Application of Gain Scheduling Programming to a 6-Axis Articulated Robot with Fuzzy Logic using LabVIEW®

Seok Jeong Kang¹, Seung Kyu Park¹, Won Jee Chung¹, Yoo Seong Jeong¹, Sung Hun Noe²
¹School of Mechatronics, Changwon National University, Changwon, The Republic of Korea
²Youngchang Robo-tech, Gimhae, The Republic of Korea

Abstract –
As demand for industrial robots and AGVs (Automated Guided Vehicles) increases, higher performance is also required. The existing PID control works well with linear systems, but proves ineffective for nonlinear or high-order systems. In contrast, as part of the intelligent control system, fuzzy controllers are a direct control method leveraging human knowledge and experience and then easily controlling highly nonlinear, uncertain and complex systems.
In this paper combines a LabVIEW®-based fuzzy controller and demonstrates better performance than the existing gain tuning. First, the work area will be set based on forward kinematics and inverse kinematics programs. Next, LabVIEW® is used to configure the fuzzy controller and perform the gain scheduling. Finally, the proposed fuzzy gain scheduling is compared with the existing gain scheduling suggested by Kim et al. [3] in terms of responsiveness.

Keywords: Gain scheduling, 6-axis articulated robot, AGV, Fuzzy inference system, Fuzzy gain tuning, RS2, LabVIEW® Programming

1 Introduction
The demand for industrial robots or AGVs has been increasing gradually, with their applications expanding. Also, to meet the need for further performance improvement, high-dimensional robots or AGV control technology are developed. Notably, the dynamic performance of 6-axis multi-joint articulated industrial robots used in a wide range of fields such as assembly of parts, welding and transport of extremely heavy objects is substantially influenced by robots’ position and direction, which warrants the PID (Proportional-Integral-Derivative) gain tuning. [1]

The existing PID control works well with linear systems, but proves ineffective for nonlinear or high-order systems. In contrast, as part of the intelligent control system, fuzzy controllers are a direct control method leveraging human knowledge and experience and then easily controlling highly nonlinear, uncertain and complex systems. A fuzzy control system combined with the PID control could provide higher responsiveness in controlling nonlinear high-dimensional systems. [2] In particular, the fuzzy PID control applied to AGVs in charge of complex functions around industrial sites is presumed to outperform the existing control methods.

Kim et al. [3] presented “Application LabVIEW® - based Gain Scheduling Programming to a 6-axis Articulated Robot Considering Kinematic Analysis” at the CSC 2015. This paper applied LabVIEW® programming to integrate the kinematic analysis of a 6-axis articulated (lab-manufactured) robot (called ‘RS2’ in Fig. 1) including its forward and inverse kinematics into a pseudo gain scheduling. Also, RS2’s work scope was divided into 24 subspaces by calculating its joint angles, which correspond to the input values of 6 degrees of freedom (X, Y, Z, α, β and γ) with an inverse kinematics program. Then, optimized gain tuning and gain scheduling were conducted in each subspace with LabVIEW®.

Fig. 1 RS2 System with Servo Driver and PXI® Equipment

The present paper combines a LabVIEW®-based fuzzy controller with the gain scheduling suggested by Kim et al., and demonstrates better performance than the existing gain tuning. First, the work area is set based on forward kinematics and inverse kinematics programs. Next, LabVIEW® is used to configure the fuzzy controller and perform the gain scheduling. Finally, the proposed fuzzy gain scheduling is compared with the existing gain scheduling suggested by Kim et al. in terms of responsiveness.

2 Forward & Inverse Kinematics
In forward kinematics, robots’ point positions, end-effector positions (X, Y and Z) and directions (α, β and γ) are calculated based on the length of each link and the angle of each joint given. Figure 2 shows a forward kinematics solution built with LabVIEW® programming, where the end-effector position and direction corresponding to each joint’s
input angle are calculated with a homogeneous transformation matrix, $\mathbf{T}$. The program facilitates the calculation of the homogeneous transformation matrices by modifying input angles. The foregoing forward kinematics routine is often called in the interpolation programs for RS2. [3]

![Forward Kinematics Program](image1)

(a) Front Panel

(b) Block Diagram

Fig. 2 Forward kinematics program

By contrast, in inverse kinematics, each joint angle as well as the joint angles corresponding to the input values of 6 degrees of freedom ($X, Y, Z, \alpha, \beta,$ and $\gamma$) is calculated based on the length of each link and the positions of points given. Figure 3 shows an inverse kinematics solution implemented with LabVIEW® programming. The inverse kinematics program is a sub VI® linked to an interpolation program, where the inverse kinematics program calculates each joint angle every sampling time (a few milliseconds). [3]

![Inverse Kinematics Program](image2)

(a) Front Panel

(b) Block Diagram

Fig. 3 Inverse kinematics program

3 Gain Tuning Parameter [4]

3.1 Proportional Gain ($K_P$) of Velocity Control Loop

It is efficient to adjust parameters in the innermost loops first, that is, from the velocity control loop toward the position control loop. Also, the proportional gain ($K_P$) should be adjusted first. As the value governs the responsiveness and normality of the velocity loop, the gain should be high. Still, as the noise increases unless the stability is considered, it should be adjusted to reach a maximum value within the range of stability. As shown in Fig. 4, the closed-loop Bode plot can be transformed into the open-loop Bode plot using the closed-loop Bode plot which is gained based on the relationship between the closed-loop and open-loop transfer functions.
As in Fig. 5, the gain and phase margins are calculated from the Bode plot of the transformed open-loop transfer function, and used as the reference for selecting the proportional gains. In general, as per Nyquist stability [5], the gain margin should be -6dB - -20dB while the phase margin should be 45 degrees or more. Thus, the equation (1) is used for better responsiveness, where the proportional gain calculated is $K_p$ and the gain margin is -6dB.

$$20 \log x = (-6dB) - (-adB)$$

$$x = 10^{rac{-6 - (-a)}{20}}$$

$$K'_p = xK_p$$  \hspace{1cm} (1)

3.2 Integral Gain ($K_i$) of Velocity Control Loop

Integral gains ($K_i$) are associated with steady-state errors. When the integral gain is high, the steady-state error decreases as the robot stops whilst the vibration in its arms increases. Therefore, $K_i$ is adjusted to reach the maximum value, so long as the arms do not vibrate. The value is determined by the integral time constant ($T_i$). The open-loop transfer function of the integrator is found through the block diagram in Fig. 6. As for the Bode plot profile of the transfer function (shown in Fig. 7), the phase approaches 0 at ten times of the point where $T_i$ is calculated (Gain Cross Over Frequency). Therefore, to prevent the phase margin from being influenced by the use of the integrator, the time constant of the integrator should be present at this point. [6]

Using the block diagram between input $X$ (velocity command – velocity feedback) and output $Y$ (current command), the transfer function $G_{vo}$ can be obtained by

$$G_{vo} = \frac{Y}{X} = \left(\frac{K_i}{S} + K_p\right)$$

$$= K_p \left(\frac{K_p^{-1}S + 1}{K^{-1}S}ight) = K_p \left(\frac{S + 1}{T_iS + 1}\right)$$  \hspace{1cm} (2)

3.3 Proportional Gain ($K_p$) of Position Control Loop

The higher the proportional gain ($K_p$) of the position control loop, the better the responsiveness of the servo system, but the greater the robot arm vibration. In case of multi-axis simultaneous control systems such as industrial robots, it is necessary for all axes to have an equal responsiveness to get correct trajectories. Also, the $K_p$'s of axes should be adjusted in order of the moment of inertia, and all the $K_p$'s of axes should share a value, or the lowest value of $K_p$ among the 6 axes. The value is at the point called the cut-off frequency, where the gain becomes -3dB in the Bode plot of the closed-loop transfer function as calculated in Fig. 8. Finally, $K_p$ is adjusted with equation (3). Here, the damping ratio is experimentally calculated. In general, for industrial robots, the value is 0.707. [6]

$$K_p = \frac{\pi f_c}{2\zeta^2}$$  \hspace{1cm} (3)
4 Fuzzy Gain Scheduling

4.1 Fuzzy Inference System [7]

Fuzzy inference system is well known to represent the complex, poorly defined and uncertain systems, which the existing mathematical system modeling fails to properly represent, based on the if-then rule. Fig. 9 shows a common fuzzy inference system. A general controller’s transfer function takes the form shown in the Eq. (4).

\[
K(s) = K_p + K_i z^{-1} = \frac{u(s)}{e(s)}
\]  

(4)

Here, \(u(s)\) and \(e(s)\) are the input and the error between the reference value and the output, respectively. To address several challenges that might arise from applying a linear controller to a nonlinear model, the fuzzy logic is used for the gain scheduling. The controller used here is built as shown in Fig. 10. It is assumed that the gain parameters, \(K_p\), \(K_i\) and \(K_v\), meet the Eq. (5).

\[
K_{min} \leq K_j \leq K_{max} \quad (j = p, i, v)
\]  

(5)

Fig. 8 Bode diagram of closed loop transfer function

Fig. 9 Bode diagram of closed loop transfer function

Fig. 10 Bode diagram of closed loop transfer function

Here, \(K_{min}\) and \(K_{max}\) are calculated experimentally or analytically. They should be noted, however, that the present paper takes those coefficients from the stable range of the root locus [8], given they should be chosen within the range that meets the performance. For convenience, \(K_j\) is normalized with the linear transformation as in the equation (6). \(K^n_j\) is within the range of \(0 \sim 1\).

\[
K^n_j = \frac{K_j - K_{min}}{K_{max} - K_{min}} \quad (j = p, i)
\]  

(6)

4.2 Fuzzy Rule and Membership Functions [9]

Now, \(\{K^n_p, K^n_i, K^n_v\}\) is determined based on the following fuzzy rule.

\[
\text{If } e(t) \in A_i \text{ and } \dot{e}(t) \in B_i, \text{ then } K^n_p \in C_i, K^n_i \in D_i, \quad (7)
\]

and \(K^n_v \in E_i\), for \(i = 1, 2, ..., N\)

Here, \(A_i\) and \(B_i\) values are determined by the general triangular membership functions of the fuzzy set. The membership functions of the fuzzy set for \(e(t)\) and \(\dot{e}(t)\) are shown in Fig. 11. In the figure, N and P stand for negative and positive, respectively. ZO indicates nearly zero. S, M and B refer to small, medium and big, respectively. The fuzzy sets of \(C_i, D_i\) and \(E_i\) are determined to be B (big) or S (small) as in the Eq. (8-1) and (8-2).

\[
\mu_A(x) = -\frac{1}{4} \ln(x) \quad (8-1)
\]

\[
\mu_B(x) = -\frac{1}{4} \ln(1 - x) \quad (8-2)
\]

Fig. 11 Membership function for \(e(t)\) and \(\dot{e}(t)\)

In the equation (7), the \(C_i, D_i\) and \(E_i\) of the fuzzy rule are expressed in the step response characteristic of a general 2nd-order system as shown in Fig. 12.
Fuzzy rules vary across designers. The fuzzy rule used in this paper is as follows. At the point $a_1$ in Fig. 12, it is predictable to give a big control input to reach the set point. That is, a big proportional gain of velocity, a small proportional gain of position, and a big integral gain of velocity are needed. Likewise, the fuzzy rule at $b_1$ should give a small control input to avoid a big overshoot. In other words, a small proportional gain of velocity, a big proportional gain of position, and a small integral gain of velocity are needed. Considering the foregoing, the following rules are possibly formulated around $a_1$ and $b_1$.

$$\begin{align*}
 & \text{if } e(t) \text{ is } PB \text{ and } \dot{e}(t) \text{ is } ZO, \\
 & \text{then } K_p^n \text{ is } Bg, K_i^n \text{ is } Sm \text{ and } K_d^n \text{ is } Bg.
\end{align*}$$

$$\begin{align*}
 & \text{if } e(t) \text{ is } ZO \text{ and } \dot{e}(t) \text{ is } NB, \\
 & \text{then } K_p^n \text{ is } Sm \text{ and } K_d^n \text{ is } Bg, \text{ and } K_i^n \text{ is } Sm \text{ and}
\end{align*}$$

Table I shows a relevant Fuzzy Rule Table.

<table>
<thead>
<tr>
<th>$\dot{e}(t)$</th>
<th>NB</th>
<th>NS</th>
<th>ZE</th>
<th>PS</th>
<th>PB</th>
</tr>
</thead>
<tbody>
<tr>
<td>NB</td>
<td>NB</td>
<td>NB</td>
<td>NS</td>
<td>NS</td>
<td>ZE</td>
</tr>
<tr>
<td>NS</td>
<td>NS</td>
<td>NS</td>
<td>NS</td>
<td>ZE</td>
<td>PS</td>
</tr>
<tr>
<td>ZE</td>
<td>NS</td>
<td>NS</td>
<td>NS</td>
<td>ZE</td>
<td>PS</td>
</tr>
<tr>
<td>PS</td>
<td>NS</td>
<td>ZE</td>
<td>PS</td>
<td>PS</td>
<td>PB</td>
</tr>
<tr>
<td>PB</td>
<td>ZE</td>
<td>PS</td>
<td>PS</td>
<td>PB</td>
<td>PB</td>
</tr>
</tbody>
</table>

In the equation (7), the actual value $\mu_i$ of the $i$-th rule is the product of membership functions based on the assumption in the Eq. (9).

$$\mu_i = \mu_{A_i}(e(t)) \cdot \mu_{B_i}(\dot{e}(t))$$  \hspace{1cm} (9)

Here, $\mu_{A_i}$ is the value of membership function of the fuzzy set $A_i$ for the given $e(t)$. $\mu_{B_i}$ is the value of membership function of the fuzzy set $B_i$ for the given $\dot{e}(t)$. Following the Eqs. (8) and (9) about the process of the fuzzy rules, the values of $\{K_p^n, K_i^n, K_d^n\}$ for each rule are decided. Figure 13 shows the diagram of the process.

Also, as the membership function in Fig. 11 is used, the condition in the Eq. (10) follows, while the defuzzification is contingent on the condition in the Eq. (11).

$$\sum_{i=1}^{N} \mu_i = 1$$  \hspace{1cm} (10)

$$K_j^n = \sum_{i=1}^{N} \mu_i K_j^n (j = p, i, v)$$  \hspace{1cm} (11)

Once $\{K_p^n, K_i^n, K_d^n\}$ is decided, the coefficients of the PID controller are calculated as in the Eq. (12) based on the Eq. (5) and Eq. (6):

$$K_j = (K_{jm,\text{on}} - K_{jm,\text{off}})K_j^n + K_{jm,\text{on}} (j = p, i, v)$$  \hspace{1cm} (12)

4.3 Gain Tuning using Fuzzy Logic

Kim et al. conducted the gain tuning of the 24 subspaces. Equation (13) represents the condition of subspace 1 and is set to the same condition as Eq. (13) for response verification.

$$\theta_1 = \theta_2 = \theta_3 = \theta_4 = \theta_5 = \theta_6 = 0^\circ$$  \hspace{1cm} (13)

The present paper takes a different approach by applying the fuzzy logic to configure the fuzzy PID controller. The relevant gain scheduling program is shown in Fig. 14. When the coordinates are entered, the angle of each joint is calculated while it moves. While moving to the given coordinates, it is controlled by the fuzzy PID in the moving paths.
4.4 Comparison of Velocity Response

To verify its performance, a velocity command is assigned at 0.5Vrms (root-mean-square Voltage) as suggested by Kim et al. The responsiveness of a random spot C on the path of RS2 moving from A to B (so-called coordinated motion in robotics) as shown in Fig. 15 is compared.

Fig. 15 Coordinated motion on RS2

Fig. 16 compares three velocity response levels on each axis: no gain scheduling vs. gain scheduling suggested by Kim et al. vs. the fuzzy gain scheduling. The comparison of the responsiveness across the three cases highlights the fuzzy gain scheduling outperforms the others in terms of responsiveness.

5 Conclusion

The present paper has explored the LabVIEW® programming capable of automatic scheduling of diverse motions of robots and ensuring high responsiveness via the fuzzy PID control. PID gain parameters are used to configure the fuzzy rules, fuzzy relations and membership functions, based on which the fuzzy gain scheduling is implemented with LabVIEW® programming. To verify the performance of the LabVIEW®-based fuzzy gain scheduling programming, the velocity response has been measured in three cases, i.e. no gain scheduling programming applied with the velocity command, an existing PID gain scheduling programming, and the proposed fuzzy gain scheduling. The comparison of the responsiveness across the three cases highlights the fuzzy gain scheduling outperforms the others in terms of responsiveness. The present findings will be conducive to further studies on controlling industrial robots and high-dimensional industrial AGVs.
6 References


7 Acknowledgement

This research was supported by The Leading Human Resource Training Program of Regional Neo industry through the National Research Foundation of Korea(NRF) funded by the Ministry of Science, ICT and future Planning(2016-0143)