMory (Burdet and Codourey, 1998) involves storing the dynamics of a system in memory for use as the feedforward term of subsequent runs, assuming that the system repeats the same trajectory. More recently, statistical regression approaches

⋆ This work was partially supported by the Natural Sciences and Engineering Research Council of Canada.

have been used to infer the optimal structure to describe the observed data, making it possible to encode nonlinearities whose structure may not be well-known. Solutions to this form of supervised learning can be broadly categorized into two types (Vijayakumar et al., 2005) - global methods such as Gaussian Process Regression (GPR) (Rasmussen and Williams, 2006) and Support Vector Regression (SVR) (Nguyen-Tuong et al., 2009), and local methods such as Locally Weighted Projection Regression (LWPR). Recent studies comparing these learning methods (Nguyen-Tuong et al., 2008) show that while SVR and GPR can potentially yield higher accuracy than LWPR, their computational cost is still prohibitive for online incremental learning.

Rather than learning the underlying model structure of the system, Iterative learning control (ILC) (Arimoto et al., 1984; Bristow et al., 2006) incorporates information from error signals in previous iterations to directly modify the control input for subsequent iterations. However, ILC is limited primarily to systems which track a specific repeating trajectory and are subject to repeating disturbances (Bristow et al., 2006), whereas the model learning approaches such as LWPR can be incrementally trained to deal with non-repeating trajectories.

(Burdet and Codourey, 1998) compared various non-parametric learning approaches, including Neural Networks and the MEMory algorithm to model based adaptive controllers. An experimental comparison of several adaptive laws is given in (Whitcomb et al., 1993). However, to the author's knowledge, since these papers, there has not been any work in comparing the newer generation of regression-based learning techniques to model based control strategies. Hence, this paper focuses on evaluating LWPR as an alternative control method to traditional model-based control schemes. The performance of the LWPR controller is compared to model-based controllers, i.e. Resolved Acceleration (Sciavicco and Scicliano, 2000), and Adaptive computed torque (Craig et al., 1986). A quantitative analysis and comparison of the performance of each controller is given by carrying out simulations involving various trajectories, accelerations and velocity profiles, as well as parameter uncertainty. By analyzing the performance of each controller under these conditions, this paper also aims to identify scenarios for which each controller is best suited.

2. OVERVIEW OF MODEL-BASED CONTROL

The dynamic equation of a manipulator characterizes the relationship between its motion (position, velocity and acceleration) and the joint torques (Sciavicco and Scicliano, 2000):

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q},\dot{\mathbf{q}}) + \mathbf{G}(\mathbf{q}) = \boldsymbol{\tau} \qquad (1)$$

$\mathbf{q}$ is the nx1 vector of joint angles for an n-degree of freedom (DOF) manipulator, $\mathbf{M}(\mathbf{q})$ is the nxn inertia matrix, $\mathbf{C}(\mathbf{q},\dot{\mathbf{q}})$ is the nx1 centripetal and Coriolis force vector, $\mathbf{G}(\mathbf{q})$ is the nx1 gravity loading vector and $\boldsymbol{\tau}$ is the nx1 torque vector. Equation (1) does not include additional torque components caused by friction, backlash, actuator dynamics and contact with the environment. If accounted for, these components are modeled as additional terms in (1).

Model-based controllers apply the joint space dynamic equation (1) to cancel the nonlinear and coupling effects of the manipulator. A common example of this is the computed torque approach (Sciavicco and Scicliano, 2000) in which the control signal $\mathbf{u}$ is composed of the computed torque signal, $\mathbf{u_{CT}}$, which is set to the torque determined directly from (1). This term globally linearizes and decouples the system, and thus a linear controller can be applied for the feedback term, $\mathbf{u_{FB}}$, which provides stability and disturbance rejection. Typically a PD scheme is used. Desirable performance of the computed torque approach is based on the assumption that values of the parameters in (1) match the actual parameters of the physical system. Otherwise, imperfect cancelation of the nonlinearities and coupling occurs. Hence, the resulting system is not fully linearized and decoupled and thus higher feedback gains are necessary to achieve good performance.

In practice, the dynamic parameters of a manipulator not known precisely enough to perfectly cancel the nonlinear and coupling terms (Craig et al., 1986). One solution is the adaptive control approach (Craig et al., 1986),(Ortega and Spong, 1988). In addition to an underlying control objective, an adaptive controller also incorporates a parameter update law which estimates unknown parameters based on the tracking error. In (Craig et al., 1986), an adaptive version of the computed torque control method is presented. In order to estimate the inertia parameters of the robot, the dynamic model (1) is reformulated as:

$$\boldsymbol{\tau} = \boldsymbol{\phi}(\mathbf{q},\dot{\mathbf{q}},\ddot{\mathbf{q}})\boldsymbol{\theta} \qquad (2)$$

where $\boldsymbol{\phi}$ is an nxr regressor matrix which depends on the kinematics of the robot, $\boldsymbol{\theta}$ is an rx1 vector of unknown parameters. This model is linear in the parameters, allowing a Lyapunov-based parameter update law to be implemented:

$$\dot{\hat{\boldsymbol{\theta}}} = \boldsymbol{\gamma}\boldsymbol{\phi}^{\boldsymbol{T}}\hat{\boldsymbol{M}}^{\boldsymbol{-1}}\boldsymbol{e} \qquad (3)$$

where $\hat{\boldsymbol{\theta}}$ is the estimate of the unknown inertia parameters, $\boldsymbol{\gamma}$ is an rxr gain matrix, $\hat{\mathbf{M}}$ is the estimated inertia matrix, $\boldsymbol{e}$ is the filtered servo error and r is the number of unknown parameters. For this paper, the inertia parameters (Khosla, 1989) as well as Coulomb and friction parameters, are treated as unknowns.

3. LEARNING INVERSE DYNAMICS

While the adaptive approach requires accurate knowledge of the structure of the dynamic model of the manipulator, the learning approach obtains a model using measured data (Nguyen-Tuong et al., 2009), allowing unknown or unmodeled nonlinearities such as friction and backlash to be accounted for. In order to be practical for manipulator control, learning algorithms must process continuous streams of training data to update the model and predict outputs fast enough for real-time control. Locally Weighted Projection Regression (LWPR) achieves these objectives using nonparametric statistics (Schaal et al., 2002).

The problem of learning the inverse dynamics relationship in the joint space can be described as the map from joint positions, velocities and accelerations to torques

$$(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}_{\mathbf{d}}) \mapsto \boldsymbol{\tau} \quad (4)$$

where $\boldsymbol{\tau}$ is the nx1 torque vector, and $\mathbf{q}$ is the nx1 vector of generalized coordinates.

LWPR approximates this mapping with a set of piecewise local linear models based on the training data that the algorithm receives. Formally stated, this approach assumes a standard regression model of the form

$$\mathbf{y} = \mathbf{f}(\mathbf{X}) + \varepsilon \quad (5)$$

where $\mathbf{X}$ is the input vector, $\mathbf{y}$ the output vector, and $\boldsymbol{\epsilon}$ a zero-mean random noise term. For a single output dimension of $\mathbf{y}$, given a data point $\mathbf{X_c}$, and a subset of data close to $\mathbf{X_c}$, with the appropriately chosen measure of closeness, a linear model can be fit to the subset of data:

$$y_{ik} = \boldsymbol{\beta_{ik}}^T \mathbf{X} + \varepsilon \quad (6)$$

where $y_i k$ denotes the k^{th} subset of data close to $\mathbf{X_c}$ corresponding to the i^{th} output dimension and $\boldsymbol{\beta_{ik}}$ is the set of parameters of the hyperplane that describe $y_i k$. The region of validity, termed the receptive field (Vijayakumar et al., 2005) is given by

$$w_{ik} = exp(-\frac{1}{2}(\mathbf{X} - \mathbf{X_{ck}})^{\mathbf{T}} \mathbf{D_{ik}} (\mathbf{X} - \mathbf{X_{ck}})) \quad (7)$$

where w_{ik} determines the weight of the k^{th} local linear model of the i^{th} output dimension (i.e. the ik^{th} local linear model), $\mathbf{X_{ck}}$ is the centre of the k^{th} linear model, $\mathbf{D_{ik}}$ corresponds to a positive semidefinite distance parameter which determines the size of the ik^{th} receptive field. Given a query point $\mathbf{X}$, LWPR calculates a predicted output

$$\hat{y_i}(X) = \sum_{k=1}^{K} w_{ik} \hat{y}_{ik} / \sum_{k=1}^{K} w_{ik} \quad (8)$$

where K is the number of linear models, $\hat{y}_{ik}$ is the prediction of the ik^{th} local linear model given by (6) which is weighed by w_{ik} associated with its receptive field. Thus, the prediction $\hat{y_i}(\mathbf{X})$ is the weighted sum of all the predictions of the local models, where the models having receptive fields centered closest to the query point are most significant to the prediction. This prediction is repeated i times for each dimension of the output vector $\mathbf{y}$.

Determining the set of parameters $\boldsymbol{\beta}$ of the hyperplane is done via regression, but can be a time consuming task in the presence of high-dimensional input data. To reduce computational effort, LWPR assumes that the data can be characterized by local low-dimensional distributions, and attempts to reduce the dimensionality of the input space $\mathbf{X}$ using Partial Least Squares regression (PLS). PLS fits linear models using a set of univariate regressions along selected projections in input space which are chosen according to the correlation between input and output data (Schaal et al., 2002).

4. SIMULATIONS

In order to evaluate the performance of the LWPR learning controller, two 'classical' controllers in the joint space were also implemented: the resolved acceleration (RA) controller (Sciavicco and Scicliano, 2000) and the adaptive computed torque (ACT) controller (Craig et al., 1986; Ortega and Spong, 1988) given by (3).The performance of these controllers was evaluated in simulation using Matlab, the Robotics Toolbox (RTB) (Corke, 1996) and the open source LWPR code (Vijayakumar et al., 2005). LWPR was used to learn the joint space dynamics of a standard 6 DOF Puma 560 with the kinematic and dynamic parameters obtained from the RTB. The control loop executed at 1ms while the model was updated every 5ms for both the LWPR and ACT algorithms.

In order to properly assess the performance of the model-based controllers, a trajectory which excites the dynamics of the system caused by inertia, gravity and Coriolis/centripetal effects in (1) must be tracked. The 'figure 8' trajectory (Nakanishi et al., 2008) is used, as it includes both straight and curved sections which induce significant torques from Coriolis/centripetal effects.

The first simulation involves the position tracking of the 'figure 8' trajectory in the horizontal XY plane, with a length and width of 0.2m. The position component of the end effector trajectory was designed in the task space and converted to a joint space trajectory using a numerical inverse kinematics algorithm (Corke, 1996). To evaluate orientation control, a sinusoidal signal was input as the desired angular velocity of the end effector. 'Figure 8' frequencies of approximately 0.25 and 0.3 Hz were used to test the tracking capabilities of the controllers under various velocities. The LWPR controller was trained for 60s on the 0.25Hz trajectory, after which training was stopped and tracking performance was evaluated. The system was then allowed to train for another 120s. Next the trajectory frequency was increased to 0.3Hz, and the system was trained for an additional 30s, after which tracking performance was evaluated. These results are then compared against the performance of the RA controller. The durations of training were determined by observing the mean squared error (MSE) of the predicted torques from the LWPR controller. Training was stopped when the observed MSE had asymptotically decreased to a low value. Since no parameter perturbation was introduced yet, only the RA and LWPR controllers are compared in the first simulation.

The second simulation involves tracking the 'figure 8' pattern with varied end effector loads, thereby introducing model parameter errors. The trained system from the first simulation (0.25Hz and 180s training) was used to track the 'figure 8' trajectory, but with additional end effector masses of 0.5 and 1kg. The same conditions were repeated on the ACT and RA controller. After 30s of training time for the LWPR controller and ACT controller, the performance of all three controllers was evaluated.

The third simulation adds varying amounts of error in the inertia parameters of the model while observing the resulting performance of the three controllers when tracking the 'figure 8' trajectory. Training and adaptation of these models was done in the same manner as above.

The fourth simulation introduces friction in addition to inertia parameter uncertainty while tracking a persistently exciting (PE) trajectory as described in (Craig et al., 1986), which is designed directly in the joint space as a linear combination of sinusoids. The resulting task space trajectory is a cardioid-like shape, as seen in figure 2. This trajectory is used for two reasons. Firstly, by using a trajectory with significant frequency content, the effect of persistence of excitation on the tracking and performance of ACT will be evaluated. Secondly, by shifting the operating range of the manipulator away from that of the figure 8 trajectory, the generalization performance of LWPR will be tested. Furthermore, both Coulomb and viscous friction are introduced into the simulation. In order to assess the ACT controller's ability to cope with unmodeled dynamics, two cases are tested: one in which both Coulomb and viscous friction are accounted for in equation (1), and one in which only Coulomb friction is modeled. Friction is modeled as:

$$\boldsymbol{\tau_f} = c sign(\dot{\mathbf{q}}) + v\dot{\mathbf{q}} \tag{9}$$

where $\boldsymbol{\tau_f}$ is the torque due to Coulomb and viscous friction, c is the Coulomb friction constant, and v is the viscous friction constant. The friction constants were obtained from the defaults for the Puma 560 in the RTB.

4.1 Parameter Tuning and Initialization

The stability of the ACT controller was found to be highly sensitive to the adaptive gain parameter, γ (3). While a higher value of γ generally results in faster adaptation time, it increases the system's sensitivity to noise and numerical errors from integration of the time derivative of the estimated parameters (3). An adaptive gain of 0.01 was found to be the best tradeoff.

Although LWPR incorporates many algorithms which enable the system to automatically adjust its parameters for optimal performance, initial values of these parameters can significantly impact the convergence rate. The initial value for the distance parameter $\mathbf{D}$ (7) dictates how large a receptive field is upon initialization. Too small a value of $\mathbf{D}$ (corresponding to large receptive fields) tends to delay convergence while a larger value of $\mathbf{D}$ results in overfitting of the data (Vijayakumar et al., 2005). This parameter was generally tuned through a trial-and-error process which involved monitoring the MSE of the predicted values during the training phase. The initial performance of the LWPR controller is also highly dependent upon the data sets that are used to train the LWPR model. Because LWPR is a local learning approach, it must be trained in the region(s) of input space that the manipulator will be operating in. In order to train the model, a low-gain PD controller was used to track the desired trajectory while the LWPR model obtained training data according to the mapping in (4). The initial value if the distance parameter, $\mathbf{D}$, was set to 0.05 for each input dimension.

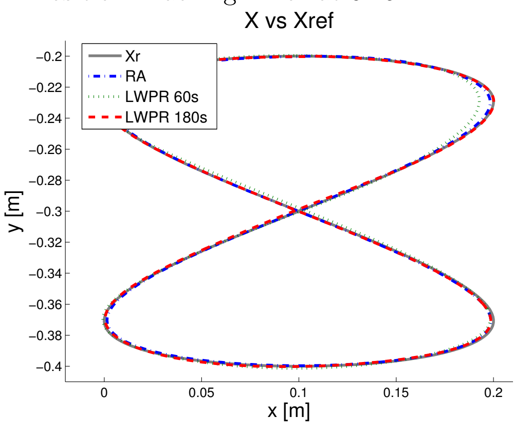
4.2 Results

The LWPR model was trained on the 'figure 8' trajectory at 0.25Hz, enabling it to predict the necessary torques for tracking at frequencies near 0.25Hz. As seen in Figure 1 and Table 1, after an additional training period of 40s, the LWPR controller compensated for the 0.3Hz trajectory, allowing it to perform nearly as well as the ideal RA controller. This illustrates the ability of the LWPR controller to rapidly adjust to changes in its operating conditions. However frequencies greater than 0.3Hz were sufficient to push the system far enough from the trained region of input space, eventually resulting in the prediction of zero for all the joint torques. A potential solution is a more complete initial training of the LWPR model. If the initial training set were to include a larger subset of the input space obtained through motor babbling (Peters and Schaal, 2008) for example, it is expected that the LWPR controller would be able to handle larger perturbations.

Table 1. Frequency - RMS tracking error (mm,deg)

Frequency	**0.25Hz**	**0.30Hz**
RA	1.80, 0.15	2.03, 0.22
LWPR, 60s	2.71, 1.31	/
LWPR, 180s	1.95, 0.25	2.20, 0.42

Fig. 1. Position Tracking Error at 0.25Hz



The second simulation evaluated the ability of the controllers to handle unmodeled end effector loads. As seen in Table 2, due to the unknown mass of the end effector, imperfect linearization and decoupling cause a decrease in tracking performance of the RA controller. Both the LWPR and ACT controller are able to outperform the RA controller after 30s of additional training. Although the ACT controller has a-priori knowledge of the structure of the dynamic equation of the manipulator, it does not perform any better than the LWPR controller in position tracking after the same length of adaptation time. This is due to the slow convergence of estimated masses to their actual values, which can in turn be explained by the relatively low adaptive gain and the lack of a PE trajectory. However, as seen in the orientation results, by tracking sinusoidal angular velocities on each joint of the wrist, the ACT controller yields much better orientation tracking as compared to the LWPR controller, illustrating the importance of PE trajectories in the performance of the ACT controller. The LWPR controller was able to learn the inverse dynamics for both the 0.5kg and 1kg payload, but not for masses greater than 1kg. Similar to the first simulation, a sufficiently large disturbance will push the system to operate in a region outside its training, thus yielding poor tracking performance.

Simulation three introduces parameter estimate errors into all the link masses, centre of mass locations and the moments of inertia of each link. Table 3 illustrates that the inaccurate knowledge of the dynamic parameters causes significant degradation in performance for the RA controller, due to the imperfect linearization of the system

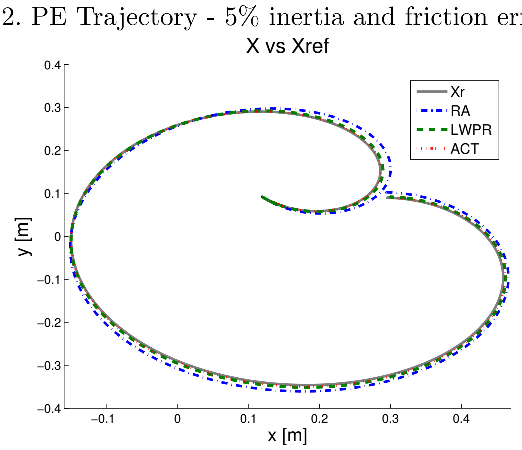
Table 2. Payload - RMS tracking error (mm,deg)

Payload	+0.5kg	+1kg
RA	4.95, 0.60	16.45, 1.15
ACT, 40s	4.44, 0.44	2.70, 1.26
LWPR, +40s	4.16, 0.65	7.01, 1.29

dynamics. As seen in Table 3, the performance of the ACT controller was particularly poor in comparison to the LWPR controller, and even the RA controller. This can be explained by the fact that the perturbation of the inertia parameters was applied to all the joints of the manipulator, unlike the case of end effector loading where only the parameters of one link was perturbed. Hence, it is expected that the adaptive controller would require both a persistently exciting trajectory, which excites all the dynamic modes of the structure, (Craig et al., 1986) and a longer adaptation time than 40s to yield better tracking results in this scenario. The importance of PE trajectories will be illustrated in the next simulation. For the LWPR controller, similar findings to that of simulations one and two were observed in that inertia parameter perturbations greater than 10% were sufficient to prevent LWPR from predicting accurate joint torques.

Simulation four introduces the PE trajectory and Coulomb and viscous friction in addition to inertia parameter error. As seen in Table 4, the RA controller performed the poorest due to the error in its model parameters. When the ACT is presented with the structure of the full friction model, the resulting tracking performance is significantly better than the RA controller.

Fig. 2. PE Trajectory - 5% inertia and friction error



However, when only partial knowledge of the model is present (in this case only viscous friction), the performance gain is no longer present. Unlike simulation three, the ACT controller now outperforms the LWPR controller, provided that the friction model is fully known. This further illustrates the importance of: 1) a persistently exciting trajectory and 2) accurate knowledge of structure of the dynamic model when using adaptive control. The PE trajectory was also chosen to be significantly different from the figure 8 in order to test the generalization of the LWPR model. The first test involved using the model that was learned in simulation three to attempt to track the PE trajectory. This model had seen roughly 10,000 training points, all of which were localized to the figure

Table 3. Parameter Error - RMS tracking error (mm,deg)

Parameter Error	+1%	+5%
RA	2.15, 0.25	2.65, 0.30
ACT, 40s	2.20, 0.12	2.75, 0.15
LWPR, +40s	1.55, 0.75	1.66, 0.86

Table 4. Parameter Error and Friction - RMS tracking error (mm,deg)

Parameter error	+1%	+5%
RA	2.45, 0.55	3.10, 0.80
ACT, partial friction, 40s	2.0, 0.40	2.1, 0.45
ACT, full friction, 40s	1.65, 0.30	1.70, 0.32
LWPR, 240s	1.75, 0.45	1.80, 0.50
LWPR, motor babbling, 120s	1.82, 0.50	1.85, 0.55

8 trajectory. As expected, this LWPR model was unable to predict enough torques in the operating range of the PE trajectory. Hence, the LWPR model had to be retrained on the PE trajectory, taking roughly 240 seconds to achieve good performance in the presence of friction. This simulation was then repeated with a model that was initialized through the use of motor babbling (Peters and Schaal, 2008). Here, a joint space trajectory was made by randomly selecting a point in the robot's expected operating range about which small sinusoidal trajectories were executed by the joints. This sequence was repeated at different points until sufficient coverage of the operating range was seen. Roughly 10,000 training points were generated from motor babbling. A PD controller was then used to track this trajectory and the resulting data was used to train the LWPR model. As seen in Table 4, by initializing the LWPR model this way, good performance on the PE trajectory could be learned in half the time compared to the case without motor babbling.

Even without any a-priori knowledge of the structure or parameters of the dynamics, LWPR is able to learn the inverse dynamics function of a manipulator accurately enough to yield near-optimal control results within minutes of training. Unlike the adaptive controller which relies on persistence of excitation for tracking performance, the LWPR approach can be trained on an arbitrary trajectory provided that it is given sufficient time to learn.

When tracking a PE trajectory with a known dynamic model, the ACT clearly outperforms the LWPR controller in terms of tracking accuracy and adaptation time, which is expected due to its incorporation of a-priori knowledge. However, if LWPR is presented with sufficient time to learn, its performance will closely approach that of ACT, but will not surpass it due to the use of local linear approximations of the system dynamics. However, the ACT controller is at a disadvantage since not all trajectories meet the PE requirement. For this reason, the identification of system parameters is often done offline on a predetermined trajectory which is optimized to yield the best parameter estimates (Khosla, 1989). While this may yield results better than the LWPR controller, the benefits of online, incremental learning are lost, where LWPR excels.

The performance of LWPR outside of areas in which it has trained is poor. This was clearly illustrated when a large perturbation to the inertia parameters caused the manipulator to move outside its trained region. This problem can be overcome through proper initialization of the model through motor babbling. However, for applications in which the manipulator workspace is large, significant amounts of training covering the robot's entire workspace must be carried out in order to achieve good results. This is not an issue for the adaptive controller, as it incorporates a-priori knowledge of the system dynamics, which is applicable to the entire workspace of the robot.

5. CONCLUSIONS AND FUTURE WORK

In this paper, the performance of model based controllers was evaluated against the learning approach to robot manipulator control. In particular, resolved acceleration and adaptive computed torque control were compared in simulation to Locally Weighted Projection Regression.

The performance of the RA control scheme was highly dependent upon accurate knowledge of the dynamic parameters of the system. A slight perturbation in the actual inertia and friction parameters of the system caused a decrease in performance, while the LWPR and ACT performance remained constant.

The ACT approach was able to outperform the RA controller in most scenarios due to its ability to generate online estimates of the actual inertia parameters of the system. It was shown that when the trajectory is persistently exciting and the structure of the dynamics is well known, the ACT controller also outperforms the LWPR. However, in practice, not all trajectories are PE and the structure of the dynamics of friction may not be known well. Hence, it is expected that when dealing with a physical robot, the simplified models of friction used in this paper will further degrade the performance of the ACT controller while the LWPR will be able to learn the additional nonlinearities present in the physical robot.

Lastly, although the LWPR controller was able to handle parametric uncertainty without any a-priori knowledge of the system, its greatest limitation is its local learning, which dictates that successful performance requires adequate initial training of the system. It was observed that significant perturbations caused the system to operate outside of its trained region, resulting in poor performance. Initializing the model through motor babbling can partially mitigate this issue, but the larger the workspace of the robot, the larger the training data set must be.

Future studies will involve experimental validation on physical robots as well as incorporating a-priori knowledge of the dynamic model of the system to improve performance of learning methods outside of the trained regions.

REFERENCES

Arimoto, S., Kawamura, S., and Miyazaki, F. (1984). Bettering operation of robots by learning. *Journal of Robotic Systems*, 1(2), 123–140.

Armstrong-Hélouvry, B., Dupont, P., and Canudas de Wit, C. (1994). A survey of models, analysis tools and compensation methods for the control of machines with friction. *Automatica*, 30(7), 1083–1138.

Ayusawa, K., Venture, G., and Nakamura, Y. (2008). Identification of humanoid robots dynamics using floating-base motion dynamics. In *IEEE/RSJ Int Conf on Intelligent Robots and Systems*, 2854–2859.

Bristow, D., Tharayil, M., and Alleyne, A. (2006). A survey of iterative learning control. *IEEE Control Systems Magazine*, 26(3), 96–114.

Burdet, E. and Codourey, A. (1998). Evaluation of parametric and nonparametric nonlinear adaptive controllers. *Robotica*, 16(1), 59–73.

Corke, P. (1996). A robotics toolbox for MATLAB. *IEEE Robotics and Automation Magazine*, 3(1), 24–32.

Craig, J., Hsu, P., and Sastry, S. (1986). Adaptive control of mechanical manipulators. In *IEEE Int Conf on Robotics and Automation*.

Khosla, P. (1989). Categorization of parameters in the dynamic robot model. *IEEE Trans on Robotics and Automation*, 5(3), 261 –268.

Landau, I. and Horowitz, R. (1988). Synthesis of adaptive controllers for robot manipulators using a passive feedback systems approach. In *IEEE Int Conf on Robotics and Automation*.

Nakanishi, J., Cory, R., Mistry, M., Peters, J., and Schaal, S. (2008). Operational Space Control: A Theoretical and Empirical Comparison. *Int Journal of Robotics Research*, 27(6), 737–757.

Nguyen-Tuong, D., Peters, J., Seeger, M., Schlkopf, B., and Verleysen, M. (2008). Learning inverse dynamics: A comparison. URL `http://edoc.mpg.de/420029`.

Nguyen-Tuong, D., Scholkopf, B., and Peters, J. (2009). Sparse online model learning for robot control with support vector regression. In *IEEE/RSJ Int Conf on Intelligent Robots and Systems*, 3121–3126.

Ortega, R. and Spong, M. (1988). Adaptive motion control of rigid robots: a tutorial. In *IEEE Conf on Decision and Control*, 1575–1584.

Peters, J. and Schaal, S. (2008). Learning to Control in Operational Space. *Int Journal of Robotics Research*, 27(2), 197–212.

Rasmussen, C. and Williams, C. (2006). *Gaussian Processes for Machine Learning*. The MIT Press.

Schaal, S., Atkeson, C., and Vijayakumar, S. (2002). Scalable techniques from nonparametric statistics for real time robot learning. *Appl. Intelligence*, 16, 49–60.

Sciavicco, L. and Scicliano, B. (2000). *Modelling and Control of Robot Manipulators*. Springer, 2nd edition.

Slotine, J.J.E. and Li, W. (1987). On the adaptive control of robot manipulators. *Int Journal of Robotics Research*, 6(3), 49–59.

Slotine, J.J.E. (1985). The robust control of robot manipulators. *Int Journal of Robotics Research*, 4(2), 49–64.

Vijayakumar, S., D'souza, A., and Schaal, S. (2005). Incremental online learning in high dimensions. *Neural Computation*, 17(12), 2602–2634.

Whitcomb, L., Rizzi, A., and Koditschek, D. (1993). Comparative experiments with a new adaptive controller for robot arms. *IEEE Trans on Robotics and Automation*, 9(1), 59–70.

Yao, B. and Tomizuka, M. (1996). Smooth robust adaptive sliding mode control with guaranteed transient performance. *Journal of Dynamic Systems, Measurement and Control*, 118(Dec), 764–775.

Yu, H. and Lloyd, S. (1995). Adaptive control of robot manipulators including motor dynamics. In *Proceedings of the American Control Conf*, 3803 –3807.