論文 89－26－11－19

Integrated Dynamic Modeling and Hardware Oriented Control Scheme for a S'mulator of an Industrial Robot

(산업용 로보트의 시뮬레이터를 위한 종합적인 동적모델링과 하드웨어 구성과 일치하는 제어구조)

李 敏 基*, 李 光 男*, 林 桂 榮*

(Min Ki Lee, Gwang Nam Lee, and Kye Young Lim)

要 約

본 논문은 산업용 로보트의 시뮬레이터 개발에 대하여 기술한다. 여기에서 개발된 시뮬레이터는 로보트 전체의 동적모델과 하드웨어 구성과 일치되는 제어구조로 특징 지울수 있다. 동적모델은 manipulator의 동특성 뿐만아니라 actuator 동특성을 포함하므로써 로보트 전체 시스템의 동적 모델링을 구현하였다. 제어구조는 일반적으로 산업용 로보트에서 사용되는 하드웨어 구조로 되어있다. 즉 PI 제어가 위치, 속도, 전류를 제어하고 PWM 발생기는 actuator에 공급하는 전압을 변조한다.

이러한 시뮬레이터는 산업용 로보트 시스템과 일치하므로 로보트 개발시 중요한 설계개념을 제공한다. 실제로 이 시뮬레이터는 본 연구소에서 개발한 SCARA 로보트에 적용되어 여러가지 설계 가변수를 변화시키면서 로보트의 성능과 특성을 관찰한다. 이러한 관찰은 actuator, control gain, trajectory planning 등을 선택하는 기준이 된다.

Abstract

This paper presents the development of a simulator for an industrial robot. The simulator is characterized by a fully integrated dynamic model and a hardware oriented control scheme. The dynamic model includes the actuator dynamics as well as the manipulator dynamics to integrate the entire dynamics of the robot system.

On the other hand, the control scheme is oriented as a hardware structure which is usually implemented in the industrial robot. That is to say, a conventional PI control law is used to regulate the position, the speed, and the current. A Pulse Wave Modulation (PWM) generator modulates the supplied voltage to the actuator. Since the simulator is consistent with the industrial robot system, it provides the essential design concepts for the development process of the robot.

In practice, the simulator is applied to the SCARA robot which has been developed in GSIS. Here, it investigates the characteristics and performance of the robot with changing design parameters. Thus, the investigation furnishes criteria for the selection of actuator, control gain, trajectory planning, etc.

*正會員, 金星産電(株)研究所
(GoldStar Industrial System Co., Ltd. R & D Lab.)

接受日字 : 1989年 4月 12日

I. Introduction

A large number of simulators have been used

for the development of a robot manipulator and controller. The simulator makes it possible for a robot designer to investigate the performance of the robot with various design parameters and to design the robot according to the desired specification. The simulation is conducted in a computer and it requires the dynamic model and the control scheme of the robot system. The main difficulty in the dynamic model is to attain the manipulator dynamics because it contains multidegrees of freedom and its dynamic equations are complex. Due to the complexity, researchers in the past have proposed the efficient dynamic formulations; Lagrange[1-3] Virtual Work[4], and Newton Euler[5-8]. These dynamic formulations are implemented in the simulator to analyze the constrained or unconstrained motion of the manipulator. Besides, the objective of the simulator is to examine the performance of the robot with the designed controller[9,10].

When a controller is designed on the basis of a new control algorithm, the simulator presents how the robot performs under the application of the control input generated from the new controller. From the simulation results, a control designer compares the performance of the new controller with that of a conventional controller. The comparision is an essential procedure for the development process of the new controller.

The problem occurring during the procedure of comparison is that the dynamic model is formulated from the manipulator dynamics only[11]. Consequently, the control input generated from the controller is a joint torque or force acting at the joint of the manipulator. Therefore, the simulation results reflect the relationship between the manipulator dynamics and the control input ignoring the actuator dynamics. As indicated in reference[12,13], the joint torque or force cannot be directly applied to the robot as a control input. It must be converted into an electrical input by the actuator. Thus, when the performance of the robot is investigated through the simulator, the actuator dynamics has to be included in the dynamic model[14].

This paper places the emphasis on the following subjects: developing an integrated dynamic model and developing a hardware oriented control scheme which is usually implemented in an industrial robot. The integrated dynamic model is established by combining the manipulator dynamics and the DC servo motor dynamics. In addition, the dynamic model includes viscous friction caused by the joint speed and the payload which the robot carries. Since the motor dynamics is included in the dynamic model, the control input becomes the supplied voltage to the motor. The voltage is attained after the conventional PI position, speed, and current controllers are sequentially executed. It is modulated by a PWM generator and is furnished to the motor. Based on the supplied voltage, the simulator computes the torque generated in the motor with the motor dynamics and finds the torque or force acting at the joint of the manipulator with the connecting mechanism between the manipulator joint and the motor. Finally, the simulator finds the dynamic response of the robot manipulator under the application of the joint torque or force through the mathematical integration of the manipulator dynamics. These procedures are conducted in the industrial robot system and they are mathematically modeled in the simulator to be consistent with those of the actual system.

Beside the integrated dynamic model and the control scheme, the simulator includes a trajectory generator which yields velocity and position profiles at each joint for the desired motion plan. Kinematics is also involved to analyze various motions of the robot.

II. Manipulator and Actuator

1. Kinematics

As shown in Fig.1, the SCARA is an open-loop type manipulator composed of A, B, Z, and W axes. The A and B axes are actuated by the DC servo motors through the harmonic drives whose backlashes do not almost exist. These axes yield the horizontal motions of the robot arm within the workspace. The Z axis is moved up and down by the ball screw which converts the rotational motion of the motor into the translational motion of the axis. This axis generates the vertical motion of the robot end-effector. Finally, the W axis is driven by the DC servo motor through a gear box. This axis produces the rotation of the end effector. Table 1 presents the specification of the SCARA robot which has been developed in GSIS.

When link coordinates are established by Denavit-Hartenberg representation, the forward

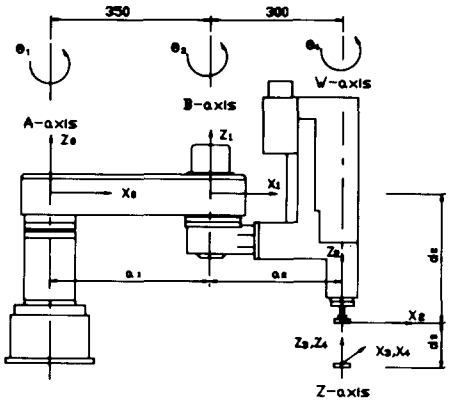


Fig.1. Coordinate Frames assigned to SCARA manipulator.

Table 1. Specification of a SCARA robot

Item	Spec.		
Operational angle & Maximum speed	Operational angle		Maximum speed
	A axis	±120°	200°/sec
	B axis	±140°	250°/sec
	Z axis	150mm	500mm/sec
	W axis	±300°	450°/sec
Arm length	First arm		350mm
	Second arm		300mm
Maximum compounded speed	3.6m/sec		
Repeatability	±0.02mm		
Payload	Max. 4 kg$_f$		

kinematics describes the motion of the manipulator[15]. It is the expression of the end-effector frame related to the base frame;

$$\mathbf{A}_4^0 = \mathbf{A}_1^0\ \mathbf{A}_2^1\ \mathbf{A}_3^2\ \mathbf{A}_4^3 = \left(\begin{array}{ccc|c} n_X & o_X & a_X & P_X \\ n_Y & o_Y & a_Y & P_Y \\ n_Z & o_Z & a_Z & P_Z \\ \hline 0 & 0 & 0 & 1 \end{array}\right) = \left(\begin{array}{c|c} \mathbf{R}_4^0 & \mathbf{P} \\ \hline \mathbf{0} & 1 \end{array}\right) \quad (1)$$

where $\mathbf{R}_4^0$ and P represent the orientation and the position of the end-effector, respectively. Those are

$$\mathbf{R}_4^0 = \begin{pmatrix} -\sin(\theta_1+\theta_2+\theta_4) & -\cos(\theta_1+\theta_2+\theta_4) & 0 \\ \cos(\theta_1+\theta_2+\theta_4) & -\sin(\theta_1+\theta_2+\theta_4) & 0 \\ 0 & 0 & 1 \end{pmatrix} \text{and}$$

$$\mathbf{P} = \begin{pmatrix} a_1\cos\theta_1 + a_2\cos(\theta_1+\theta_2) \\ a_1\sin\theta_1 + a_2\sin(\theta_1+\theta_2) \\ -d_2\ -d_3 \end{pmatrix}$$

Based on this analytical expression, the inverse kinematics is solved by the closed form solutions. In order to simplify the expression, we define the following variables:

$$r^2 \triangleq p_X^2 + p_Y^2 \text{ and}$$
$$b \triangleq (r^2 - a_2^2 - a_1^2)/2a_1a_2$$

The displacement of each joint is now expressed by the orientation and the position of the end-effector.

$$\theta_2 = \tan^{-1}[\pm\sqrt{1-b}/b] \quad (2)$$

$$\theta_1 = \tan^{-1}[p_X/p_Y] + \tan^{-1}[a_1\sin\theta_2/(a_1\cos\theta_2 + a_2)] - \theta_2 \quad (3)$$

$$d_3 = -d_2 - p_Z \quad (4)$$

$$\theta_4 = \tan^{-1}(n_X/o_X) - \theta_1 - \theta_2 \quad (5)$$

This inverse kinematics is implemented in a main controller to compute the set of the joint positions for various motions of a robot hand.

2. *Manipulator dyamics*

The dynamic behavior of the robot manipulator is described by the differential equations. These equations relate the joint forces or torques exerted by the actuators to the manipulator motions. Largrange formulation is here used to derive the dynamic equations of the motion[2]. The formulation represents the dynamics of the manipulator based on the kinetic and potential energy of the manipulator system;

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{q}_i} - \frac{\partial L}{\partial q_i} = \tau_i \ (i=1, 2, \cdots, n) \quad (6)$$

where

q_i = generalized coordinate representing the displacement of joint i,
L = Kinetic energy-Potential energy, and
τ_i = generalized force acting at the joint i.

When the formulation is applied to the SCARA manipulator, it yields inertia, gravity, Coriolis, and centrifugal forces for four degrees of freedom. With these terms, the joint torques or forces are described as

$$\tau_i = \sum_{j=1}^{4} D_{ij}(\theta)\ddot{\theta}_j + \sum_{j=1}^{4}\sum_{k=1}^{4} H_{ijk}(\theta)\dot{\theta}_j\dot{\theta}_k + G_i(\theta) \quad (i=1, 2, 3, 4) \tag{7}$$

$D_{ij}(\theta)$ is a 4x4 inertia positive symmetric matrix ($D_{ij}=D_{ji}$). While the element $D_{ii}(\theta)$ is an effective inertia at joint i, the element $D_{ij}(\theta)$ is the coupling inertia between joint i and j. $H_{ijk}(\theta)$ represents the Coriolis or the centrifugal forces combining the joint velocities: $H_{iij}(\theta)\dot{\theta}_j^2$ is the centrifugal force acting at joint i due to the velocity at joint j and $H_{ijk}(\theta)\dot{\theta}_j\dot{\theta}_k$ is the Coriolis force acting at joint i due to the velocities at joints j and k. Finally, $G_i(\theta)$ represents the gravity force at joint i. For the efficient dynamic computation in the simulation package, each element of the terms is expressed as a symbolic closed form equation.

3. *Actuator Dynamics*

The actuator is a DC servo motor which is usually used in an industrial robot. As shown in Fig.2, the actuator-gear-link assembly can be split into three major parts. They are electrical, electromechanical, and mechanical parts[12]. The dynamic equations of these three parts are as follows:

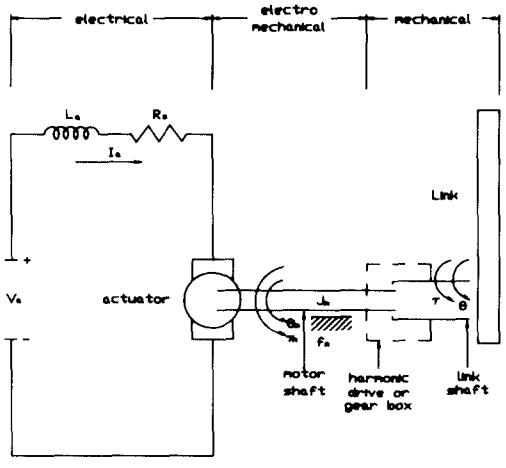


Fig.2. Actuator-gear-link assembly.

1) Electrical Part

By applying the Kirchhoff's voltage law to the armature circuit, we obtain

$$V_{ai}(t) = R_{ai}\, I_{ai}(t) + L_{ai}\, dI_{ai}(t)/dt + V_{bi}(t) \tag{8}$$

where V_{ai} = voltage in volt,
R_{ai} = resistnace in Ohm,
I_{ai} = current in ampere,
L_{ai} = inductance in henry, and

$V_{bi}(t)$ is the back electromotive force (emf) in the armature winding which can be represented by

$$V_{bi}(t) = K_{bi}\,\dot{\theta}_{mi}(t) \tag{9}$$

where K_{bi} = voltage constant and

$\dot{\theta}_{mi}(t)$ = angular velocity of the motor shaft.

2) Electromechanical Part

The DC motor is operated in its linear range so that the generated torque is proportional to the armature current;

$$\tau_{mi} = K_{ti}\, I_{ai}(t) \tag{10}$$

where τ_{mi} = motor torque and
K_{ti} = motor torque constant.

3) Mechanical Part

The motor shaft is mechanically connected to the actuator-gear-link assembly. As shown in Fig. 2, the motor torque τ_{mi} must be equal to the torque required to overcome the inertia, the viscous friction, and the load torque referred to the motor shaft. It is described for each joint.

$$\tau_{mi} = J_{mi}\,\ddot{\theta}_{mi} + f_{mi}\,\dot{\theta}_{mi} + \tau_i / n_i \quad (i=1, 2, 3, 4) \tag{11}$$

where J_{mi} = motor inertia, and
f_{mi} = motor viscous friction coefficient.

For the revolute joints i=1,2, and 4, τ_i is a load torque referred to the link shaft and n_i is gear ratio between the motor shaft and the link shaft, θ_{mi}/θ_i. For the prismatic joint i=3, τ_3 is a load force acting at the link shaft and n_3 is the transfer

ratio between translational distance and rotational angle, $2\pi\ \theta m_3/d_3$.

4. Integrated Dynamic Model

For the integrated dynamic model, the actuator dynamics is combined with the manipulator dynamics. The load torque or force acting at the link shaft τ_i is obtained from Eq. (7). It is converted into torque τ_{mi} acting at the motor shaft through the gear ratio as shown in Eq. (11). The conversion is accomplished by the harmonic drives, the gear box, and the ball screw. The efficiency of their conversions is considered in the dynamic model.

If τ_i is replaced by Eq. (7), the torque acting at the motor shaft is expressed as

$$\tau_{mi} = Jmi\ \ddot{\theta}mi + fmi\ \dot{\theta}mi + \{\sum_{j=1}^{4} Dij(\theta)\ddot{\theta}j + \sum_{j=1}^{4}\sum_{K=1}^{4} Hijk(\theta)\dot{\theta}j\dot{\theta}k + Gi(\theta)\}/n_i \qquad (12)$$

To simplify the expression, the following vector and matrix are defined:

An effective inertia matrix $De \in \overset{4\times4}{R}$ is

$$D_{eij}(\theta) \triangleq Diag[Jm_1,\ Jm_2,\ Jm_3,\ Jm_4] + D_{ij}(\theta)/n_i\ n_j$$

An friction coefficient matrix $F_m \in \overset{4\times4}{R}$ is

$$F_m \triangleq Diag[fm_1,\ fm_2,\ fm_3,\ fm_4]$$

A gravity, Coriolis and centrifugal vector $T_d \in \overset{4\times1}{R}$ is

$$T_{di} \triangleq \{\sum_{j=1}^{4}\sum_{K=1}^{4} Hijk(\theta)\ \dot{\theta}j\dot{\theta}k + Gi(\theta)\}/ni$$

With the defined vector and matrix, the motor torque $\tau_m=[\tau_{m1},\ \tau_{m2},\ \tau_{m3},\ \tau_{m4}]^T$ is rewritten with the angular displacement vector of motor $\theta_m = [\theta_{m1},\ \theta_{m2}, \theta_{m3}, \theta_{m4}]^T$.

$$\tau_m = De\ \ddot{\theta}m + Fm\ \dot{\theta}m + Td \qquad (13)$$

The motor torque τ_m is generated by torque constants $Kt = Diag\ [Kt_1, Kt_2, Kt_3, Kt_4]$ and armature currents $Ia = [Ia_1, Ia_2, Ia_3, Ia_4]^T$. That is

$$Kt\ Ia = De(\theta)\ddot{\theta}m + Fm\ \dot{\theta}m + Td \qquad (14)$$

The current Ia is modeled in the electrical part of the actuator dynamics. By combining Eqs. (8) and (9), the current Ia is expressed as

$$\dot{I}a = La^{-1}\{-Ra\ Ia - Kv\ \dot{\theta}m + Va\} \qquad (15)$$

where

$La = Diag\ [La_1,\ La_2,\ La_3,\ La_4]$,

$Ra = Diag\ [La_1,\ La_2,\ Ra_3,\ Ra_4]$,

$Kv = Diag\ [Kv_1,\ Kv_2,\ Kv_3,\ Kv_4]$, and

$Va = Diag\ [Va_1,\ Va_2,\ Va_3,\ Va_4]$,

The above Eqs. (14) and (15) represent the dynamic model in the actuator space combining the actuator dynamics and the manipulator dynamics.

III. Control Scheme

The control scheme of an industrial robot consists of three major parts. They are main controller, axis controller, and servo driver as shown in Fig.3. The detailed control scheme of the SCARA robot is presented in the reference[16] This section will describe the control logic flow from the main controller to the servo driver.

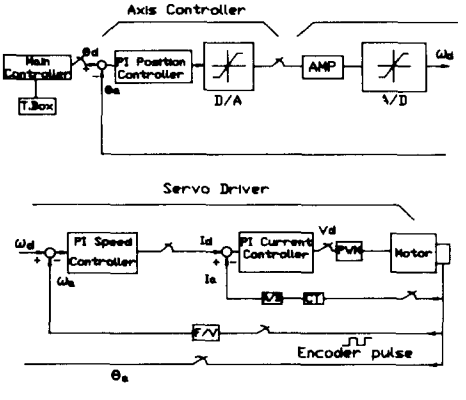


Fig.3. Conventional control scheme of industrial robot.

1. Main Controller

A main controller takes charge of a teach mode, a program mode, and a play back mode. In the teach mode, the main controller creates the data defining the robot motion which is taught by an operator through a teaching box. A job program is built in the program mode. It selects the teach program and defines the task of the robot. In the play back mode, the main controller interprets the programmed robot motion and generates a trajectroy planning based on the taught position and

the defined path. If the defined path is not a point to point (PTP) motion but a linear or a circular motion, the main controller executes an inverse kinematics to find the corresponding joint positions.

For a smooth motion of the robot, a velocity curve $\dot{\theta}d(t)$ is generated in the third order form. This velocity curve is integrated to produce the desired position of each joint;

$$\theta d(t) = \int \dot{\theta} d(t) dt \tag{16}$$

The joint position is converted into the desired motor shaft position through the gear reduction. With the motor shaft position, the controller generates the number of encoder pulses according to the resolution of the encoder. These encoder pulses become the desired position command and are sent to the axis controller.

2. Axis Controller

The purpose of the axis controller is to position the motor shaft to the desired position. In the process of the trajectory generation, the main controller must execute the inverse kinematics and generate the high order position profiles in a real-time. Since this process requires plenty of computational time, the desired position command is generated every 16 msec. This sampling time is not fast enough to do a position control. Therefore, the axis controller does a linear interpolation and finds ten intermediate points for each desired position command coming from the main controller. Consequently, the intermediate points become the desired position command and they are updated every 1.6 msec in the position control.

An encoder is attached to each motor shaft detecting the actual position of the motor shaft. The actual position is transformed to the corresponding number of pulses by the encoder circuit and is sent back to the axis controller. Then, the axis controller finds the position error between the desired and the actual positions. This position error goes through a PI position controller to generate the desired velocity command;

$$\omega d = Kp(\theta d - \theta) + Ki \int (\theta d - \theta) dt \tag{17}$$

where Kp and Ki are proportional and integral gains, respectively. In order to communicate with a servo driver, the digital velocity command ω_d is converted into the analog command by a D/A converter. The analog command is scaled such that $\pm$ Vmax corresponds to the maximum motor speed in both CW and CCW.

3. Servo Driver

The servo driver is composed of a speed controller, a current controller, and a PWM generator. The velocity command is sent to the servo driver and is converted into the digital signal through an A/D converter. The driver senses the motor speed by counting the number of the encoder pulses during a fixed time interval. The PI speed controller computes the desired current input I_d based on the desired and actual velocities as the PI position controller does. The I_d is now sent to the current controller which generates the desired voltage V_d in the same way as the previous controllers. In order to reduce a speed ripple, the speed of the current loop is five times faster than that of the speed loop. Finally, this desired voltage is sent to the PWM generator which modulates the pulse width[17]. The pulse width is determined by comparing the desired voltage with the reference triangular wave as shown in Fig.4. Thus, the current controller with the PWM acts as a current controlled voltage source inverter.

The PWM generator finds crossing points between the desired voltage and the reference triangular wave. These crossing points determine on and off time intervals for A and B transistors and complement $\bar{A}$ and $\bar{B}$ transistors. In practice, the on and off time intervals are adjusted by dead and storage time of the transistors. According to the adjusted time intervals, the PWM circuit switches either on or off the power output transistors which connect the motor in H-bridge configuration. Thus, the DCV is supplied to the motor during T_2 and T_4 but is cut off during T_1, T_3, and T_5. These time intervals play the role of the input for the motor controlled.

IV. Development of Simulator

A simulator is used to demonstrate how a robot will move under the application of the control input. The control input can be obtained from various control algorithms. Here, the simulator employs a conventional PI control algorithm as an industrial robot does. Since the simulator

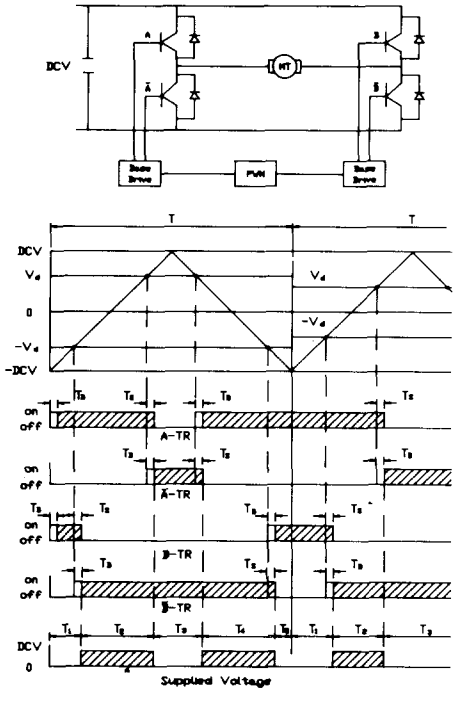


Fig.4. Pulse modulation from PWM generator.

includes the actuator dynamics, the control input is the supplied voltage and the on and off time intervals for each motor.

In order to conduct the simulation sequentially, the simulator assigns Fine Time Intervals (FTI) organizing all the time intervals of motrs as shown in Fig.5. In general, 4n+1 FTIs will be assigned for n-motors. In the case of the SCARA, 17-FTIs are required for each sampling time of the current loop. The simulation is sequentially conducted from the first FTI to the last FTI with the supplied voltages.

The integrated dynamic model of the simulator is formulated with all the parameters used in the SCARA robot which has been developed in GSIS. Tables 2 and 3 show parameters for a link and parameters for an acuator, respectively. The parameters for a link are measured from the coordinates assigned to the links. The parameters for an actuator are obtained from the specification of the DC servo motors used in the SCARA robot.

the simulator is constructed from the dynamics of the manipulator and the actuator. From Eq. (15), the current is updated by integrating the

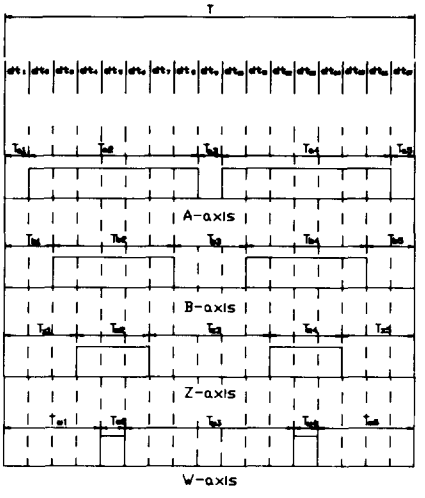


Fig.5. Sequential fine time intervals.

Table 2. Parameters for each link

Parameter	A-link	B-link	Z-link	W-link
mass(kg)	16.81	8.71	0.3	0.97
radius of gyration(m)	0.290	0.201	0.126	0.009
mass center (m)	0.203	0.162	0.025	0.075

Table 3. Parameters for each actuator (DC servo motor)

Parameter	A axis	B axis	Z axis	W axis
Rated Output(W)	200	120	100	60
Rated speed(rpm)	3000	3000	3000	3000
Rated torque(kg_f-cm)	6.5	3.9	3.25	1.9
Torque constant, Kt (kg_f-cm/A)	1.21	0.857	0.828	0.59
Voltage constant, kb (v/krmp)	12.4	8.8	8.5	6.08
Armature resistance, Ra (ohm)	0.4	0.58	0.92	1.07
Armature inductance, La (mH)	0.6	0.7	0.9	0.39
Gear ratio, ni	1 : 100	1 : 80	1 : 785.4	1 : 13.3
Weight(kgf)	2.0	1.8	0.98	0.45
Encoder resolution (pulses/rev)	1000	1000	500	500

dynamic equations of the actuator. That is,

$$d\,Ia/dt = -Ra/La\ Ia - Kv/La\ \dot{\theta}m + Va/La \tag{18}$$

where dt is the FTI which the PWM generator has determined and Va is the supplied voltage DCV.

The position and the velocity are updated by integrating the dynamic equations of motion of manipulator. The combined dynamic model is reformed and the acceleration becomes a function of the motor torque;

$$\ddot{\theta}m = De(\theta)^{-1}(\tau_m - Td - Fe\dot{\theta}m) \tag{19}$$

where τ_m is replaced by the updated current in Eq. (18) and the torque constant, $\tau_m = Kt\ Ia$. The $De(\theta)^{-1}$ always exists since it is a symmetric and positive definite matrix. Once the acceleration is obtained, state equations are developed and are integrated to find the velocity and the position. They are.

$$d\,\dot{\theta}m/dt = De(\theta)^{-1}|Kt\ Ia - Td - Fe\dot{\theta}m| \tag{20}$$

$$d\,\theta m/dt = \dot{\theta}m \tag{21}$$

We define $\mathbf{Y} \in R^{12\times 1}$ vector as follows;

$$\mathbf{Y} = |Y_1,\ Y_2,\ Y_3|^t = |Ia,\ \theta m,\ \dot{\theta}m|^t$$

From Eqs. (18), (20), and (21), the integrated dynamic model becomes

$$\mathbf{Y} = \begin{pmatrix} \dot{Y}_1 \\ \dot{Y}_2 \\ \dot{Y}_3 \end{pmatrix} = \begin{vmatrix} (Va - Kv\ Y_3 - R_a\ Y_1)/La \\ Y_2 \\ De(Y_2/N)^{-1}|Kt\ Y_1 - Fe\ Y_3 - Td| \end{vmatrix} = F(Y,\ Va) \tag{22}$$

Given the current state Y(k) and the input voltage Va(k), the next state Y(k+1) is solved by the numerical integration of the state equation over the sampling time dt. The integration technique used here is the fourth order Runge-Kutta method[18]. As shown in Fig. 6, the position, the velocity, and the current are updated on the basis of the corresponding sampling time of their loops.

V. Simulation Results

The simulation is conducted on VAX/VMS 11-750 computer to investigate the characteristics and performance of the SCARA. Th investigation is performed with changing various actuators,

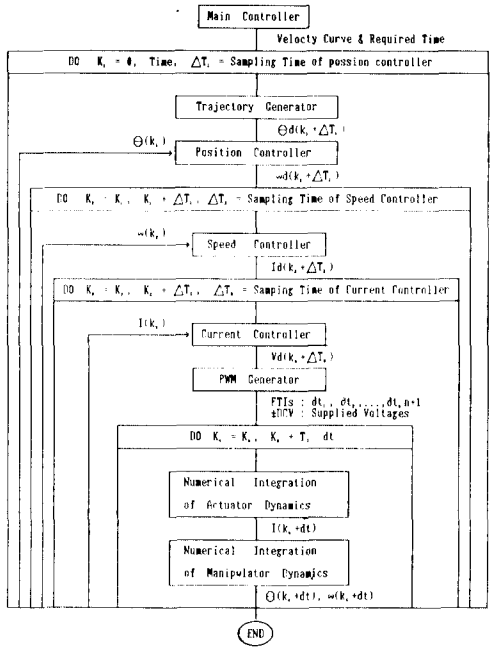


Fig.6. Flowchart of computer simulation.

trajectory plans, control gains, etc. Some of the simulation results are here presented.

Fig. 7 shows the required torques at the A-axis motor when the robot moves full range for each joint with the maximum payload and speed. This is the case in which the maximum torques are acting at each motor. The torque profile shown in the graph provides criteria to select the appropriate motor and to design the manipulator. Once the motor is selected and the manipulator is designed, the simulation is conducted to determine the control gains and the trajectory plans. Fig.8 shows the actual velocity and position profiles at the A-axis joint when the robot moves from 0° to 100° at the axis. Using these profiles, we can determine the velocity slope of the trajectory plan reflecting the capacity of the motor and the manipulator dynamics. The control gains are also selected from the response of the robot. Table 4 shows the sampling time and the control gains used in the position loop, the speed loop, and the current loop. These gains are selected on the basis of the simulation results and are implemented in the SCARA robot controller.

Fig. 9 and 10 represent degrees of difference

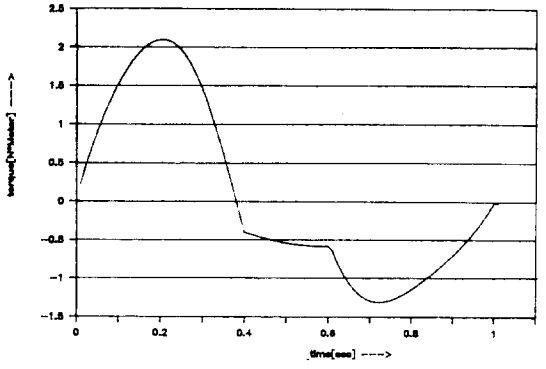


Fig.7. Required torque at A-axis motor.

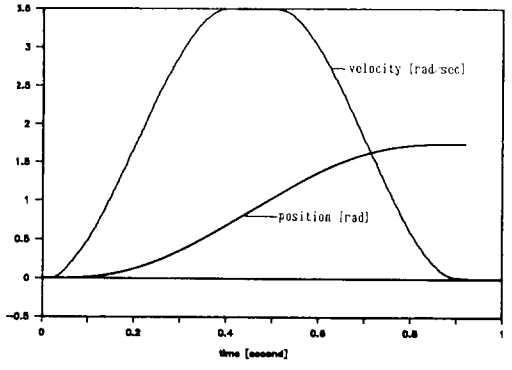


Fig.8. **Actual** velocity and position profiles at A-axis joint.

Table 4. Sampling times and control gains of posiion, speed, and current loops.

item		position	speed	current
Sampling time (msec.)		$\triangle T_1 = 1.6$	$\triangle T_2 = 1.0$	$\triangle T_3 = 0.2$
A-axis	Kp	1	9984	1536
	Ki	0	64	128
B-axis	Kp	1	9984	1536
	Ki	0	48	128
Z-axis	Kp	1	5588	1536
	Ki	0	48	128
W-axis	Kp	1	1792	928
	Ki	0	48	112

between the actual robot and the simulator in the position and the velocity, respectively. As shown in the graphs, the degrees of difference are small at the low speed but are large at the high speed since the dynamic model strongly affects the simulation results at the high speed. In practice, the dynamic model of the simulator is not perfectly matched to the actual system due to the existence of uncertainty. However, the degree of difference is relatively small enough to examine the performance of the robot. Finally, Fig. 11 demonstrates the error occurred at the end of the robot arm. The SCARA has small position error at the final position, which is less than 0.05 mm. In fact, the repeatability of the SCARA, which has been developed in GSIS , is close to 0.01 mm without any payload.

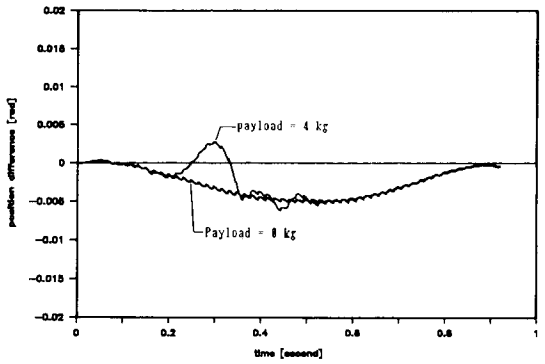


Fig.9. Position differece between a robot and a simulator at A-axis.

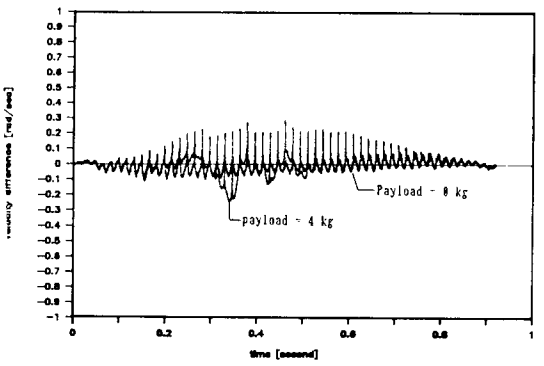


Fig.10. Velocity difference between a robot and a simulator at A-axis.

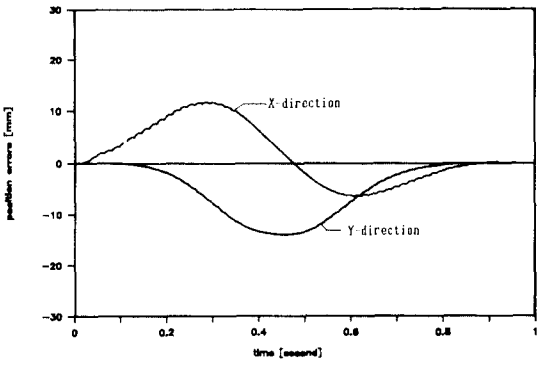


Fig.11. Position errors at robot end-effector in X- and Y-directions.

VI. Conclusions

A fully integrated simulator is made for the development of an industrial robot. In the simulator, a dynamic model is established by combining the dynamics of the manipulator and the DC motor. As the dynamic model includes the mechanical parameters as well as the electrical parameters, it is possible to analyze the entire dynamics of the robot system. In practice, this dynamic model is applied to the SCARA robot and computes the torque acting at each motor. From the computed torque, peak and rated torques are found for the selection of an appropriate motor. In addition, the dynamic model determines what trajectory planning can be applied to the robot considering the characteristics of the manipulator and the motor.

The control scheme of the simulator is composed of a main controller, an axis controller, and a servo driver including a PWM generator. Thus, the simulator is distinguished from the others in that it is consistent with an industrial robot. The simulation is conducted with changing various design parameters and shows the performance of the robot. It plays an important role in selecting the parameters. The simulator is practically applied to the SCARA robot with the chosen motors, control gains, and sampling time and demonstrates the performance of the robot. The simulation results tend to be close to those of the actual system but they are not perfectly matched. This problem is cased by both simplification of some parts and existence of uncertainty in the dynamic model. Nonetheless, the simulation results are quite enough to provide the design concept in the development process of the industrial robot.

References

[1] Hollerback, J.M., "A recursive lagrangian formulation of manipulator dynamics and a comparative study of dynamics formulation complexity," *IEEE Trans. System, Man, and Cybernetics,* vol. 10, no. 11, pp. 730-736, nov. 1980.

[2] Paul, R.P., Robot Manipulator: Mathematics, Programming, and Control, M.I.T. Press, 1981.

[3] Meghed, S. and Renaud, M., "Dynamics modeling of robot manipulator containing closed kinematic chain," *Proc. of an International Meeting on Advanced Software in Robotics,* North-Hollend, Belgium, 1983.

[4] Williams, R.J. and Seireg, A., "Interactive modeling and analysis of open or closed loop dynamic systems with redundant actuators," *ASME Journal of Mechanical Design,* vol. 101, no. 3, pp. 407-416, July 1979.

[5] Stepanenko, Y. and Vukobratovic, M., "Dynamics of articulated open-chain active mechanisms," Mathematical Biosciences, vol. 28, no. 1/2 pp. 137-170, Feb. 1976.

[6] Orin, D.E., Mcghee, R.B., Vukobratovic, M., and Hartoch, G., "Kinematic and kinetic analysis of open-chain linkage utilizing Netwton-Euler methods," Mathematical Biosciences, vol. 43, no. 1/2, pp. 107-130, Feb. 1979.

[7] Luh, J.Y.S., Walker, M.W., and Paul, R.P.C., "On-line computational scheme for mechanical manipulators," *ASME Journal of Dynamic Systems, Measurement, and Control,* vol. 102, pp. 468-474, June 1980.

[8] Luh, J.Y.S. and Zheng, Y.F., "Computation of input generalized forces for robots with closed kinematic chain mechanisms," *IEEE Journal of Robotics and Automation,* vol. 2, pp. 95-103, June 1985.

[9] Walker, M.W. and Orin, D.E., "Efficient dynamic computer simulation of robotic

mechanisms," *ASME Journal of Dynamic System, Measurement, and Control,* vol. 104, pp. 205-211, Sept. 1982.

[10] Kankanranta, R.K. and Koivo, H.N., "Dynamics and simulation of compliant motion of a manipulator," *IEEE Robotics and Automation,* vol. 4, no. 2, pp. 163-173, April 1988.

[11] Lee, C.S.G. and Chung, M.J., "An adaptive control strategy for mechanical manipulators," *IEEE Automatic Control,* vol. AC-29, no. 9, pp. 837-840, Sept. 1984.

[12] Luh, J.Y.S., "An anatomy of industrial robots and their control," *IEEE Automatic control,* vol. AC-28, no. 2, pp. 133-153, Feb. 1983.

[13] Luh, J.Y.S., "Conventional controller design for industrial robots -A tutorial," *IEEE Trans. System, Man, and Cybernetics,* vol. SMC-13, no. 3, pp. 298-316, May/June 1983.

[14] Lee, M.K., Lee, G.N., and Lim, K.Y. "Development of a fully integrated simulation package for industrial robot," *Proc. of '88 Korean Automatic Control Conference,* pp. 1028-1032, Oct. 1988.

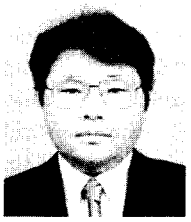
[15] Craig, J.J., Introduction to Robotics: Mechanics and Control, Addison-Wesley Publishing Co., 1986.

[16] 정강익, "산업용 robot의 설계개요", pp. 65-73, 1988년 5월.

[17] Muir, P.F. and Neuman, C.P., "Pulsewidth modulation control of brushless DC motors for robotic applications," *IEEE Trans. Industrial Electronics,* vol. IE-32, no. 3, pp. 222-229, Aug. 1985.

[18] James, M.L., Smith, G.M., and Wolford, J.C., Applied Numerical Methods for Digital Computation, Happer & Row, Publishers, Inc., 1985.

著 者 紹 介



李 敏 基(正會員)

1955년 6월 17일생. 1982년 2월 인하대 공대 기계공학과 졸업. 1985년 미국 West Virginia Univ. 기계공학과 공학석사학위 취득. 1988년 미국 West Virginia Univ. 기계공학과 공학박사 학위 취득. 현재 금성산전(주)연구소 선임연구원. 주관심분야는 자동제어 이론 및 로보틱스 임

李 光 男(正會員)

1961년 5월 23일생. 1984년 2월 서울대 공대 기계설계학과 졸업, 1984년~1987년 Kia 산업, 중앙기술연구소 연구원, 現在 금성산전(주) 연구소, 주임연구원, 주관심분야는 로보틱스.

林 桂 榮(正會員)

1953년 5월 3일생. 1975년 서울대공대 전기공학과 졸업. 1978~1981년 국방과학연구소 연구원. 1985년 미국 State Univ. of New York at Story Brook 전기공학과 졸업(공박). 1987년 미국 New York Institute of Technology, Visiting Assitant Professor. 현재 금성산전(주)연구소 책임 연구원.