Dynamic Sensitivity Evaluation of Retrofit SR-600 SCARA Robot

Edeh O.D.F., Ossia C.V.*

Department of Mechanical Engineering, University of Port Harcourt, Port Harcourt, Nigeria

*Corresponding author: chinwuba.ossia@uniport.edu.ng

Received December 12, 2018; Revised January 19, 2019; Accepted February 13, 2019

Abstract Rapid technological advances to meet increasing demand for optimal, economical and environmentally friendly solutions has lead to the challenge of increased discontinuities of some service parts by manufacturers. These discontinuities had resulted in increased abandonment of some machines after a few years. Electronic and control modules often receive great share of this abandonment. Hence, the Selective Compliant Assembly Robot Arm (SR-600 SCARA Robot) had been cut-up by these discontinuities. Retrofitting enables existing machines access to new technologies and opportunities. In this study, SR-600 SCARA Robot was reviewed for this purpose. The forward and inverse kinematics equations were solved. An insulated-gate bipolar transistor (IGBT)-based power amplifier design was developed. Results of the amplifier simulation, closed-loop control implementation and the PID (proportional integral derivative) tuning of the axes servos were presented. The results does not only offer cheap, compact and power efficient control solution, but also an improved multimode motion control using PID controls with velocity and acceleration feed-forward and high speed Ethernet communication between the Robot and the host computer. In this retrofit, results indicate an excellent angular resolution of 0.001° for the three revolute joints, a translational resolution of 2.223µm and an overall motion accuracy of 0.038° for the PID tune configuration of \( K_i = 1, K_p = 50, \) and \( K_d = 100 \).

Keywords: retrofit, SCARA, IGBT, proportional integral derivative (PID)

Cite This Article: Edeh O.D.F., and Ossia C.V., “Dynamic Sensitivity Evaluation of Retrofit SR-600 SCARA Robot.” Journal of Automation and Control, vol. 7, no. 1 (2019): 7-14. doi: 10.12691/automation-7-1-2.

1. Introduction

In manufacturing systems, retrofitting is a common practice because companies cannot afford to scrap their machines, after just a few years of installation, due to lack of service parts from the original equipment manufacturers (OEM) who has moved on with further technological advances. Besides, the persistent demand for an optimal and economical system that comply with newer accessibility standards, that save energy and cost, and that improve the overall system response boost the technological quests and advances leading to rapid obsolesce of some system parts. Most often, electronic control devices have the greatest share of this rapid obsolesce due to rapid rate of advances in IC technologies. To this end, Retrofit becomes the answer, to provide the existing machine with the state-of-the-art solution giving them access to new technologies and opportunities.

The SR-600 SCARA Robot (Figure A1) from Robert Bosch is a good example of such a system that has suffered the abandonment due to obsolesce and discontinuity of the product service parts.

Since the introduction of Robots in the 10th century [1] and the PUMA by Unimation in 1976, the technology of Robotics has attracted enormous interest around the globe. Research interest in articulated Robots, SCARA Robots, Gantry Robots, and presently the Unmanned Aerial and Underwater Robots has continued to grow exponentially both in finding out new types of Robots and new ways to improve manufacturing and service systems.

Problems of position/force control of Robots have been one major issue arising due to the controller deficiencies in flexibility, processing speed and adaptability. In the past, various technologies have been employed to offer solutions to these problems in relation to robot motion.

With Pentium MMX processor and intelligent power booster, IQ-200 controller with expanded capability in parallel processing was adapted to control Bosch Swivel-Arm Robots. This offered the possibility of several data transfer and exchange with other computer vision systems or similar sensor systems. CL-500 PLC controller provides high speed and long duration recording capabilities. ARM and FPGA controllers offered high-speed and reconfigurable capabilities and less peripheral equipment, higher integration and flexibility. However, these were neither dedicated controllers for the RS-600 Robot nor optimally adapted to it.

For over 44 years since its invention [2], SCARA Robots have remained the Robot of choice for high throughput applications. Despite numerous other robot configurations in the market today, SCARAs are still the fastest robots for assembly applications [3,4]. A lot of
impressive scholarly works have been done in this 4-Degree of Freedom (4-DOF) Robot and in the retrofit of some industrial machines.

Aroca et al [5,6] explored Real Time Linux and other useful free open source tools for Real Time systems applied to control of industrial SCARA Robots for optimal reliability, determinism and stability, and presented a bottom-up approach for the upgrading of SCARA Robot using Linux Kernel Modules (LKMs), Real Time Operating System (RTOS) and open Robot control software (Orocus); while Onozato and Maeda [7] described position control of SCARA Robots using two neural networks to learn its inverse kinematics and dynamics using simulation perturbation optimization method to update weight of the two neural networks. Further, Kucuk and Bingul [8] presented the closed form solutions of Robot inverse kinematics of 16 industrial robot manipulators; while Taylan and Canan [9] developed a complete mathematical model of SCARA Robot with their servo actuator dynamics.

A method for applying the technique of retrofitting was presented by Ribeiro et al [10] to increase the life of industrial equipment; while Becerra et al [11] devised the hardware retrofit and computer torque control of PUMA-560 Robot where a mechanically-sound Robot manipulator with their obsolete interfaces were revitalized at low cost using a modern and flexible software interface based on MATLAB and SIMULINK. Although Miskon [12] used DMC-2280 Galil motion controller for same PUMA-560, but little has been reported particularly on the hardware retrofit of SR-600 SCARA Robot. The controller and the power amplifiers were redesigned and implemented using newer technologies.

In this study, the Robot mechanical system equipped with joint servo motors and the associated optical encoders, limit switches, temperature sensors and pneumatic end-effector; the electrical unit and the associated power transformers of the original SR-600 Robot, were retained. The encoder parameters and joint travel limits were experimentally deduced and the SCARA forward and inverse kinematic were solved A new control algorithm using DMC-2260 (Galil Motion Controller) and highly efficient IGBT power amplifier was developed. Further, interfacing of the sensors, encoders, power amplifiers and the SR-600 Robot to the DMC-2260 motion controller were implemented. The I/O Interfaces module was connected, and the communication between the controller and the PC was established. Finally, the PID controllers for the axes were tuned for best performance.

2. Methodology

2.1. Motion Controller

The DMC based closed-loop control system of RS-600 is shown in Figure 1. The Servo Amplifier unit was implemented with an IGBT device, replacing the previous discrete H-bridge driver. The DMC-2260 and ICM-2900 also replaces the old controller boards. The DMC communicated with the Robot via the interface module.

The encoders, limit switches, and temperature sensors act as feedback devices to close the loop.

3-Phase AC Main

PM2.2. Kinematic Models

The 4-DOF RRPR Robot frame assignment is shown in Figure 2. Joint axes 1, 2, and 4 are revolute joints, while joint axis 3 is prismatic. Servo motors for joints 1 and 2 are M0300 and M0200 Mavilor DC Servo motors, respectively, while joints 3 and 4 are E588MG DC motors. Axis 3 servo drives the rack and pinion mechanism which enables it to travel back and forth. Each of the Servos was equipped with a multi-turn optical encoder with 4320 counts per encoder revolution.

2.2.1. Forward Kinematic

The forward kinematics for the SR-600, which link parameters are shown in Table 1, was solved for the position and orientation of the robot end-effector at any point in time. Moreover, the link lengths and link offsets were determined by precise measurements, while the joint travel limits were determined experimentally.
### Table 1. Link Parameter Table

| Link | $\alpha_i$ | $a_i$ | $d_i$ | $\theta_i$ |
|------|------------|-------|-------|------------|
| 1    | 0          | $L_1$ | $d_1$ | $\Theta_1$ |
| 2    | 0          | $L_2$ | 0     | $\Theta_2$ |
| 3    | $180^\circ$| 0     | $d_3$ | 0          |
| 4    | 0          | 0     | $d_4$ | $\Theta_4$ |

The homogeneous transformation that fully describes the position and orientation of the end-effector with respect to the fixed frame is:

$$
\mathbf{T}^0_4 = \begin{vmatrix}
C(\theta_1 + \theta_2) & S(\theta_1 + \theta_2) & 0 & L_2 * C(\theta_1 + \theta_2) + L_4 * C(\theta_4) \\
S(\theta_1 + \theta_2) & C(\theta_1 + \theta_2) & 0 & L_2 * S(\theta_1 + \theta_2) + L_4 * S(\theta_4) \\
0 & 0 & -1 & d_1 + d_3 - d_4 \\
0 & 0 & 0 & 1
\end{vmatrix}
$$

(1)

#### 2.2.2. Inverse Kinematic

To map robot position and orientation from the task space to the joint space, we used the analysis method which solves for the joint variables using configuration data. This was more rigorous and tedious than the forward kinematic computation and also much more expensive and time consuming in real time control of manipulators. The configurations of the robot end-effector, given as:

$$
\begin{align*}
\mathbf{q}^T &= \begin{bmatrix}
\eta_1 & \eta_2 & \eta_3 & \eta_4 \\
\check{r}_1 & \check{r}_2 & \check{r}_3 & \check{r}_4 \\
r_1 & r_2 & r_3 & r_4 \\
0 & 0 & 0 & 1
\end{bmatrix} \\
\end{align*}
$$

(2)

was equated to the homogeneous transformation (Equation 1) and solved for $d_3$, $\theta_2$, $\theta_1$, and $\theta_4$. The results are:

$$
d_3 = r_4 + d_4 - d_1
$$

(3)

#### 2.3. Power Amplifier Design

**2.3.1. PWM Generator Circuit**

The PWM circuit of Figure 3 makes use of three Op-Amps (U1A, U2A, and U3A). U1A and U2A generate the triangular wave which is compared with a command voltage at V3 of U3A. The Op-Amp, U3A hence produce the varying ON/OFF pulses that eventually controls the motor speed via the control logic block. The frequency of the PWM is given by equation (7).

$$
f_{\text{out}} = \frac{1}{4 * R_2 * C} \left( \frac{R_1}{R_3} \right)
$$

(7)

#### 2.3.2. Logic Control Circuit

This unit receives its inputs enable signal from the over-current sense unit, and the PWM signal from the PWM generator. Its outputs (HIN1, HIN2, LIN1, and LIN2) serve as inputs to the H-Bridge of the IGBT module.

![Figure 3. PWM Generator](image)

![Figure 4. Logic Control Unit](image)
2.3.3. Over-Current Sense Circuit

The controller can send enable/disable signal to this unit (Figure 5) which, in effect, turns the power amplifier ON or OFF. So if something goes wrong and an over-current is sensed, the over-current monitor generates a logic Low which turns OFF the power amplifier irrespective of the logic state of EN from the controller.

2.4. Robot Retrofit

The interconnection module was first connected to the motion controller via the 100-pin high density cable, then the motors to the power amplifier. With power supply switched ON, and the command voltage to the power amplifier set to zero, the PWM generator of the power amplifier was adjusted till the motor was stationary.

With the ground level of the amplifier at the same potential, the AMPENA of the ICM-2900 was connected to the amplifier enable input. This was used to enable and disable the amplifier based on the control algorithm.

The terminals of the optical incremental encoders were connected to the DB-15 male connectors. The encoder pin designation is shown in Table 2.

Table 2. DB-15-Encoder connections

| DB-15 Pin | 5 | 10 | 4 | 3 | 2 | 1 | 7 | 6 |
|-----------|---|----|---|---|---|---|---|---|
| Encoder Terminal | +Vcc | gnd | A+ | A- | B+ | B- | I+ | I- |

The channels each have a complementary pair. The lead signals (CHA, CHA-, CHB, CHB-, INDEX, INDEX-) were connected to designated terminals (+MAX, -MAX, +MBX, -MBX, +INX, -INX) of the ICM using the module screw terminals. With the controller powered and PC communication established, the response of the encoder was monitored on the Galil Tool platform and recorded, as axis A was turned back and forth.

To determine the number of pulses per encoder revolution, axes 1 and 2 were turned to the robot’s full reach and initialized to zero count. Axis 1 was then moved ±90°. The result indicate a total angular displacement of 174960 encoder counts which is equivalent to 0.0010288 degree of joint movement per encoder count, or 81 (349920/4320) encoder revolutions per 360°. Similar result was evaluated for axis 2.

2.4.1. Jumper Installation

The jumpers in the DMC main board were provided for the hardware configuration of the controller. Since only servo DC motors were used, and neither unconditional firmware update nor erasure of programs, arrays, variables and motion parameters in the EEPROM when powered up were needed, MRST, UPGRD, SMA-SMH jumpers were uninstalled. However, there was need for a +5V supply to opto-isolators, and power made available to motors immediately the system was turned ON, hence LSCOM, INCOM and MO jumpers were installed.

2.4.2. DIP Switch Configurations

Only hardware handshake was needed to operate the system at 19200bps. It was also intended that the controller should use the auto-negotiate function to set the Ethernet connection speed, hence HSHK, 19.2k, and ENET on the DIP Switch were turned ON.

2.4.3. Communication between Robot and Software

The RS-232 and the Ethernet were registered in the Galil Smart terminal and configured. The former was setup for a communication speed of 19200bps whereas the later extracted the IP address of the controller (DMC-2260 Rv-1.0-n-f) as 128.240.28.178. The Ethernet protocol was set to TCP. With the controller connected directly to the PC via Ethernet crossover cable, test command “TPA return” was ran and the controller returned 00000000, which was the current position of axis A, and indicates proper communication between the software and controller hardware.

2.5. Tuning the PID

The basic structure of a PID controller is shown in Figure 6. The purpose is to make the output variable of the process, y follow the desired value, r. The manipulated variable, u is changed at the command of the controller, C(s).

First, the Proportional element provides the system’s responsiveness and is characterized by a steady state error such that increase in gain (Kp) results in faster speed, but also increased noise sensitivity and worse stability. Second, the Integral element provides the system’s accuracy and is characterized by zero steady state error. Hence small integral time constant results in faster steady-state error removal but higher overall system overshoot and instability. Third, the Derivative element provides for the system’s stability or damping. When it’s time constant is too small, it will have no influence, but performance degrades as the time constant gets too large. The three elements, hence, take into consideration the present, the past and the future errors. The transfer function of the PID controller is given below:
\[ C(s) = K_p \left(1 + \frac{1}{T_1 s} + T_d s \right). \] (8)

Tuning of the PID is the assignment of values to the PID filter variables (K_p, K_i, and K_d) to achieve the desired optimal response of the system. The manual tuning method was adopted because it provides a means of testing the responsiveness of the system. A step response test was used to evaluate the response test of the system.

Figure 6. Basic PID for (a) Control Structure, (b) Model

Axis 4 was commanded to move from an initial 0° position to a 30° position within 3 seconds. This translated to a 29190 encoder count movements. It was also desired that the required velocity be reached within 1 second and back to zero velocity at the end of the run within 1 second. The PR (Position Relative – in counts), AC (Acceleration – in counts/sq-seconds), DC (Deceleration – in counts/sq-seconds) and SP (Speed - in counts/seconds) were set as follows:

3. Results and Discussion

3.1. Power Amplifier PWM Simulation Result

Simulation result of the PWM circuit with Command Voltage set at 0V is shown in Figure 7. From this response, the triangular wave which is almost symmetric about 0V indicates a peak-to-peak of 20.3V and Period, T of 8.7ms. As expected, the final output square wave of Op-Amp U3A is of 50% duty cycle. This output generates the required mark-space signals for the motor speed control.

3.2. Axis 1 Travel Limit Result

The limit of travel of axis 1 was determined and the results were shown in Figure 8. From analysis, the mean value of 226078 counts and standard deviation of 36 counts evaluates to the joint travel of ±116.3°. The result also reveal that the distribution is negatively skewed which suggests that it is more likely to have values slightly above the mean value.

Figure 7. PWM Simulation Result
3.3. Axis 2 Travel Limit Result

The limit of travel of axis 2 was also determined and the result shown in Figure 9. From analysis, the mean value of 246152 counts and standard deviation of 9 counts evaluates to the joint travel of ±126.6°. The result also reveal that the distribution is slightly negatively skewed which suggests that it is more likely to have values slightly above the mean value.

3.4. Axis 3 Travel Limit Result

The relationship between the rotary movements of axis driving motor and the linear displacement of the axis was then determined by regression analysis and the result was shown in Figure 10. From analysis and as expected, there was a linear relationship \( y = 2.161x - 0.269 \) between the angular displacement (in encoder counts) of the driving motor and the linear displacement (in mm) of the axis with a coefficient of determination, \( R^2 = 0.999 \). This evaluates to a translational resolution of 2.223µm. Other parameters as determined by measurement were \( L_1 = 320\text{mm}, \ L_2 = 280\text{mm}, \ d_1 = 5\text{mm}, \ d_4 = 30\text{mm}, \) and \( \theta_4 = ±360° \).

The sensitivity of the linear displacement with respect to angular displacement, shown as the slope of the regression was found to be above two (m = 2.161), and expectedly, the intercept of the linear regression is negative (-0.269) showing the translational inertia or the response error that must be overcome before Robot linear displacement occurs.

3.5. PID Tuning Results

The program (subsection 2.5 above) was repeatedly ran (four times each) for various combinations of \( K_i (0,1), \ K_c (1,5,50,500), \) and \( K_d (0 ,100, 300, 500, 1000, 3000) \), alternating the sign of the relative position move command to enable the axis move back and forth. The results, which were observed and recorded, are shown in Figure 11 to Figure 14.

A one-way ANOVA for Case 1 (constant \( K_i=K_d=0, \) and \( K_p = 1, 5, 500) ), Case 2 (constant \( K_p=0, K_c=50, \) and \( K_d = 100, 300, 500, 1000), \) Case 3 (constant \( K_c=50, K_d=100, \) and \( K_i = 0, 1), \) and Case 4 (constant \( K_i=1, K_p=50, \) and \( K_d = 100, 1000) \) were performed to compare the various effects of \( K_i, K_p \) and \( K_d \) on the position errors of the tuned axes.

For Case 1, Figure 11, the ANOVA result indicated no significant difference (\( F = 0.000, \ p\ -value = 0.996, \) and \( R^2 = 0.0008 \)). However, individual responses showed that there was less skewness for \( K_p = 5 \) (-0.009) and \( K_p = 500 \) (-0.818) than for \( K_p = 1 \) (1.719). The slight increased in the negative skewness of \( K_p = 500 \) could be due to increased noise sensitivity effect of higher \( K_p \); although \( K_p = 5, \) and \( K_p = 500 \) both showed over 50% p-Values (0.525 and 0.594).

For Case 2, Figure 12, the ANOVA result indicated low significance level (\( F = 0.15, \ p\ -value = 0.926, \) and \( R^2 = 0.0369 \)). However, individual responses showed that there was less skewness for \( K_p = 5 \) (-0.009) and \( K_p = 500 \) (-0.818) than for \( K_p = 1 \) (1.719). The slight increased in the negative skewness of \( K_p = 500 \) could be due to increased noise sensitivity effect of higher \( K_p \).
gave some stabilization effects. The p-value of 0.139 for $K_d = 500$ was comparatively satisfactory, however, the skewness of 0.12463 was not good compared to the former. With similar p-values (0.232 and 0.231, respectively) for $K_d = 100$ and $K_d = 1000$, the former was better with 0.02217 skewness against -1.20229 skewness for the latter.

For Case 3, Figure 13, the ANOVA result also indicated low significance level ($F = 0.09$, p-value = 0.773, and $R^2 = 0.0149$). However, individual response showed an overall excellent P-value of $p < 0.05$ (0.043) for $K_i = 1$, although with poor skewness (1.88877). On the other hand, for $K_i = 0$, the p-value and the skewness are 0.122 and -0.01424 respectively.

For Case 4, Figure 14, the ANOVA result also indicated low significance level ($F = 0.66$, p-value = 0.448, and $R^2 = 0.0988$). Individual performances showed that increased $K_d$ value ($K_d = 1000$) degraded the earlier performance ($K_i = 1$, $K_p = 50$, and $K_d = 100$) with p-value of 0.403 and skewness of 0.80738.

Similar performance analysis were performed but not reported for situations with constant $K_i = 0$, $K_p = 5$; and constant $K_i = 0$, $K_d = 100$, due to their less significance.

4. Conclusion

The quest for better systems, that will meet the increasing demand for economic, optimal and energy efficient, will never stop. Hence the challenge of keeping up with older machines with newer technologies will ever be in demand. Galil motion controller was found to be an efficient and effective solution in this regard and has settled this issue with SR-600 SCARA Robot. In this retrofit, our result indicate an excellent angular resolution of 0.001° for the three revolute joints, and a translational resolution of 2.223µm with an overall motion accuracy of 0.038° for the PID tune configuration of $K_i = 1$, $K_p = 50$, and $K_d = 100$.

References

[1] Hemal A. K, Menon M, Robotics in Genitourinary Surgery, p.7, 2011, Springer.
[2] Keenan, P. A; "Method for calibrating a SCARA Robot", US Patent 4725965, 1988.
[3] Nunes J; “SCARA Robots: Still Speedy and Dependable”, Robotics World; 2005.
[4] Sprovieri J; "X-Y-Z: SCARA Robots Excel at Assembly", 2003.
[5] Rafael V. A, Dalton M. T, Glauco C; "SCARA Robot Controller Using Real Time Linux"; IEEE 1-4244-1264-1/07; 2007.
[6] Dalton M. T, Rafael V. A., Glauco A.; “Upgrade of SCARA Robot using OROCOS”; Proceedings of the 13th IASTED International Conference, Robotics and Applications, Wurzburg, Germany; 2007.
[7] Onozato K, Maeda Y; “Learning of inverse-dynamics and inverse-kinematics for two-link SCARA robot using neural networks”; SICE Annual Conference, Kagawa University, Japan; 2007.
[8] Kucuk S, Bingul Z; "The Inverse Kinematics Solutions of Industrial Robot Manipulators"; IEEE 0-7803-8599-3/04; 2004
[9] Taylan Das M, Canan Dulger L; 'Mathematical Modelling, simulation and experimental verification of a SCARA Robot'; Simulation Modelling Practice and Theory, Vol 13, pp. 257-271; 2005.
[10] Ribeiro A. da S, Almeida A. G. S, Souza M. B. de, Lima E. J; "Methodology for Implementation of Retrofitting for Drivers of Equipment for Automation Procedures"; 8th Iberoamerican Congress of Mechanical Engineering, 2007.
[11] Becerra V. M, Cage C. N. J, Harwin S. W, Sharkey P. M; "Hardware Retrofit and Computed Torque Control of a Puma 560 Robot"; IEEE 0272-1708/04; 2004.
[12] Miskon M. F; "Retrofitting PUMA 762 Robot"; Masters thesis, Mechatronics, Newcastle University; 2003.
Appendix

Figure A1. SR-600 Bosch SCARA Robot

Figure A2. Power Amplifier Circuit

© The Author(s) 2019. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (http://creativecommons.org/licenses/by/4.0/).