Image Cover Sheet

CLASSIFICATION	SYSTEM NUMBER 149721
UNCLASSIFIED	

TITLE

A PRACTICAL DYNAMIC MODEL OF A PUMA 560 MANIPULATOR FOR AIRCRAFT MAINTENANCE

AN: 95-00526

System Number:

Patron Number:

Requester:

Notes:

DSIS Use only:

Deliver to: NL

A Practical Dynamic Model of a Puma 560 Manipulator for Aircraft Maintenance

Gordon Hardy
Defence Research Establishment Pacific
FMO Victoria, B.C. V0S 1B0, Canada

Abstract

Manipulator arms performing aircraft maintenance must have high positioning and orientation accuracy. External position sensors may be necessary to achieve this and a dynamic model of the manipulator is required to use the sensor information optimally. This paper describes a system-identification-based method to determine a computationally efficient dynamic model for a PUMA 560 manipulator that can be easily adapted to other manipulators. The resulting model consists of a sample and hold, a delay, and a plant. The model was derived by fitting standard first- and second-order functions to responses measured at a variety of positions, speeds and motion directions. Statistical analysis of the model coefficients indicated that the manipulator's steady state gain is sensitive to speed and direction. In addition, it showed that radial and vertical motion can be modeled adequately with a first-order plant, but tangential motion requires a second-order plant. When subjected to a white noise input, the model estimated the measured position with a root mean square error of 0.8 mm.

Introduction

Background

Currently, aircraft such as the CF-18 are manually inspected for defects by scanning suspected areas with ultrasonic or magnetic sensors. A PUMA 560 manipulator was used at the Defence Research Establishment Pacific (DREP) to demonstrate the feasibility of using a robotic based scanner to perform non-destructive aircraft inspection. Such a scanner has the advantages of producing repeatable results and reducing tedious, error-prone work. Inspection devices must be accurate enough to repeatedly detect delaminations 1 mm long in composite structures, and to keep the ultrasonic transducer normal to the surface. For most manipulators, including the PUMA, external position sensors are required to achieve these criteria. However, a dynamic model of the manipulator is required to optimally use these sensors' data.

Existing Dynamic Models

One approach to modeling a robot's dynamic behavior, is to model its individual components, the control algorithm and the physical model. The control algorithm receives a command to go to a position and uses sensor information to direct the joint motors. The physical model relates the motor torques to the joint accelerations.

The control algorithm first translates the position command to joint angles by solving the inverse kinematic problem. The solution requires 24 parameters including joint lengths, linear offsets and angular offsets. These parameters can be difficult to obtain for a specific robot arm. The control algorithm employs angular position and velocity sensors to drive the joint motors to the joint angle vector solution. To make matters worse the control algorithm is typically not published

Physical models are also difficult to implement. Fu[1] described three conventional physical models: Lagrange-Euler, Newton Euler and generalized D'Alembert. Unfortunately, these models are either computationally slow or complex to derive. Their greatest problem, however, is that they require measurements of mass, moments of inertia and length for each manipulator link. To illustrate the difficulty, Armstrong[2] reported having to disassemble an arm to measure these properties.

A Practical Approach

A more practical and generic method uses system identification theory to determine the transfer functions relating the requested-position input to the end-effector-position output. This method does not require knowledge of the internal control algorithms or the physical parameters of the manipulator. Bossert[3] described a method to determine the model of a PUMA 560 manipulator based on system identification theory. He related torque to angular position for each joint. Whereas in this paper, the position command is related directly to the end-effector position, thus treating the entire manipulator as a black box. The method involves measuring manipulator responses and fitting standard step-responses to them. First-order transfer function coefficients include: steady state gain (K), time delay (T_d), and the time constant (τ). Second-order coefficients include steady state gain, time delay, natural frequency (ω) and the damping factor (ζ). Analysis of variance was used to determine the dependency of the model coefficients to speed, position and direction of motion. The most dependent relationships were then incorporated into the model.

PUMA 560 "ALTER" Mode

Like many manipulators, the PUMA 560 has a facility to use external sensor information. Normally, user written programs direct robotic motion from one position to another. In "ALTER" mode, the PUMA receives position correction commands to direct motion, which must come every 28 ms from an external computer ("external ALTER") or an internal task ("internal ALTER"). The method described in this report can be applied to any manipulator that accepts position commands in a similar way.

The System Identification Approach

A manipulator can be represented as a linear model (Fig. 1) for each of three directions in a cylindrical coordinate system: radial or x-direction, to and from the robot base, tangential or y-direction, and vertical or z-direction. When operating in "ALTER" mode the PUMA 560 manipulator receives a motion command every 28 ms, which is modeled by a sample and hold transfer function

$$G_H(s) = \frac{1 - e^{-Ts}}{s} \quad (1)$$

where T - update period = 28 ms

s - Laplace domain independent variable.

After receiving the command, the arm jumps to the destination. This action is modeled by a plant, P, and a delay, $G_D(s)$.

$$G_D(s) = e^{-sT_d} \quad (2)$$

where T_d - time delay (s).

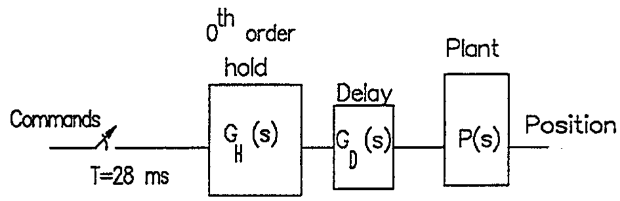

FIGURE 1: Block diagram of "ALTER" motion.

Identifying the Plant

Like the impulse response, the step response can identify a plant, but unlike the impulse response, the step response is more practical to measure. In this case, a step response is a recording of the end-effector position when the arm receives a position command.

Step Response Models

The measured step response of the manipulator is fitted to standard first- and second-order step responses. The first-order plant representation is

$$H(s) = \frac{K}{1 + \tau s} \quad (3)$$

where K - steady state gain

τ - time constant

H - plant representation.

Multiplying the above equation by a step function, and translating to the time domain gives the first-order step response

$$c(t) = K(1 - e^{-t/\tau}). \quad (4)$$

The standard second-order representation is

$$H(s) = K \frac{\omega^2}{s^2 + 2\zeta\omega s + \omega^2} \quad (5)$$

where H(s) - second-order transfer function

K - steady state gain

ω - natural frequency

ζ - damping factor.

The second-order step response depends on whether the system is over damped or under damped. If over damped (ζ>1), the step response is [4]

$$c(t) = K + \frac{K\omega}{2\sqrt{\zeta^2 - 1}}\left(\frac{e^{-s_1 t}}{s_1} - \frac{e^{-s_2 t}}{s_2}\right), \quad (6)$$

where $$s_1 = \omega\left(\zeta + \sqrt{\zeta^2 - 1}\right) \quad (7)$$

$$s_2 = \omega\left(\zeta - \sqrt{\zeta^2 - 1}\right). \quad (8)$$

If under damped (ζ<1), the step response is [4]

$$c(t) = K - \frac{Ke^{-\zeta\omega t}}{\sqrt{1 - \zeta^2}} *$$

$$\sin\left(\omega t\sqrt{1 - \zeta^2} + \tan^{-1}\frac{\sqrt{1 - \zeta^2}}{\zeta}\right). \quad (9)$$

Measuring the Step Responses

Configuration

Fig. 2 shows the test bed used to measure the step responses for a PUMA 560 manipulator. All measurements were taken with no end-effector load. The arm was oriented so that the radial axis corresponded to the puma's internal y-axis.

Measuring position as a function of time

The response to a step input was measured using a toothed wheel connected to a potentiometer, as shown in Fig. 3. A wooden platform structure was placed in contact with the wheel so that when the arm moved, the wheel would rotate. The resulting voltage change was recorded on an oscilloscope. A photograph of the oscilloscope traces was manually digitized. To measure time delay, the program toggled a digital output to indicate on the oscilloscope when the step command was given.

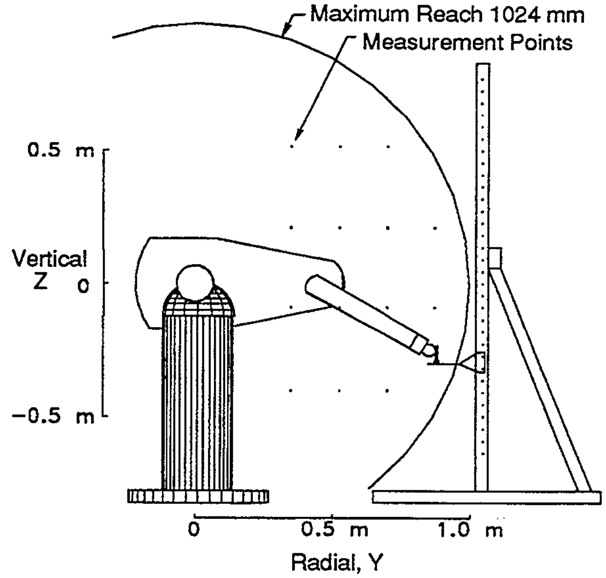

FIGURE 2: Manipulator test bed.

VAL II program

The essential parts of the program producing a step in the x-direction are listed in Appendix A. It was written in a BASIC-like robotic language called VAL II, running on the PUMA 560 controller. It consists of two routines: "move", a routine interrogating the user and initializing the robot; and "jumpx.pc", a background task producing a step command synchronous to the internal 28 ms clock.

Measurement Parameters Considered

Three test parameters that may affect the dynamic response of the manipulator were considered. Position and motion directions were considered because they involve different inertias which change the dynamic model. Speed was considered because the arm response may differ with speed. Taking measurements throughout the entire work volume was unnecessary, since the arm inertias remain constant with rotation around the vertical axis. Consequently, only 14 positions within a vertical slice of the volume were examined (see Fig. 2). Tests were done at four different speeds (0.071, 0.14, 0.28 and 0.68 m/s) and three different motion directions (tangential, radial, and vertical). Fig. 2 defines two of the three axes.

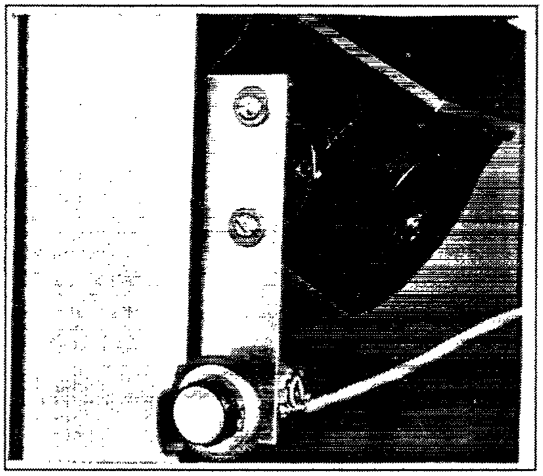

FIGURE 3: End effector.

Determining and Combining Model Coefficients

Using Minimization to Determine the Model Coefficients

A trial was run for 86 combinations of position, speed and direction. For each trial, the manipulator response was measured and fitted with a standard first- and second-order step response. Each solution set includes five model coefficients: delay, steady state gain, time constant τ, natural frequency ω, damping factor ζ. Time delay and gain were measured directly. The other terms were determined by minimizing a root mean square (RMS) error function that compared the measured step responses to a standard first- and second-order step response model using the program MATLAB[5].

$$\text{RMS error} = \sqrt{\frac{\sum_{i=1}^{N}\sum_{k=1}^{K}\left(\mathrm{R}_i(\mathrm{T}_s k) - M(\mathrm{T}_s k)\right)^2}{N \cdot K}} \tag{10}$$

where, T_S - sampling period = 5.63 ms

K - number of samples = 30

M - model position predicted

- First-order model : Equation 4

- Second-order model : Equation 6 or 9

R - measured step response

N - number of step responses.

Starting with an initial estimate, MATLAB's minimization algorithm varied the model coefficients to converge the error function to its minimum.

Local minima

A major concern with using a minimization algorithm is the danger of converging to a local minimum. Plotting the error function as a function of the model coefficients can indicate potential problems. In many cases, such plots indicate a unique solution. However, in some cases multiple solutions were observed. For example, Fig. 4 shows contours of the RMS error function as a function of natural frequency and damping factor. The error function surface is smooth, indicating no isolated local minima. However, the surface has a long flat minimum, indicated by the dotted line segment. Cases with this structure have multiple second-order solutions, indicating that a first-order solution may suffice (corroborated in the section entitled "Graphical Method").

Combining the Solution Sets

To create a practical model, the 86 solution sets must be combined into one model. Not all the coefficients will vary significantly as the test parameters change. Some coefficients may vary for different positions, speeds or motion directions, and therefore, have to be accounted for in the final model. Other coefficients may prove to be independent of the parameters, and can be treated as constants. Analysis of variance[6] can quantify the dependency of the coefficients on the experimental parameters, providing that the coefficient values are normally distributed.

Normality of the coefficients

Coefficient histograms compiled from the solution sets showed which were normally distributed. Histograms for ω, ζ and RMS errors were clearly not bell shaped, and therefore not normally distributed. Fig. 5 shows histograms of the remaining coefficients considered to have a near normal distribution.

Graphical method

The sensitivity of the non-normally distributed coefficients, ω and ζ, and the RMS errors, can be tested graphically. Fig. 6 shows the dependency of the RMS errors. Each subplot was composed by sorting the solution sets from all 86 trials into groups according to the subplot's corresponding parameter. The range of values for each group is illustrated as a two-standard-deviation error bar. The RMS error values were used as a measure of how well the responses fit standard step functions. Fig. 6a shows that to ensure an accuracy of less than 1 mm, the final model should only be used for y-positions less than 700 mm. Fig. 6d and 6h indicate that tangential motion is best modeled with a second order plant. Graphs for ω and ζ were not included because their values approached infinity, consequently their averages and graphs were meaningless.

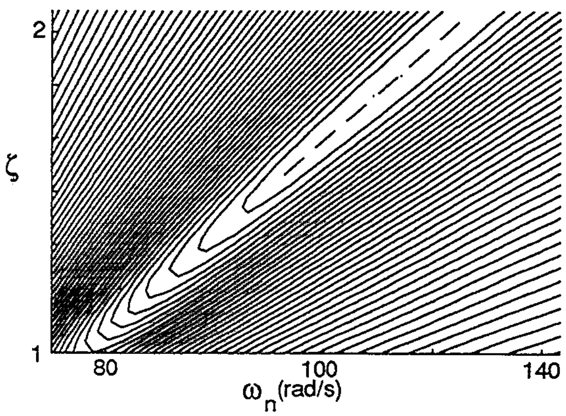

FIGURE 4: A contour plot of a second-order error function with multiple solutions.

Statistical method using analysis of variance

Analysis of variance[6] tests how sensitive the normally distributed coefficients are to the experimental parameters. The results in Table I are expressed as an i-ratio (the ratio of the result to the value in the 95%-confidence f-distribution table[6]). If the ratio is less than one, the means are statistically the same and consequently the corresponding coefficient and parameter are independent.

The final model should first account for the relationships with the greatest dependencies (highest i-ratio). Table I shows that the steady-state gain is highly dependent on motion direction; consequently, the model includes a separate equation for each direction. Table I also shows that the gain is dependent on speed and consequently, the gain is expressed as a linear function of speed. Other dependent relationships are not incorporated in the model, because they would greatly increase model complexity with only a small increase in accuracy.

Since τ is independent of all parameters, it is regarded as a constant.

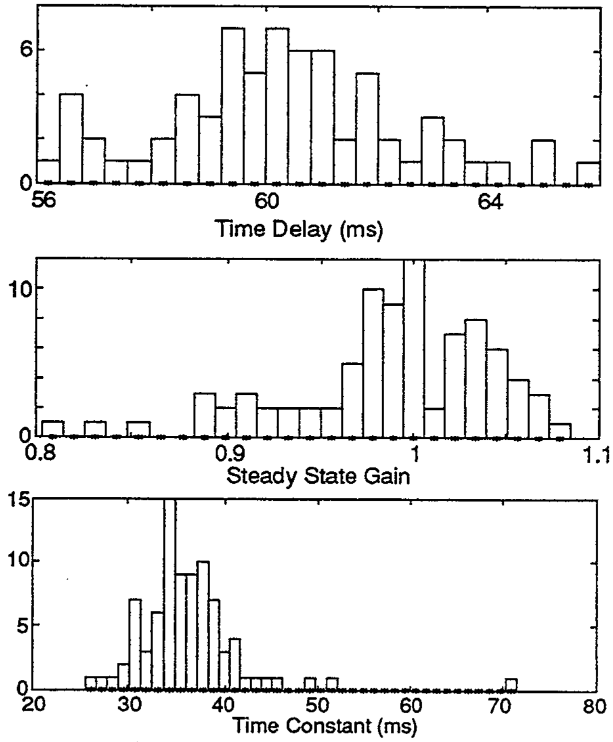

FIGURE 5: Histograms of the model coefficients.

The Combined Model

The resulting model consists of a transfer function for each of the three directions along the axes of a cylindrical coordinate system. Each transfer function, shown in Fig. 1, consists of a 28-ms sample and hold, a delay, and a plant.

Delay

"Internal ALTER" mode

For the program listed in Appendix A, the average measured delay was 60.45 ms. This figure greatly depends on what instructions are executed between the "WAIT" command, which waits for an internal 28 ms clock transition, and the "ALTOUT" command, which starts motion.

"External ALTER" mode

In "external ALTER" mode, the delay from when the puma requests data to when it starts moving is 38.5 ms. A complete discussion appears in the manual[7].

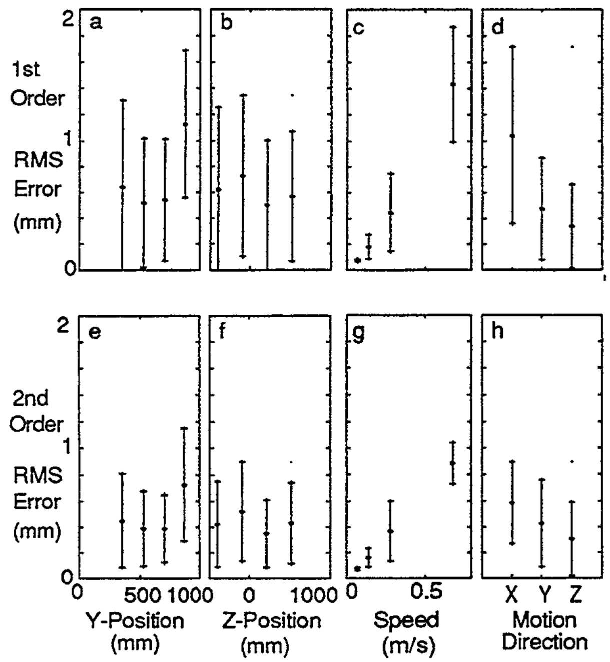

FIGURE 6: Dependence of RMS error coefficients to position, speed and direction of motion.

The Plant

From 68 trials, a combined plant was created and is summarized in Table II. Steady state gain is expressed as a linear function of speed.

$$k = a\,s + b \quad (11)$$

where, k - steady state gain

s - speed (in m/s)

a,b - constants.

Motion in the radial and vertical direction is modeled with a first-order plant (Equation 3). The time constants are the average for all trials in the appropriate direction.

Motion in the tangential direction is best modeled with a second-order plant (Equation 5). Since ζ and ω approach infinity, an average value would produce a poorly fitting model. Consequently, another minimization of the error function (Equation 10) was performed using data from all the tangential trials.

TABLE II: Combined model coefficients

direction	a (m/s)	b	ζ	ω (rad/s)	τ (ms)
x (tangential)	0.0518	1.010	0.864	53.9	
y (radial)	0.0818	0.933			42.5
z (vertical)	0.134	0.941			35.6

Model Performance

The combined-model RMS error was calculated for each trial. The results were grouped according to their y-position. The mean and standard deviation of

TABLE I: Model coefficients sensitivity to measurement parameters

parameter	value	delay (ms) mean	delay standard deviation	delay i-ratio	steady state gain mean	gain standard deviation	gain i-ratio	τ (ms) mean	τ standard deviation	τ i-ratio
speed	0.14 m/s	60.4	2.5		0.95	0.06		38.0	4.8	
	0.28 m/s	60.0	1.9	0.52[a]	1.00	0.04	4.57	36.3	6.8	0.11[a]
	0.68 m/s	61.3	2.2		1.02	0.03		39.0	22.5	
direction	x	62.1	1.7		1.04	0.028		34.9	3.5	
	y	60.4	2.1	1.58	0.96	0.058	7.44	42.6	20.5	0.37[a]
	z	59.9	2.1		0.97	0.042		35.6	4.2	
y[b] position	350 mm	61.3	2.3		0.99	0.058		34.5	3.3	
	525 mm	60.0	2.0	0.73[a]	1.00	0.044	0.37[a]	35.8	4.0	1.11
	700 mm	60.3	1.5		0.97	0.066		37.6	5.1	
z position	-400 mm	61.7	2.0		0.96	0.064		34.7	5.5	
	-90 mm	59.5	2.2	1.63	1.00	0.042	1.12	41.2	21.5	0.42[a]
	210 mm	60.8	2.0		0.99	0.061		37.0	4.3	
	515 mm	59.8	1.5		1.01	0.042		36.7	3.4	

[a] Considered independent

[b] 870 mm absent because it is greater than the 700-mm model constraint

each group are compared in Fig. 7a. It shows that for 1 mm accuracy, the model should be used for y-positions less than 700 mm. The procedure was repeated for the remaining three parameters. Fig. 7c shows that model error is approximately proportional to speed and for 1 mm accuracy, the speed should be less than 0.5 m/s

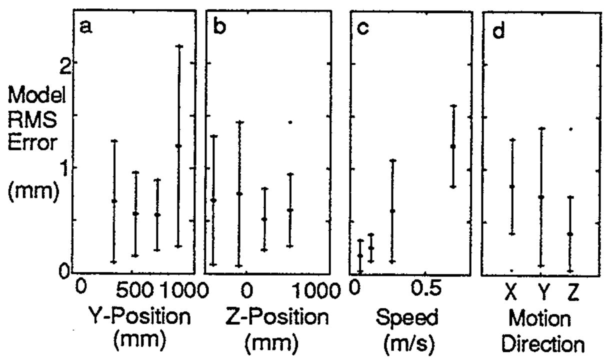

FIGURE 7. Model error verses the four parameters grouped by position, speed and direction of motion.

Model Verification Using White Noise

Most models can be verified by comparing the responses of the simulated and the actual systems to a white noise input[8]. The chosen white-noise signal, shown in Fig. 8, is a binary random signal that switches between states with a 50% probability every 28 ms.

Figs. 8 and 9 summarize the simulated and actual responses to tangential, radial and vertical motion respectively. Measurements were taken at a single point (Y=700 mm, Z=-90 mm) using the same position sensor shown in Fig. 3. A digital storage oscilloscope recorded the input and the measured position, with 8-bit resolution every 4 ms, and transferred them to two data files. A MATLAB[5] routine read the input data, fed the result into the model, and compared the output to the measured position. To compare the input and output, a bias was applied to give them an average value of zero[8].

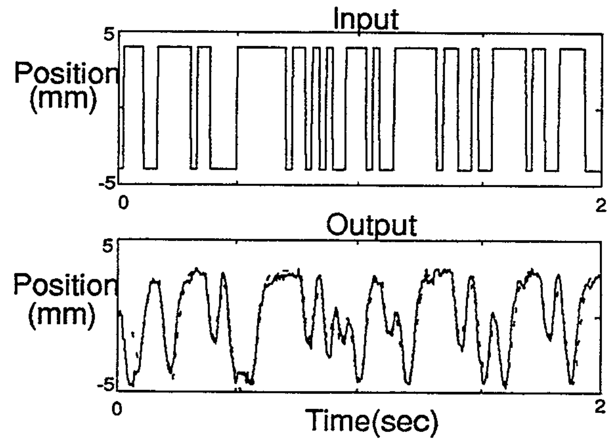

FIGURE 8: Model (dashed line) and measured (solid line) response to white noise in tangential direction.

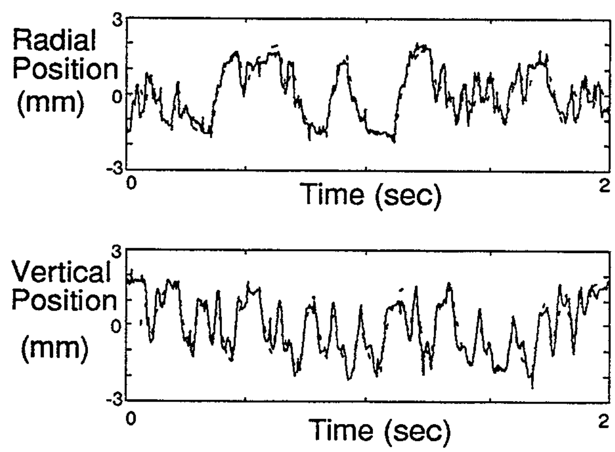

FIGURE 9: Model (dashed line) and measured (solid line) response to white noise in radial and vertical direction.

Table III tabulates the RMS error for the data shown in Figs. 8 and 9. The tabulated data are well within the predicted model errors of Fig. 7.

The graphs and the table clearly indicate that the model accurately predicts the measured position, although not necessarily the actual position, and that the method used to derive the model is trustworthy.

TABLE III: White noise RMS error

Direction:	X (Tangential)	Y (Radial)	Z (Vertical)
RMS error (mm):	0.67	0.35	0.41

Conclusion

Significance

Manipulator arms performing aircraft maintenance must have high positioning and orientation accuracy. External position sensors may be necessary to achieve this criteria, but they require a dynamic model of the manipulator to use the sensor information optimally. This paper described a system-identification-based method to determine a computationally-efficient dynamic model for a PUMA 560 manipulator. The method can be applied to any manipulator with an operating mode like PUMA's "ALTER" mode.

The Model

The resulting PUMA 560 model consists of a transfer function for each of the three directions along the axes of a cylindrical coordinate system. Each transfer function consists of a 28-ms sample and hold, a delay, and a plant. The time delay is 60.45 ms for internally generated path corrections. The plant was derived by fitting standard first- and second-order step responses to step responses measured for different parameters (position, speed, and direction). Analysis of variance showed that the most sensitive model coefficient, steady state gain, is dependent on motion direction and speed. A first-order plant is capable of modeling radial and vertical motion, but a second-order plant is needed for tangential motion.

Performance

The resulting model worked best for radial distances less than 700 mm from the robot base and end-effector speeds less than 0.5 m/s. The root-mean-square position error of the model is roughly proportional to the end-effector speed, and has an average value of 0.8 mm.

When subjected to white noise, a simulation of the model tracked the actual measured position within the predicted error. The white noise tests verified that the model is sufficiently accurate and the method trustworthy.

References

1. K.S. Fu, R. Gonzalez, C. Lee, "Robotics: Control, Sensing, Vision and Intelligence",McGraw Hill, Toronto, 1987.

2. B. Armstrong, B. Khatib, O. Khatib, J. Burdick, "The Explicit Dynamic Model And Inertial Parameters Of The Puma 560 Arm", IEEE International Conference On Robotics And Automation, 1, Pg 510-18, 1986.

3. D. Bossert, G.Lamont, M. Leahy, I. Horowitz, "Model-based Control with Quantitative Feedback Theory; Empirical Model Analysis",Proc. of the 29th IEEE Conference on Decision and Control, Pg 1974-5, 1990.

4. K. Ogata, "Modern Control Engineering", Prentice Hall Inc., Toronto, 1970.

5. Computer code MATLAB (The Math Works Inc., South Natick, M.A., 1990)

6. R. Walpole, R. Myers, "Probability And Statistics For Engineers And Scientists", Macmillian Publishers Co. Inc., New York, 1978.

7 "User's Guide to VAL II", Unimation, A Westinghouse Company, Pittsburgh, 1986.

8. L. Ljung, "System Identification, Theory for the User", Prentice-Hall Inc., Toronto, 1987.

Appendix A: VAL Program Listings

```
.PROGRAM move
        ...
  8     ALTER (-1, 18)  ;non-cumulative world
          coordinates, alter enabled
  9     PCEXECUTE jumpx.pc, -1, 0
        ...
.PROGRAM jumpx.pc
  1     x = 0
  2     angle = angle+step
  3     IF angle >= 10 THEN      ; if triggered
  4       x = a                  ;step size
  5       SIGNAL -17
  6     ELSE
  7       SIGNAL 17
  8     END
  9     TYPE /N, x, angle        ;print to queue
 10     ALTOUT 0, x*TODIS
 11     WAIT                     ;until 28 ms
```