CONSTRUCTING HYDRAULIC ROBOT MODELS USING MEMORY-BASED LEARNING

Murali Krishna¹ John Bares²

ABSTRACT: Hydraulic machines used in mining and excavation applications are non-linear systems. Besides the non-linearity due to the dynamic coupling between the different links there are significant actuator non-linearities due to the inherent properties of the hydraulic system.

Optimal motion planning for these machines, i.e. planning motions that optimize a user-selectable combination of criteria such as time, energy etc., would help the designers of such machines, besides aiding the development of more productive robotic machines. Optimal motion planning in turn requires fast (computationally efficient) machine models in order to be practically usable.

This work proposes a method for constructing hydraulic machine models using memory-based learning. We demonstrate the approach by constructing a machine model of a 25-ton hydraulic excavator with a 10m maximum reach. The learning method is used to construct the hydraulic actuator model, and is used in conjunction with a linkage dynamic model to construct a complete excavator model which is much faster than an analytical model. Our test results show an average bucket tip position prediction error of 1m over 50 seconds of machine operation. This is better than any comparable speed model reported in the literature. The results also show that the approach effectively captures the interactions between the different hydraulic actuators.

The excavator model is used in a time-optimal motion planning scheme. We demonstrate the optimization results on a real excavator testbed to underscore the effectiveness of the model for optimal motion computation.

Portions of this paper were presented at the IEEE Intelligent Robot Systems Conference, Victoria, British Columbia, Canada, 1998.

1. Graduate Student, The Robotics Institute, Carnegie Mellon University, Pittsburgh, PA - 15213 (mkrishna@ri.cmu.edu)
2. Research Scientist, The Robotics Institute, Carnegie Mellon University, Pittsburgh, PA - 15213 (bares@ri.cmu.edu)

Key Words: Hydraulic Excavator, Modeling, Memory-based Learning.

1. INTRODUCTION

Time-optimal motion planning for robotic machines has been the subject of research for a number of years. The focus has primarily been on open-chain manipulators driven by electric actuators, since they are the most commonly used class of robots. However, the range of robot applications has been steadily growing and other types of robotic machinery are emerging.

Today, attention is being focused on machines used in outdoor field applications such as mass excavation and mining. The machines used in these applications are commonly driven by hydraulic actuators. A hydraulic excavator (HEX) is shown in Fig . It fills its bucket with material from a pile or a rock face, transports the bucket load to a waiting truck or conveyer belt, and dumps the load. Such tasks are ideal candidates for automation since they are repetitive and can reap huge benefits from increased productivity.

FIGURE 1. A Caterpillar 325 Hydraulic Excavator (HEX)

Automation of such applications is practical and cost-effective only if the robotic machine is more productive than a manually operated one. This requires that tasks be performed optimally to minimize not only time, but a combination of performance objectives such as time per bucket load and fuel consumption. Throughout this document the term “*optimal motion planning*” will refer to planning motions that optimize a combination of user-selected criteria.

Besides the use of optimal motion planning in building productive robotic machines, it also has the potential to aid in (non-robotic) machine design. Currently, machine designers have no means of knowing the effect of a design change on the productivity of a machine unless they build and test a prototype. While it is possible to simulate the effect of a design change on the machine’s step or ramp response, existing tools do not enable complete examination of cumulative motions that affect overall performance. An optimal motion planner could give a designer that information.

Optimal motion planning requires a machine model which defines the constraint surface for the path optimization problem. A complete machine model consists of an actuator model and a linkage dynamics model. “*Linkage dynamics model*” refers to the dynamics of the excavator’s links; “*Actuator model*” describes the actuator behavior - this term also includes the hydraulic system model; “*Machine model*” refers to a model which includes both of the above. The linkage dynamics for an excavator can be modeled using the well-known Newton-Euler equations. The actuator model is rather complex and non-linear, unlike that for electric-drive robots. The non-linearity is due to the fundamentally non-linear nature of fluid flow through an orifice, and the complex interactions that arise when several actuators share a power supply. In the case of the HEX, the four hydraulic actuators receive high pressure hydraulic fluid from two pumps. When multiple actuators request flow simultaneously the power demand may (and usually does) exceed the capacity of the engine. The system is forced to reduce the flow to the hydraulic cylinders to keep the engine from stalling. Such a power limited condition is very common during normal operation. These characteristics must be modeled if the machine is to

be used near a power-saturated condition. Slowing down the response to ensure that the saturated condition is never reached necessarily reduces the performance of the machine, which may be unacceptable.

Although optimal motion planning can be performed with slower machine models, fast models raise the possibility of computing the optimal motions as needed, even on-board the robot, rather than pre-computing it off-line. In this document, the terms “*fast model*” and “*slow model*” will refer to computationally inexpensive and expensive models respectively. Computing optimal motions may require tens of thousands of evaluations/simulations, and the speed difference between a slow and a fast model could translate into the optimization taking a few days versus a few hours.

Collision avoidance is a key requirement for a robotic machine. Fast models can also be used for on-line predictive simulation of motion commands before they are executed. The expected trajectory through space can be scanned for collisions and the robot stopped in time to prevent a collision. The use of predictive models is necessary when the masses are large and/or the velocities are high since the dynamics of the system can make the response quite different from a linear extrapolation of the velocity [Kelly 98].

This paper describes the construction of a fast actuator model for an excavator using memory-based learning. The learned model has been used to construct a complete excavator model, which includes a second-order linkage dynamics model in addition to the actuator model. This complete model is an order of magnitude faster than a comparable analytical model.

2. BACKGROUND

Background: Memory-based Learning and Neural Networks

Memory-based Learning (MBL) methods are becoming increasingly popular as computer memory sizes grow. MBL refers to the class of methods that use different regression techniques to model a function given a training data set [Atkeson 97]. MBL methods usually store the entire training data set and use subsets to make predictions, unlike neural networks that do not refer to the training data once the data has been learned using a method such as backpropagation. The training phase of a neural network consists of performing a gradient descent to compute the weights of the network which best fit the training data, while the training for a MBL method is trivial - it merely consists of storing the training data.

One of the main advantages of MBL techniques over neural networks is the explicit connection between an output prediction and the inputs. Thus, when using a MBL technique to learn a data set, it is possible to find the data points in the training data set that resulted in a certain output, which is almost impossible to do in a neural network where the connection between inputs and outputs is implicit.

Previous work: Learning the robot dynamics to improve control

Several researchers have used machine learning techniques to learn the robot linkage dynamics. In [Centikunt 91], [Narendra 90], and [Song 95], the authors use neural networks to learn the inverse dynamics of a robot manipulator. This learned model is then used in a model-based controller. In [Zomaya 95] the author uses a neural network to learn the error between an analytical dynamic model and actual robot behavior during operation of the controller. This learned error function is used to improve controller performance. In [Koivo 96] the authors use an autoregressive time-series vector difference equation to model the input-output data of an excavator's motion, which is then used in a self-tuning controller. The advantage of using a data based model to learn the robot's dynamic model is that the model can adjust to the specific characteristics of a particular robot instead of having to rely on the parameters provided by the robot manufacturer which may be inaccurate due to robot wear or manufacturing variations.

Although all the above cited researchers describe how the neural networks improved controller performance, they do not describe how well the neural network learned the dynamic model. This is probably because their goal was to improve controller performance and not learn the dynamic model.

Previous work: Hydraulic and pneumatic system models for control

Some researchers have worked on the problem of modeling hydraulic and pneumatic systems for real-time control. In [McDonnell 97] the authors describe the construction of an analytical pneumatic cylinder model which was used to improve the control. However, due to a large enough reservoir of high-pressure air their pneumatic robot does not encounter any flow limitations (and hence no actuator interactions of the type seen in a HEX).

In [Singh 95] the authors use a simple approach to handle the flow distribution between multiple hydraulic cylinders on a hydraulic machine during a flow-limited condition. They assume that the circuit with a valve closest to the pump gets all the flow it requires, and the remaining flow is distributed among the rest. This assumption is valid only if the cylinder closest to the pump has a much lower load than the other cylinders demanding flow. It is therefore limited in its modeling ability.

Testbed for this work

The focus of this work is directed towards hydraulic machines such as the 25-ton *Caterpillar 325* HEX in Fig , which is also the testbed used for this work. It has six actuated joints -

- two tracks which are independently controllable,
- a swing joint with a vertical axis of rotation, and
- boom, stick and bucket joints.

The boom, stick and bucket links are planar with axes of rotation normal to the plane of the links. All three joints are actuated by hydraulic cylinders visible in Fig . The swing joint uses a rotary hydraulic actuator (a hydraulic motor) not visible in the figure. The tracks are usually not actuated when excavating; they are periodically used to reposition the excavator. The HEX is powered by a single diesel engine which drives two hydraulic pumps. The hydraulic pumps take low-pressure hydraulic oil and output it at high pressure, which is then used to drive the hydraulic cylinders.

The HEX belongs to a general class of machines with the following characteristics:

- They use hydraulic actuators to drive the different joints,
- They are powered by a single engine or other power source located within the machine, and
- They are under-powered even during normal operation resulting in dynamic power redistribution, i.e. the actuators (and other systems powered by the engine) routinely request more power than the engine can supply. This causes the available power to be non-uniformly distributed amongst the different actuators due to different loads on each joint.

Sec 3 gives a brief description of the nature of the HEX modeling problem. The model construction details are described in Sec 4, followed by some results in Sec 5 which show the model's ability to simulate the response of the HEX. Before concluding in Sec 7 we describe results from an optimal motion planning application in Sec 6 that uses the model.

3. PROBLEM DESCRIPTION

The HEX can be viewed as a combination of linkages and hydraulic actuators as shown in Fig 2. The linkages are a system of masses being acted upon by force/torques applied by the hydraulic actuators. The complete HEX model is

therefore a system of linkage dynamic equations and actuator dynamics, which must be solved simultaneously to obtain the HEX response.

FIGURE 2. HEX viewed as a combination of linkages and actuators

Linkage dynamics

The linkage dynamics are described by the system of Newton-Euler equations ([Craig 89], Page 177):

$$M(\underline{\Theta})\ddot{\underline{\Theta}} + V(\underline{\Theta}, \dot{\underline{\Theta}}) + G(\underline{\Theta}) = \underline{\Upsilon} \quad (\text{EQ } 1)$$

where M is the positive definite inertia matrix, V is the matrix of non-linear (coriolis and centripetal) terms, G is the vector of gravity terms, $\underline{\Theta}$ is the vector of joint angles (θ_i) and $\underline{\Upsilon}$ is the vector of torques (τ_i) applied on each joint.

Each joint torque τ_i in the torque vector $\underline{\Upsilon}$ is related to the corresponding cylinder force f_{cyl_i} via the transform:

$$\tau_i = f_{cyl_i} \cdot \frac{\partial}{\partial \theta_i} R_i(\theta_i) \quad (\text{EQ } 2)$$

where $R_i(\theta_i)$ is the non-linear function that maps cylinder extension l_i to the joint position θ_i :

$$l_i = R_i(\theta_i) \quad (\text{EQ } 3)$$

In the above equations, the subscript i denotes the different joints. The complete expressions for the excavator's kinematics and linkage dynamics can be found in [Vaha 91] and [Vaha 93]. The cylinder force f_{cyl_i} and the cylinder extensions l_i connect the linkage dynamic model and the actuator model.

FIGURE 3. Schematic of a typical hydraulic excavator
(All cylinders/motor are connected to linkages which are not shown)

Actuator model

Fig 3 shows a simplified schematic of the hydraulic system of the HEX. This schematic only corresponds to one direction of motion of the cylinders. One of the two pumps supplies the boom and bucket cylinders while the other supplies the stick cylinder and swing motor. The HEX has different work modes and this arrangement describes one work mode. Different flow routing is used for other work modes. For the rest of the discussion the tracks will not be mentioned since they are not used during excavation. When they are used, one pump is dedicated to each track motor.

FIGURE 4. Spool valve controls flow to hydraulic cylinders (and motors)

The flow from the pumps to the cylinders is controlled through variable-area-orifices shown in Fig 3. The orifice areas are controlled by the movement of a *spool* valve, as shown in Fig 4. The different orifice areas on the HEX are determined by the position of four *control spools* - one per joint. A single *spool* position determines all the orifice area variables for a joint. For example, the position of the boom *control spool* determines the boom pump-to-cylinder, cylinder-to-tank, and bypass orifice areas.

The hydraulic system described above is an *open-center system*. In an open-center system the pumps deliver some high pressure oil even when no actuator flow is being demanded - between 10% and 20% of maximum flow for the test-

bed in Fig . This “idle” flow goes to the tank through the bypass (or “center”) passages shown in Fig 3, Fig 4. When the actuators are being commanded to move, the bypass passages slowly close and are fully closed when maximum flow is being demanded.

A complete analytical model of the hydraulic excavator includes orifice flow equations, fluid compressibility equations for all the oil volumes, as well as the force balance equations for all the cylinders. The orifice flow equation governing flow and pressure drop across an orifice for turbulent flow is:

$$Q = C_d A \sqrt{\Delta P} \quad (\text{EQ } 4)$$

where Q is the flow rate through an orifice, C_d is the orifice discharge coefficient, A is the orifice area, and ΔP is the pressure drop across the orifice.

Fluid compressibility can be modeled using the following equation:

$$\dot{P} = -\frac{\beta Q}{V_{cv}} \quad (\text{EQ } 5)$$

where $\dot{P}$ is the derivative of pressure in a control volume, β is the bulk modulus of the oil, V_{cv} is the volume of oil in the control volume, and Q is the flow rate through the control volume.

FIGURE 5. Schematic of bi-directional hydraulic cylinder

The force balance equation for a hydraulic cylinder is (Fig 5):

$$(m\ddot{l} = P_1 A_1 - P_2 A_2 - f_{cyl} - f_{friction})_i \quad (\text{EQ } 6)$$

where m is the mass of the cylinder rod, l is the cylinder extension, $P_j (j = 1, 2)$ is the pressure in the two cylinder chambers, $A_j (j = 1, 2)$ is the surface area on the two sides of the cylinder piston, f_{cyl} is the force on the cylinder due to the linkages and is the same term that also appears in Eqn 2, $f_{friction}$ is the friction force on the piston, and subscript i denotes the different joints. In the work described in this paper, no forces (e.g. digging forces) are applied at the bucket tip - only free space motion is considered. Therefore f_{cyl} only consists of forces due to the linkage dynamics.

A complete analytical model of the excavator which includes linkage and actuator dynamics is a coupled eighth-order non-linear system of equations which is partially described in [Medanic 97]. It includes multiple orifice equations (Eqn 4), multiple compressibility equations for all the oil volumes (Eqn 5), multiple flow balance equations, force balance equations for each cylinder (Eqn 6, Eqn 3) and the linkage dynamic equations (Eqn 1, Eqn 2). The HEX hydraulic system also has non-linear components such as the check-valves (shown in Fig 3) which prevent oil flow from the cylinder to the pump.

A detailed analytical model of the complete excavator hydraulic system has been constructed using a proprietary numerical solver, and its performance has been verified against results obtained from the HEX testbed. This detailed model takes approximately 100 seconds to simulate 1 second of excavator motion when running on a SUN Sparc20 workstation. This model is far too slow for use in optimal motion planning which requires tens of thousands of simulations of candidate motions. We therefore use a non-analytical approach to constructing the actuator model.

4. MODEL CONSTRUCTION

Since the system of analytical equations that describe the actuator model is computationally complex, we use memory-based learning to learn the mapping from variables that affect the actuator response to the actuator response itself. One purpose of this HEX model is to aid in optimal motion computation which requires many robot motion simulations. In this application it is not essential to capture all the transients in the robot response since the time period of the oscillatory transients is much smaller than the time-window of interest. As a result most of the oscillatory error due to a steady-state approximation would integrate to zero. For instance, the lowest resonant frequency of the boom joint is 1.5 Hz while our time window of interest is of the order of 5-10 secs. We therefore use memory-based learning to construct a *steady-state* actuator model for the four joints of the HEX. Memory-based learning was selected since it offers an explicit connection between the training data set and the prediction, a feature that proves to be quite valuable.

A multivariate locally weighted linear regression [Atkeson 97] is used to learn the mapping from the space of inputs (spool positions and cylinder loads) to the space of outputs (actuator velocities) for each HEX joint. To populate the actuator model space we can collect data either from the detailed analytical model or from the actual HEX testbed. Once the space is populated we can generalize to predict the actuator velocities for any set of inputs (the query). For instance, such a scheme can be used to predict the boom actuator steady-state velocity for a given set of boom and bucket spool positions, and boom and bucket cylinder loads. We need to use a regression scheme to make predictions since a general query will not necessarily coincide with a point in the training data set. Also, the data will not be regularly distributed in the actuator model space. The memory-based learning is performed using a software tool - *Vizier* - developed by Moore et. al. ([Moore 97a][Moore 97b]) at Carnegie Mellon University. *Vizier* uses locally weighted linear regression to make predictions from a training data set.

Model structure details

To understand the construction of the actuator model it is useful to first understand the basic concept behind the physical operation of the hydraulic system shown in Fig 3. The flow of hydraulic oil into a cylinder determines the velocity of that cylinder. The flow from the pump is distributed between the three parallel paths. The amount of flow going down each branch is determined by the ratio of the resistances of the different branches. The resistance offered by each branch is due to the orifice area and the cylinder load. For a given set of orifice areas - boom, bucket and bypass - and for a given set of boom and bucket cylinder loads, the distribution of flow between the different paths can be determined for a *steady-state* condition. This in turn allows the determination of the *steady-state* cylinder velocities. This relationship is illustrated in Fig 6. For ease of viewing 3-dimensional slices of the boom and bucket (*steady-state*) velocity maps are plotted (as a function of boom and bucket spool positions) for fixed boom and bucket cylinder loads of 300 kN and 15kN respectively. The data for these plots was obtained using the detailed analytical HEX model. The equations in play here are those described in Sec 3 (except the compressibility equation Eqn 5 - since compressibility is not a factor in the *steady-state* solution).

Spool positions of ± 11 mm indicate that the pump-to-cylinder areas are fully open, while a spool position of zero indicates that no flow is being sent into the cylinder. A positive spool displacement routes flow from the pump in a man-

ner that results in a positive cylinder velocity (cylinder extension) while a negative spool position causes cylinder retraction (See Fig 4).

**FIGURE 6. (a) Boom cylinder velocity as a function of boom/bucket spool positions
(b) Bucket cylinder velocity as a function of boom/bucket spool positions
(In both cases boom cyl. force = 300kN, bk. cyl. force = 15kN)**

Examining the boom-spool-to-boom-velocity relationship in Fig 6(a) for a bucket spool position of zero we see that the boom velocity increases to 0.13 m/s almost monotonically, with a dead-band around zero. However, when the bucket spool is fully open at -11 mm the maximum velocity that the boom can achieve (0.03 m/s) is much lower than the 0.13 m/s that it could achieve when the bucket spool was at zero. This is because the bucket *steals* most of the available flow because of its lower cylinder force, resulting in little flow for the boom. Fig 6(b) shows the surface for the bucket cylinder velocity. As we can see the bucket cylinder is fairly insensitive to the boom spool position since the bucket force in this case is much lower than the boom force. The peak bucket velocity is affected only for large values of the boom spool displacement - it changes from -0.45m/s to -0.3m/s as the boom spool moves from closed (0 mm) to fully open (+11 mm). These surfaces will be different for different force loads since the distribution of flow between the circuits will change.

The swing is more complicated to model since the inertial acceleration term in the dynamic equation is quite significant. This makes it impractical to use the same assumptions used for the other three joints: the boom, stick, and bucket, are in a vertical plane and have a large gravity load term in the dynamic equation, which is almost zero for the swing (It is zero if the HEX is on a perfectly flat surface). The acceleration phases of those joints are rather short during a typical move while the swing joint response is dominated by the acceleration and deceleration phases. The swing map

therefore uses swing velocity as an extra input. The output dimension is swing acceleration. Note that the swing inertia is dependent on the position of boom, stick, and bucket joints.

	Input #1	Input #2	Input #3	Input #4	Input #5	Output
Bm. Map	Bm. Spool Pos.	Bk. Spool Pos.	Bm. Cyl. Force	Bk. Cyl. Force	Not applicable	Bm. Cyl. Vel.
Bk. Map	Bm. Spool Pos.	Bk. Spool Pos.	Bm. Cyl. Force	Bk. Cyl. Force	Not applicable	Bk. Cyl. Vel.
St. Map	Sw. Spool Pos.	St. Spool Pos.	Sw. Inertia	St. Cyl. Force	Not applicable	St. Cyl. Vel.
Sw. Map	Sw. Spool Pos.	St. Spool Pos.	Sw. Inertia	St. Cyl. Force	Sw. Vel.	Sw. Accn.

TABLE 1: Summary of inputs and outputs for the different joint maps.

Data collection

Vizier is used to construct four memory-based learning maps (MBL maps) - one for each of the four joints of the excavator. The input and output dimensions for all the MBL maps are listed in Table 1. Training data for learning the actuator model was collected using the detailed analytical machine model which had been verified to match the testbed. The testbed was not used to collect training data due to our inability to measure spool positions directly on it. This slow model was driven through a number of motion sequences to adequately cover the entire operating space for each MBL map. The spool positions have a range of ± 11 mm. Data was sampled at a resolution of 1 mm along the spool position axes. While the spool positions are directly controllable the cylinder forces are not. The cylinder forces are determined by the excavator's configuration. The motions were therefore repeated for a fully loaded bucket, half-empty bucket, and completely empty bucket and for a range of HEX linkage configurations. All motions were performed slowly to minimize transient effects. The boom and bucket maps had 5500 points each while the stick and swing maps had 6400 points each. Fig 6 is an example of the type of data collected for the maps.

Details of the Memory-based Learning

Vizier performs a locally weighted linear regression to model the training data. The best parameters for the locally weighted linear regression are determined by a blackbox utility available within *Vizier* which performs leave-one-out cross-validation on the learning data set.

When making a prediction for a query point $\underline{q}_i$, the weight for a neighbor $\underline{q}_j$ is computed as:

$$w_i = \exp\left(\frac{-L(\underline{q}_i, \underline{q}_j)^2}{K_w^2}\right) \quad (\text{EQ } 7)$$

In Eqn 7, L is the scaled Euclidean distance between $\underline{q}_i$ and $\underline{q}_j$, and K_w is the width of the *kernel* for which the weighting function is applied. The *kernel* width determines the weights associated with neighbors of the query point - the farther the neighbor, the lower the weight, and hence the lower its contribution to the prediction. The Euclidean distance L (L_2 norm) is computed as:

$$L(\underline{q}_i, \underline{q}_j) = \sqrt{(\underline{q}_i - \underline{q}_j)^T \Sigma (\underline{q}_i - \underline{q}_j)} \quad (\text{EQ } 8)$$

where Σ is a diagonal normalizing matrix.

The locally linear model is:

$$P = Q_{aug} \beta_{aug} \quad (\text{EQ } 9)$$

where P is the prediction matrix with one column for each output attribute and one row for each data point, Q_{aug} is an augmented query matrix with one column for each input attribute and one row for each data point and an additional column whose value is always 1, and β_{aug} is an augmented matrix with the number of rows being one more than the number of inputs and as many columns as the number of outputs. For a given set of data, β is computed as:

$$\beta = ((WQ)^T (WQ))^{-1} ((WQ)^T (WP)) \quad (\text{EQ } 10)$$

where W is a diagonal matrix composed of the weights. Refer to [Moore 97a], [Moore 97b] [Atkeson 97] for more details of the regression.

The locally weighted linear regression used to model the data uses a kernel width of

$$K_w = \left(\frac{2^{-6}}{\text{RangeAlongDim}} \right) \quad (\text{EQ } 11)$$

for the spool position dimensions for all the maps, and also for the swing velocity dimension in the swing map. In Eqn 11 *RangeAlongDim* is the range along the input dimension for which the kernel is being computed. The regression is globally linear with respect to the force dimensions for all the maps.

Complete model construction

The complete excavator model is constructed by partitioning the actuator dynamics and linkage dynamics into two separate problems. Instead of solving them simultaneously they are solved in a serial fashion. The steps are:

- Step #1: The linkage dynamic model (Eqn 1) is used to compute the force loads on the different hydraulic cylinders. This force is assumed to remain constant over the time period that the actuator response is simulated using the actuator model.
- Step #2: The force loads computed in Step #1 are used with the given spool commands to predict the resulting cylinder velocities using the corresponding MBL maps. The swing map also uses the current swing velocity to compute the swing acceleration.
- Step #3: The computed velocities (or accelerations) are integrated to obtain an updated excavator state. (Repeat steps 1 through 3). We used a fixed time step of 0.02 sec, which was found through iteration to yield a satisfactory result.

Modeling actuator delays

The HEX actuators exhibit a certain delay between the issuing of the command and the actuator response. A clear example of stick actuator delay can be seen in Fig 8(d) in the testbed response. There are two main reasons for HEX actuator delays. One is due to the fact that a smaller hydraulic system (*pilot* system) is used to move the larger main spool valves. This results in a delay between when the spool valve motion is initiated by activating the pilot system, and

when the main spool valve actually shifts. The second cause of the delay is due to the finite pump response, and the time it takes to fill a cylinder chamber. Our model only captures the former delay since it is easy to model as a first-order lag. The latter cause of the delay is not modeled since it is dependent on the state of the pump and cylinder chambers when motion is initiated. Our steady-state actuator model cannot capture such effects.

5. RESULTS

This approach was used to construct a complete model of the HEX. The performance of the model was evaluated by comparing it to the HEX testbed. The results from three tests are shown in Fig 7 through Fig 10. In all the tests a human operator in the cab of the HEX moved joysticks which control the orifice areas, and hence the cylinder velocities. No soil interaction was involved during any of the tests and therefore no digging forces at the bucket tip were applied. In Fig 7 and Fig 8 the actuator commands shown in the fifth subplot at the bottom are the displacements of the main hydraulic spools caused by the joystick commands.

The main hydraulic spool displacements were computed indirectly for these plots. The joysticks activate solenoids, which move the pilot spool valves, and which in turn move the main hydraulic spools. The relationship between the joystick and solenoid displacements is fixed by the HEX manufacturer. No delay was associated with the solenoid response. We used a simple first-order lag model for the relation between the pilot and main spool displacements.

FIGURE 7. Position and Velocity Plots (Test #1)

During the first test (Fig 7) the boom and bucket cylinders were actuated to demonstrate the interaction between them. The boom and bucket joints were both commanded to move but due to the actuator interaction the heavier boom joint did not receive much flow until the bucket had stopped moving at $t=4.7$ secs. The second test (Fig 8) demonstrates interaction between the swing and stick joints. The swing velocity can be seen to increase dramatically once the stick

stops moving at $t=7.8$ secs. This inter-actuator coupling is because the excavator reaches a power limit, resulting in insufficient actuator flow.

FIGURE 8. Position and Velocity Plots (Test #2)

In the third test (Fig 9-Fig 10) the operator performed operations similar to that during normal loading operations, but without terrain interaction. Fig 9 shows plots of the four joint velocities. Fig 10 shows the total error in the prediction of the bucket tip position. This is plotted along with the distance from the base of the boom joint to the bucket tip to give the reader an idea of the scale of the tip position error.

FIGURE 9. Joint velocities (Test #3)

FIGURE 10. Overall bucket tip position prediction error (Test #3)

The model used for these tests requires 1 sec. of computation time on a SUN Sparc 20 workstation to simulate 20 seconds of HEX motion. We also constructed an analytical steady-state model of the HEX with accuracy comparable to the above described model and it was found to be an order of magnitude slower.

6. APPLICATION

One application for the machine model developed here is in planning the optimal free-space motion between specified start and end states. Our optimal motion computation approach involves discretizing the *input* space and then performing a search in that space to compute the most optimal command sequence. An advantage of using low-level joint controllers is that the closed-loop HEX (Fig 11) is a lower order system, thus allowing a coarser discretization of the command *input* space. The *inputs* for the closed-loop HEX are the joint position commands (u_{joint} ; ref. Fig 11). The controllers also help insulate the robot from disturbances.

FIGURE 11. Complete closed-loop HEX machine model

The HEX testbed has independent PD (proportional-derivative) joint position controllers for each of its four joints. The joint controllers are modeled in state-space form as:

$$\dot{x}_{joint} = A_{joint}x_{joint} + B_{joint}u_{joint} \quad (\text{EQ } 12)$$

$$y_{joint} = C_{joint}x_{joint} + D_{joint}u_{joint} \quad (\text{EQ } 13)$$

where, u_{joint} is the position command to the joint, x_{joint} is a 5 element vector of state variables, y_{joint} is the spool position command, and A_{joint} , B_{joint} , C_{joint} , D_{joint} are the state-space matrices for the controllers. These controllers cannot be described here in greater detail for proprietary reasons.

Since we wish to primarily demonstrate the use of this modeling approach in optimal motion computation, we will treat the optimizer as a black box whose details will be described in a future paper. The optimizer takes four inputs -

- Task specification - an initial and final position
- Task constraints - can include obstacles in the environment that the robot should avoid, in addition to task specific constraints such as - "The robot should not open the bucket until it reaches the truck".

- Robot model - describes a constraint surface for the optimization since it embodies the limitations of the robot. We used the MBL-based model for the optimization.
- Initial guess - this provides the seed to start the search for the optimum. The search approach used to compute the optimum is not very sensitive to the choice of the initial guess.

**FIGURE 12. (a) Initial position: $S_w = -90^\circ$; $B_m = -4^\circ$; $S_t = -82^\circ$; $B_k = -27^\circ$;
 (b) An intermediate position for the initial guess motion
 (c) Final position: $S_w = -180^\circ$; $B_m = 25^\circ$; $S_t = -94^\circ$; $B_k = -50^\circ$;**

The initial, and final positions used in our example are shown in Fig 12(a) and (c). An intermediate position of the initial guess is shown in Fig 12(b). The initial guess is chosen to clear the obstacle and takes approx. 12 secs to reach the goal. The joint angles in Fig 13 and Fig 14 are shown in degrees. We use the Denavit-Hartenberg convention ([Craig 89] [Vaha 91]) for the kinematics of the excavator.

FIGURE 13. Initial guess for the example task

The optimal command sequence and resulting motions are plotted in Fig 14. The final motion takes 8 secs to reach the goal, which is about 30% faster than the initial guess. In Fig 14 the simulated response obtained using the HEX model is plotted along with the response obtained from the HEX testbed. The model's prediction agrees closely with the testbed response.

FIGURE 14. Optimal commands and resulting motions

7. CONCLUDING REMARKS

A model of a hydraulic excavator has been constructed by decomposing the complete 8th order non-linear system of equations into two separate models - the 2nd order linkage dynamic model and the actuator model. The models were used in a serial fashion with the results of one being fed to the other at every time step. The non-linear actuator model was learned using memory-based learning techniques.

The results show that the learned model does a good job of predicting the response of a hydraulic excavator during a typical excavating cycle. The mean bucket tip position prediction error is around 1.0m for most of the 50 secs of motion in Test #3 (1.5m = width of the excavator's bucket). This model is focused at capturing the interactions between the different actuators of the excavator and it manages to accomplish this goal fairly well.

The excavator model created using this approach is much faster than a comparable analytical model. This approach to model construction is ideal for applications where the model is used for extensive simulation, and where time is a constraint. We have shown its use in a command space search scheme to compute the optimal robot commands for a free-space motion task, while avoiding obstacles in the environment. The model can also be used for on-line collision avoidance by checking commands for the possibility of collision before execution.

Another advantage of using the proposed approach is that changing relationships between the different variables can be easily incorporated in the model. For example, the HEX has work modes where the flow routing is different from that described in Fig 3. Modeling these work modes only requires changing the structure of the MBL maps and repopulating the MBL maps - the rest of the model and regression approach remains unchanged.

Traditionally, the approach to dealing with actuator saturation has been to reduce actuator speed, thus keeping the system in a linear regime of operation. While the speed reduction prevents actuator saturation and improves controllability, it also limits the peak performance of the robot. Having a model that captures power/flow saturation effects enables us to compute optimal motions that allow the robot to operate in a saturated condition.

However, the model cannot simulate actuator transients since the learned actuator model is a steady state model. Also, this method would not be suitable if the cylinder response (or output) is dependent on a large number of factors. As the number of input dimensions grow, the size of the MBL map increases exponentially, which slows the speed of the

predictions. In such cases it may be better to use neural networks, whose prediction times are not affected by the size of the training data set - only the network training time is increased.

APPENDIX I: REFERENCES

- [Atkeson 97] Atkeson, C.G., Moore, A.W., Schaal, S.A., "*Locally Weighted Learning*", AI Review, Vol. 11, pp. 11-73, 1997.
- [Centikunt 91] Centikunt, S., Chiu, H.T., "*A Study of Learning Controllers for Tip Position Control of a Flexible Arm using Neural Networks*", Proc. of the Winter Annual Meeting of the ASME, pp. 15-19, 1991.
- [Craig 89] Craig, John J., "*Introduction to Robotics: Mechanics and Control*", 2nd Ed., Addison-Wesley Publishing Company, 1989.
- [Kelly 98] Kelly, A., Stentz, A., "*An Approach to Rough Terrain Autonomous Mobility - Part I*", Autonomous Robots, Vol. 5, pp. 129-161, 1998.
- [Koivo 96] Koivo, A.J., Nease, A.D., "*Adaptive Self-Tuning Control of Excavators*", Proc. of the Conf. on Robotics for Challenging Environments, pp. 220-6, June 1-6 1996, Pub by the ASCE.
- [Lawrence 95] Lawrence, P.D., et al., "*Coordinated and Force-Feedback Control of Hydraulic Excavators*", Fourth International Symposium on Experimental Robotics, Proc. ISER '95, Pub. by Springer-Verlag, June 30 - July 2 1995.
- [Lin 96] Lin, L.C., Yih, T.W., "*Rigid Model-Based Neural Network Control of Flexible-Link Manipulators*", IEEE Transaction on Robotics and Automation, Vol. 12, No. 4, pp. 595-602, Aug. 1996.
- [McDonell 97] McDonell, B.W., Bobrow, J.E., "*Modeling, Identification, and Control of a Pneumatically Actuated Robot*", Proceedings of the IEEE Conference on Robotics and Automation, pp. 124-9, April 1997.
- [Medanic 97] Medanic, J., Yuan, M., Medanic, B., "*Robust Multivariable Non-linear Control of a Two Link Excavator: Part I*", Proc. IEEE Conference on Decision and Control Systems, pp. 4231-6, Dec 1997.
- [Moore 97a] Moore, A., Atkeson, C., Schaal, S., "*Locally Weighted Learning for Control*", AI Review, Vol. 11, pp. 75-113, 1997.
- [Moore 97b] Moore, A.W., Schneider, J., Deng, K., "*Efficient Locally Weighted Polynomial Regression Predictions*", Proc. 1997 Intl. Machine Learning Conf., Morgan Kaufmann Publishers, San Francisco, CA.
- [Narendra 90] Narendra, K.S., Parthasarathy, K., "*Identification and control of dynamical systems using neural networks*", IEEE Tran. on Neural Networks, Vol. 1, pp. 4-27, 1990.
- [Singh 95] Singh, N., et al., "*Coordinated Motion Control of Heavy Duty Industrial Machines with Redundancy*", Robotica, Vol. 13, pp. 623-633, 1995
- [Song 95] Song, B., Koivo, A.J., "*Neural Adaptive Control of Excavators*", Proceedings of the Intelligent Robots and Systems Conference, Pittsburgh, PA, pp. 162-7, April 1995.
- [Vaha 91] Vaha, P.K., Skibniewski, M.J., Koivo, A.J., "*Kinematics and Trajectory Planning for Robotic Excavation*", Proc. ASCE Construction Congress II, Cambridge, MA, 1991.
- [Vaha 93] Vaha, P.K., Skibniewski, M.J., "*Dynamic Model of Excavator*", ASCE Journal of Aerospace Engineering, Vol. 6, No. 2, April 1993.
- [Zomaya 95] Zomaya, A.Y., "*An Error-learning Neural Network for Tuning of Robot Dynamic Models*", International Journal of Systems Science, Vol. 26, No. 1, pp. 13-31, 1995.

APPENDIX II: NOTATION

The following symbols are used in this paper:

Q	= Rate of flow of hydraulic oil;
C_d	= Orifice discharge coefficient;
P	= Fluid pressure;
β	= bulk modulus of oil;
V_{cv}	= Volume of oil in control volume;
A_1, A_2	= Cylinder piston areas;
f_{cyl}	= Cylinder force load;
$f_{friction}$	= Cylinder friction;
m	= Mass of rod of hydraulic cylinder;
l	= Hydraulic cylinder rod displacement;
M	= Excavator's positive definite inertia matrix;
V	= Matrix consisting of non-linear coriolis and centripetal terms;
G	= Gravity vector;
$\underline{\Theta}$	= Vector of excavator's joint angles;
$\underline{\Upsilon}$	= Vector of excavator's joint torques;
R	= Function that maps linear cylinder motion to angular joint motion;
t	= time;
$\underline{q_i}, \underline{q_j}$	= Vectors representing points in a training data set;
w_i	= Scalar weight assigned to a neighbor of a query;
L	= Scaled Euclidean distance function;
K_w	= Kernel width used computing the weights assigned to a query neighbor;
P	= Matrix representing the output of a memory-based learner;
Q, Q_{aug}	= Matrices representing the input to a memory-based learner;
β, β_{aug}	= Matrices used in the locally linear model for the memory-based learner;
W	= Diagonal matrix composed of the weights w_i ;
$u(t)$	= Excavator joint position commands to the closed-loop excavator robot;
A, B, C, D	= State space matrices for the excavator's PD controller;
$y(t)$	= Spool position commands generated by the controller;
$x(t)$	= Vector of state variables for each joint;