System Identification Solution for Developing an AdeptThree Robot Arm Model

Adelhard Beni Rehiara

***Abstract*— An AdeptThree robot arm is a SCARA robot that widest working envelope in its class. The robot should be represented in its mathematical model in order to be able to study the behavior and also to design a controller for the robot. Jacobean matrix of a robot can be used to determine the position and motion of a robot end-effector. In this paper, an AdeptThree robot was selected to develop its represented model and a Jacobean model was also built for the robot. System identification method had been chosen to build four models of each joint of the AdeptThree robot in time domain system. All of the models were accepted after passing both whiteness and independent tests in residual analysis plots.**

***Index Terms*— AdeptThree robot, SCARA, System Identification, Jacobean.**

I. Introduction

Robots are mostly used to replace workers in such dangerous, high precision or in routine and repeated works. Robots often do a better job than human. A robot arm is a type of robot which works similarly to a human arm. A robot arm or also called manipulator is composed of a set of joints separated in space by the arm links and it looks like our own wrist and elbow [1].

Due to imprecision, it is not easy to control the robot motion to do an appropriate job perfectly. The imprecision will happen along the robot motion or operation and it might be caused by its structure or control algorithm. Dynamic parameters of the robot also will not able to be brought together into the robot model because of the imprecision. On the other hand, knowledge about the robot parameter values should match to the robot system in detail to get a good robot model.

The uncertainty in modeling a robot will make difficulty to design its good model. Therefore, system identification is often needed to include the uncertainty into the robot model. System identification is widely used in engineering and non-engineering areas [2]. The system identification gives possibility to construct a model from experimental data.

An AdeptThree robot arm is a selectively compliant assembly robot arm (SCARA) manufactured by the Adept Company. In general, traditional SCARA's are 4-axis robot arms within their work envelope. They have the jointed two-link arm layout similar to our human arms and commonly used in pick-and-place, assembly, and packaging applications. As a SCARA robot, an AdeptThree robot has 4 joints which mean that it has 4 degree of freedom (DOF). The Adept robot has been designed with complete components including operating system and programming language namely V+ [1].

Adelhard Beni Rehiara is with the University of Papua, Manokwari, 98314 Indonesia (e-mail: adelhard.rehiara@fmipa.unipa.ac.id).

Some studies and researches were done in the field of robot arm i.e. Bulent [3] studied about possibility to used two cooperating SCARA manipulator while doing a job, Mustafa [4] implemented fuzzy and neural network for control 3-DOF robot manipulator, Rasit [5] presented a neural network for solving 3-joint robot inverse kinematics, Toshio [6] introduced a neural network in case of robotic arm on-line learning,. The work presented in this paper is aimed to investigate and to develop the mathematical model of an AdeptThree robot arm and its Jacobean model.

II. Supporting Theory

A. System identification

System or process identification is the field of mathematical modeling of systems (processes) from test or experimental data. The input-output data are usually collected from an identification test or experiment that is designed to make the measured data maximally informative about the system properties that are of interest to the user [7]. A system identification procedure can be exploited in which experimental data is used directly [8], whereas identification produces usually input-output models [9].

A general procedure for process estimation includes the following steps [9]:

- Determination of the model structure. This often makes use of empirical experience about the process, or information from some basic experiments
- Parameter estimation. The procedure for parameter estimation depends on the type and characteristics of the process input, as well as the desired model structure.
- Model verification. A suitable model should agree with the experimental data, it should describe the process accurately, and it should meet the purpose in which it was obtained for. Further, it can be verified whether the parameters obtained are within physical limits. It is also possible to reduce the model and to compare it with the original model to see if a simpler model suffices.

Three ways to define a mathematical model in system identification are:

- White box modeling. A model will be built from a clear system which is mean that all of its parameters can be found.
- Grey box modeling. In this case, many parameters are known but not all of these parameters. A measurement can be used to construct unknown parameter(s). This is a collaborated model between white box and black box modeling.
- Black box modeling. In this kind of system identification, no parameter is known. The way to get the model is by using data measurement. Data input and output will be compared to implement the relationship of both data.

In order to get a good model, a model should be validated. MATLAB System Identification Toolbox has provided two kinds of model validation by using residual analysis plots or Akaike's Final Prediction Error (FPE) criterion.

Residual Analysis Plots

Residual analysis plots can be used for either time domain or frequency domain input-output data. In time domain validation data, the plots will show both autocorrelation function of residuals for each model and cross-correlation between the input and the residuals for each input-output pairs. Conversely, for frequency domain validation data, the plots will give both estimated power spectrum of residuals for each model and transfer-function amplitude from the input to the residuals for each input-output pairs [10].

Residual analysis consists of whiteness test and independent test. In the whiteness test criteria, a good model has autocorrelation function inside the confidence interval of the corresponding estimates, indicating that the residuals are uncorrelated. The confidence interval corresponds to the range of residuals value with specific probability of being statistically significant for the system. On the other hand, in independent test, a good model has residuals uncorrelated with past inputs. The evidence indicates that the model does not describe how part of the output relates to the corresponding input [10].

Akaike's Criterion

Akaike's criterion provides a measure of model quality by simulating the situation where the model is tested on different data set. The FPE of Akaike's criterion is formulated as follows [10].

$$FPE = V\left(\frac{1+\frac{d}{N}}{1-\frac{d}{N}}\right) \tag{1}$$

$$V = \det\left(\frac{1}{N}\sum_{1}^{N}\varepsilon(t,\theta_N)(\varepsilon(t,\theta_N))^T\right) \tag{2}$$

Where V is the loss function, d is the number of estimated parameters, N is the values number in estimation data set and θ_N represents the estimated parameters. By assuming that FPE is asymptotic for d<<N, the eq.1 can be simplified as:

$$FPE = V\left(1 + 2d/N\right) \tag{3}$$

The FPE can be computed for either linear or non linear model and according to Akaike's theory, a model that has smallest FPE is the most accurate model [10].

B. Kinematics

The aim of kinematics is to define relative position of a frame to its original coordinates. Basic vector algebra can be used to solve the problem in kinematics. The sequences to get the relative position are finding the A matrixes, building an arm T matrix and calculating it with the coordinate position which is desired.

A matrix

The A matrix is a homogenous 4x4 transformation matrix which describe the position of a point on an object and the orientation of the object in a three dimensional space. The A matrixes can be built by using the Denavid-Hartenberg (D-H) convention with the relation of frame *i-1* and base frame *i* given in following equation[11].

$$A_i^{i-1} = \begin{bmatrix} \cos\theta_i & -\cos\alpha_i\sin\theta_i & \sin\alpha_i\sin\theta_i & a_i\cos\theta_i \\ \sin\theta_i & \cos\alpha_i\cos\theta_i & -\sin\alpha_i\cos\theta_i & a_i\sin\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4}$$

The matrix contains link parameters and joint parameters. The link parameters are α_i which is the twist of the link i and α_i which is the length of link i. The joint parameters are joint angle θ_i and the joint offset d_i. The matrix can be simplified as follows [11].

$$A_i^{i-1} = \begin{bmatrix} R_i & P_i \\ 0 & 1 \end{bmatrix} \tag{5}$$

Where R_i is a 3x3 rotation matrix and P_i is a 3x1 translation matrix.

Arm T Matrix

The arm T matrix is a kinematics chain of transformation. It can be built from 2 or more A matrixes which is shown by following equation [11].

$$T \equiv T_n^0 = A_1^0 A_2^1,\ldots, A_n^{n-1} \tag{6}$$

The arm T matrix is usable to obtain coordinates of a point in terms of the base link.

Direct Kinematics

To get the position of the frame which is relative to the base frame, the arm T matrix should be multiplied with the coordinate matrix r_n given by position of the end effector. A parameter, which is a scaling factor, should be added to the matrix r_n to become a 4x1 matrix and be able to be multiplied. The final matrix of the robot kinematics is [11],[12],[13]:

$$r_o = \begin{bmatrix} R_i & P_i \\ 0 & 1 \end{bmatrix} \begin{bmatrix} r_n \\ 1 \end{bmatrix} \quad (7)$$

The direct kinematics can be found in the matrix P_i. The *X, Y* and *Z* positions are P_1, P_2 and P_3 respectively.

Velocity Transformation

Given joint variable coordinate of the end effectors $q=[q_1, q_2 \ldots q_n]^T$. On the matrix *q*, $q=\theta$ for a rotary joint and $q=d$ for a prismatic joint. Then a generally nonlinear transformation from joint variable *q(t)* to *y(t)* is *y=h(q)* and the velocities of joint axes is given as[11][12]:

$$\dot{y} = \frac{\partial h}{\partial q}\dot{q} = J\dot{q} \quad (8)$$

Where *J* is the Jacobean manipulator and inverse of the Jacobean J^{-1} relates the change in the end-effector to the change in axis displacements.

The Jacobean is an important component in many robot control algorithms. Many ways to design a Jacobean matrix of a robot arm were presented. Zomaya *et al.* (1999) had presented three kinds of algorithms to perform a Jacobean matrix. The other algorithm which uses tool configuration vector ω was provided by Manjunath and Ardil (2007) and Frank *et al.*(2006) as follows:

$$w(q) = \left[\frac{P_i}{R_{i3} e^{(q_n/\pi)}} \right] \quad (9)$$

Therefore the Jacobean matrix can be found using following equation [11][12].

$$J(q) = \frac{\partial w}{\partial q_i} \quad (10)$$

III. Results

A. Physical System

An AdeptThree robot consists of 4 joints and the joint motions are revolution, revolution, prismatic and revolution (RRPR) respectively from 1st until 4th joint [1]. Fig. 1 shows the schematic figure and the joints of AdeptThree robot.

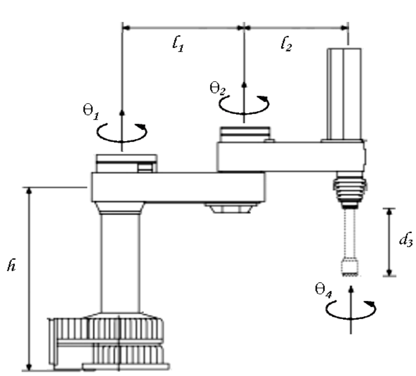


Fig. 1 Physic of AdeptThree Robot

The AdeptThree robot is controlled by a central processing unit (CPU). The CPU has input/output devices and communication devices. The input/output devices are keyboard, mouse, monitor, printer and robot arm itself. The CPU also supports the communication using some devices, such as area network (LAN), digital input/output (DIO), and serial port. With almost two meters in diameter of working area, the AdeptThree robot has widest working envelope in its class [1].

B. Mathematical Model

Many parameters of the AdeptThree robot are not provided by its manufacturing company. Therefore, the appropriate model will be constructed by assuming the robot as black box models and gaining the models by using MATLAB System Identification Toolbox based on the data input-output measured from the robot. The input to the black box is voltage and the output is either joint angle or link offset.

A number of experimental joint tracking has been made to collect the joint data. From the data, transfer functions of each joint were designed in time domain system identification. Each joint model is built using a 2nd order system with a gain (K), a zero (T_z), an integrator and a pole (T_p). The estimated model of first joint was fixed at 3rd iteration and the parameters of the model are K_1=0.22773, T_{z1}=0.004892 and T_{p1} =0.0015202. Second joint model was found at 5th iteration and its model parameters are K_2=0.24557, T_{z2}= 0.004892 and T_{p2}= 0.0014101. The model parameters of third joint and fourth joint are K_3 = 1.218, K_4= 2.1202, T_{z3}= 0.001106, T_{z4}=0.001517, T_{p3} and T_{p4}= 0.001. The third and fourth joints were defined in 7th iteration and 1st iteration for third and fourth joint respectively. Henceforward the transfer functions of first to fourth joints are given in the following equations.

$$H_1(s) = \frac{0.004892s + 0.22773}{0.0015202s^2 + s} \quad (12)$$

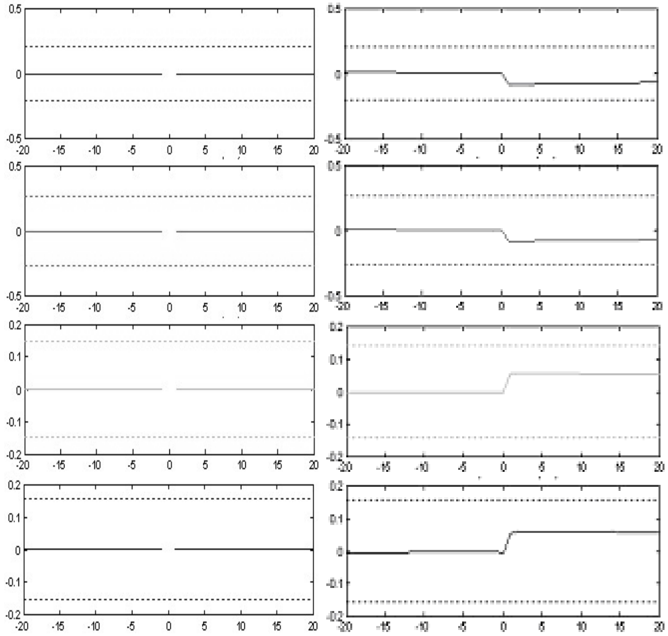
$$H_2(s) = \frac{0.004892s + 0.24557}{0.0014101s^2 + s} \quad (13)$$

$$H_3(s) = \frac{0.001106s + 1.218}{0.001s^2 + s} \quad (14)$$

$$H_4(s) = \frac{0.001517s + 2.1202}{0.001s^2 + s} \quad (15)$$

System Identification toolbox in MATLAB will also generate model validation automatically after a model had been built. The default method for validating model in this MATLAB toolbox is residual analysis plots. In this case the models for first, second, third and fourth joints are match at 99.81%, 99.81%, 100% and 100% respectively.

Using residual analysis plots for time domain input-output data, autocorrelations and cross correlations of each joint model were illustrated in fig.2 downward from 1st to 4th joint. The figure shows that the model residuals for both correlations of each joint are laid between the confidential lines (dotted lines). This is the evidence that all models are acceptable.



(a) (b)

Fig. 2 a) Autocorrelations and b) cross correlations

C. The Jacobean

The steps to get the relative position are calculating the A matrices and multiplying the A matrix of each joint to build an arm T matrix. To build the A matrices, first step that must be done is to define the DH parameters of the robot as shown in table 1.

TABLE I
DH PARAMETERS OF AN ADEPTTHREE ROBOT

Joint	θ_i	d_i	a_i	α_i
1^{st}	θ_1	h	l_1	0
2^{nd}	θ_2	0	l_2	0
3^{rd}	0	$-d_3$	0	π
4^{th}	θ_4	$-d_4$	0	0

Joints and links parameters are shown in fig.1. The joints are where the motion in the arm occurs while the links are a fixed construction. Thus the link has a fixed relationship between the joints [11].

The arm T matrix of the AdeptThree robot which is found using eq.2 and eq.3 is as follows.

$$T = \begin{bmatrix} -s_4 s_{1+2} + c_4 c_{1+2} & -s_4 c_{1+2} - c_4 s_{1+2} & 0 & l_2 c_{1+2} + l_1 c_1 \\ s_4 c_{1+2} + c_4 s_{1+2} & -s_4 s_{1+2} + c_4 c_{1+2} & 0 & l_2 s_{1+2} + l_1 s_1 \\ 0 & 0 & 1 & -d_4 - d_3 + h \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (16)$$

Where c_1, c_2, c_4 are the cosines of θ_1, θ_2, and θ_4; s_1, s_2, s_4 are the sinus of θ_1, θ_2, and θ_4; l_1 and l_2 are the length of link 1, and link 2; d_3 and d_4 are the length of link 3 and 4; h is the length of 1^{st} joint column.

By using the arm T matrix, it is possible to calculate the values of (P_x, P_y, P_z) with respect to the fixed coordinate system. Length of link 4 (d_4) was removed from the equations due to no hand gripper. Then the end-effector position obtained with direct kinematics is equations which are listed in following equations.

$$\begin{aligned} P_X &= l_2 c_{1+2} + l_1 c_1 \\ P_Y &= l_2 s_{1+2} + l_1 s_1 \\ P_Z &= h - d_3 - d_4 \end{aligned} \quad (17)$$

Where constant parameters l_1=559 mm, l_2=508 mm, and h=876.3 mm. The direct kinematics can be used to find the end-effector coordinate of the robot movement by substituting the constant parameter values to the above equation.

The manipulator's speeds are calculated by means of the Jacobean matrix. By knowing the speeds of the articulations we obtain the speed with which the SCARA describes a trajectory. With the equations of the kinematics models, the following matrix is obtained.

$$w(q) = \begin{bmatrix} l_2 c_{1+2} + l_1 c_1 \\ l_2 s_{1+2} + l_1 s_1 \\ -d_4 - d_3 + h \\ \hline 0 \\ 0 \\ e^{(\frac{\theta_4}{\pi})} \end{bmatrix} \quad (18)$$

From the formulation above, 4 tool configuration vectors can be built and a Jacobean 6x4 matrix is found from the vectors. The vectors are constructed using eq.7 and as the result four sub matrices forming the Jacobean matrix are obtained as follows.

$$J_1(q_1) = \begin{bmatrix} -l_2 s_{1+2} - l_1 s_1 \\ l_2 c_{1+2} + l_1 c_1 \\ 0 \\ \hline 0 \\ 0 \\ 0 \end{bmatrix} \quad J_2(q_2) = \begin{bmatrix} -l_2 s_{1+2} \\ l_2 c_{1+2} \\ 0 \\ \hline 0 \\ 0 \\ 0 \end{bmatrix}$$

$$J_3(q_3) = \begin{bmatrix} 0 \\ 0 \\ -1 \\ \hline 0 \\ 0 \\ 0 \end{bmatrix} \quad J_4(q_4) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \hline 0 \\ 0 \\ e^{\left(\frac{\theta_4}{\pi}\right)} / \pi \end{bmatrix} \quad (19)$$

The Jacobean is formed from the 4^{th} vectors as $J(q)=[J_1(q_1), J_2(q_2), J_3(q_3), J_4(q_4)]$. The similar Jacobean was also build by Manjunath and Ardil (2007) for a developed SCARA as follows.

$$J(q) = \begin{bmatrix} -l_2 s_{1+2} - l_1 s_1 & -l_2 s_{1+2} & 0 & 0 \\ l_2 c_{1+2} + l_1 c_1 & l_2 c_{1+2} & 0 & 0 \\ 0 & 0 & -1 & 0 \\ \hline 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & e^{\left(\frac{\theta_4}{\pi}\right)} / \pi \end{bmatrix} \quad (20)$$

To be specific for an AdeptThree robot, the constant parameters of the robot should be substituted into the eq.20. Finally, the Jacobean matrix for the AdeptThree robot is found in the following equation.

$$J(q) = \left[\begin{array}{cccc} -508\, s_{1+2} - 559\, s_1 & 508\, s_{1+2} & 0 & 0 \\ 508\, c_{1+2} + 559\, c_1 & -508\, c_{1+2} & 0 & 0 \\ 0 & 0 & -1 & 0 \\ \hline 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & e^{\left(\frac{\theta_4}{\pi}\right)} / \pi \end{array} \right] \quad (21)$$

The above Jacobean matrix is not a square matrix because it has 6x4 components. Because the Jacobean is not a square matrix, the matrix can not be inverted. Therefore, it can only be used to calculate the robot joint velocities. The Jacobean matrix of the AdeptThree robot which is built with using tool configuration vectors indicates that about 24 of 32 or about ¾ of the matrix components are equal to zero.

IV. Conclusion

MATLAB System Identification Toolbox was used in this project to develop the mathematical models of each joint of the AdeptThree robot. The models were designed with the data measurements, which contain voltages as the input and either joint angles or link offset as the output. Results of model validation show that all of the joint models are accepted.

In different way, Jacobean matrix of an AdeptThree robot also had been developed. The Jacobean consists of 6x4 components and it can be used for defining the joint speed of the robot. The Jacobean matrix of the AdeptThree robot is simple to be built by using tool configuration vectors.

References

[1] Rehiara Adelhard Beni, Smit Wim, *Controller Design of a Modeled AdeptThree Robot Arm*, Proceedings of the 2010 International Conference on Modelling, Identification and Control, Japan, July 17-19, 2010, pp 854-858.

[2] Mei Li, Chen Chen, Wenlin Liu, *Identification Based on MATLAB*, Proceedings of the 2009 International Workshop on Information Security and Application, November 21-22, 2009, China, pp 523-525.

[3] Bulent Ozkan, Kemal Ozgoren, *Invalid Joint Arrangements and Actuator Related Singular Configuration of a System of two Cooperating SCARA Manipulator*, Journal of Mechatronics, Vol.11, 2001, pp 491-507.

[4] Mustafa Nil, Ugur Yuzgec, Murat Sonmez, Bekir Cakir, *Fuzzy Neural Network Based Intelligent Controller for 3-DOF Robot Manipulator*, Procedings of Intelligent Manufacturing Systems Symposium, Sakarya University Turkey, May 29-31, 2006.

[5] Rasit Koker, Cemil Oz, Tarik Cakar, Huseyin Ekiz, *A Study of Neural Network Based Inverse Kinematics Solution for a Three-Joint Robot*, Journal of Robotics and Autonomous System, Vol.49, 2004, pp 227-234

[6] Toshio Tsuji, Yoshiyuki Tanaka, *On-line Learning of Robot Arm Impedance using Neural Network*, Journal of Robotics and Autonomous System, Vol.52, 2005, pp 257-271.

[7] Yucai Zhu, *Multivariable System Identification for Process Control*, Elsevier, London, 2001.

[8] Raymond A. de Callafon, Paul M. J. Van den Hof, *Multivariable Feedback Relevant System Identification of a Wafer Stepper System*, IEEE Transactions on Control Systems Technology, Vol. 9, No. 2, March 2001, pp 381-390.

[9] Ján Mikleš, Miroslav Fikar, *Process Modelling, Identification, and Control*, Springer-Verlag, Berlin, 2007.

[10] Lennart Ljung, *System Identification Toolbox 7 User's Guides*, The MathWorks Inc., Natick, 2010. Available at http://www.mathworks.com/help/pdf_doc/ident/ident.pdf

[11] Frank L.Lewis, Darren M.Dawson, Chaouki T.Abdallah, *Robot Manipulators Control*, Marcel Dekker, Inc., New York, 2006.

[12] Manjunath T.C., Ardil C., *Development of a Jacobean Model for 4-Axes indigenously developed SCARA System*, International Journal of Computer and Information Science and Engineering, Vol. 1 No 3, 2007, pp 152-158.

[13] John Faber Archila Diaz, Max Suell Dutra, Claudia Johana Diaz, *Design and Construction of a Manipulator Type Scara, Implementing a Control System*, Proceedings of COBEM, 19th International Congress of Mechanical Engineering, November 5-9, 2007, Brasília.

[14] Zomaya A.Y., Smitha H., Olariub S., *Computing robot Jacobians on meshes with multiple buses*, Microprocessors and Microsystems, no. 23, 1999, pp 309–324.

Adelhard Beni Rehiara received Bachelor degree in electrical engineering in 1999 from University of Widyagama, Malang, Indonesia. In 2008, he had gained Master degree in control systems engineering from HAN University, Arnhem, Netherlands. His main research interests include embedded systems, system optimization, modeling and control systems.