Journal of Physics: Conference
Series

PAPER • OPEN ACCESS

Position control of Stewart platform with electric linear actuator

Position control of Stewart platform with electric linear actuator

M R Mohamed[a], A A Roshdy[b], A A Ali[a] and M A Fayed[a]

[a] Department of Engineering Mechanics, Military Technical College, Cairo, Egypt.
[b] Department of Aerospace Engineering, Military Technical College, Cairo, Egypt.

E-mail: m.raafat@mtc.edu.eg

Abstract. In recent decades, several technological applications have depended on manipulators like Stewart platform due to its accuracy and precision. Based on actuation type, Stewart platforms could be rotary or linearly actuated. Electric linear actuators are devices commonly consist of DC or AC motors coupled with lead screw or mechanism of gears and spindle which are used to convert rotary motion into linear (push or pull) motion. To increase accuracy and precision and achieve the desired linear actuator response, controllers should be used. The main aspiration of this work is to investigate the feasibility of using PID controllers for position control of Stewart platform with linear actuators. The mathematical model has been derived and the model transfer function has been obtained. To meet the required response of performance characteristics, the PID controller has been designed based on analysis of root-locus. Model simulation analysis has been carried out on both MATLAB and Simulink. For the electric linear actuator, comparison between the obtained response and the results of Ziegler-Nichols tuning method has been discussed on basis of the specifications of the time response. The PID controller parameters for the electric linear actuator has been tested experimentally and compared with simulation results.

Keywords: Electric linear actuator, DC motor, PID controller, root-locus, MATLAB, Simulink, Ziegler-Nichols, Stewart platform, Bode plot.

1. Introduction

Parallel manipulators like Stewart platform shown in figure (1), are widely used in several fields of technology. This platform could be either rotary actuated by motors or linearly actuated by linear actuators. There are many different styles of linear actuators. The common one is a rod style at which a rod shaft moves in and out of the actuator body. Other types include, column actuators and track actuators. In electric linear actuators the rotary motion of electric motors is converted into linear motion. Electric linear actuators are commonly position controlled. One of most important feedback control systems is the proportional, integral and differential closed loop control (PID) [1]. In order to design an effective controller to achieve the required response of the electric linear actuators without negatively affecting its stability, The mathematical model of the electric linear actuator should be derived properly [2]. Based on the mathematical model, the system could be put on the block diagram form and the transfer function equation could be generated to facilitate simulation analysis. One of the best methods used to design and calculate the tuning parameters of the PID controller is the root-locus method. In this method, according to the design specifications and requirements the root locus of the system is plotted then controller gains could be calculated [3]. Many tuning methods have discussed methodologies to obtain the PID controller parameters. Ziegler-Nichols tuning method was presented in 1942 and till now this method is still widely used [4]. In electric linear actuators, the position controller is tacking travel displacement signal to drive the actuator till reaching the desired position [5]. Sensors must be used to measure the travel displacement as well as to complete the closed loop control circuit. This paper studies an electric linear actuator consists of 12V DC motor coupled with lead and screw through a flexible metal joint. The effect of (push or pull) could be achieved through using L298 H-Bridge integrated with Arduino micro-controller. An ultrasonic sensor is used to feedback the system with measured output signals. The DC motor control parameters were estimated using Simulink. The design of the PID controller was done using root-locus method and compared with Ziegler-Nichols tuning method. The PID controller parameters has been tested experimentally and the results were explained and compared with simulation results. The paper investigates the application of PID controllers in the position control of linear actuators and in turn control parallel manipulators. It also analyses their performance using simulation taking into consideration noise and disturbance. The Bode plot of the system before and after using the controller has been analysed to determine peak gain, phase margin and gain margin to study the system stability.

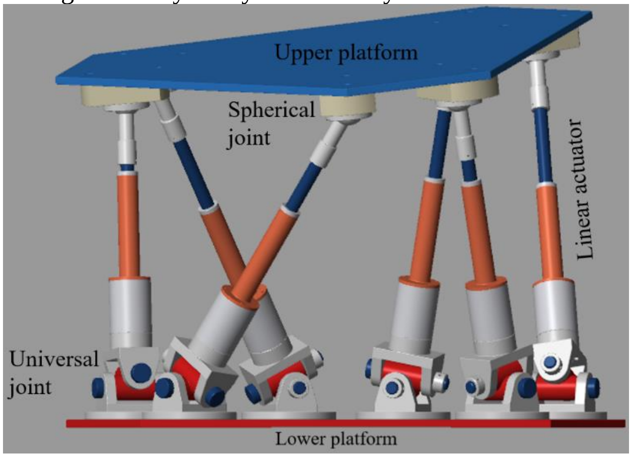

Figure 1. Stewart platform.

2. Electric Linear Actuator Model

The model under investigation is an electric linear actuator consists of 12V DC motor and lead screw coupled together with flexible metal coupling. The rotary motion of the DC motor is converted into linear as the actuator rod has been designed to move only along its axis through groove guides in rod casing. When the DC motor rotates clock-wise or counter-clock-wise the, actuator rod is extended or retracted. The direction of rotation of the DC motor is controlled by simply reversing polarity through the Arduino micro controller and motor driver. Every linear system can be described by a set of equations on the form

$$\begin{aligned}\dot{x} &= Ax + Bu + Ez \\ y &= Cx + Du\end{aligned} \qquad (1)$$

Where A is state (or system) matrix, x is state vector, B is input matrix, u is Input (or control) vector, E is error or (disturbance) matrix, z is error vector, C is output matrix and D is feed-through (or feed-forward) matrix. The model of the DC motor circuit is shown in figure (2).

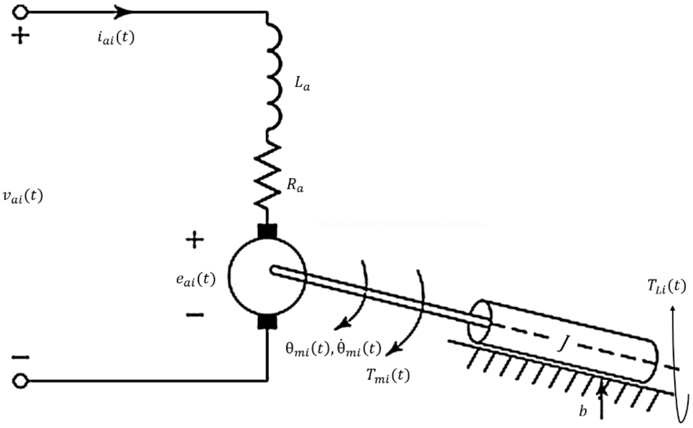

Figure 2. DC motor circuit.

The developed torque of the motor T_{m_i} and back electromotive-force voltage $e_{ai}(t)$ could be computed as follows

$$\begin{aligned} T_{m_i}(t) &= K_m i_{ai}(t) \\ e_{ai}(t) &= K_b \dot{\theta}_{mi}(t) \end{aligned} \quad (2)$$

Where K_m is motor torque constant, $i_{ai}(t)$ is armature current, K_b is back electromotive-force voltage constant, $\theta_{mi}(t)$ is motor angular coordinate and $\theta_{mi}(t)$ is motor angular velocity. By applying Kirchhoff current law on the circuit shown in figure (2) the following equation could be derived

$$\begin{aligned} v_{ai}(t) &= R_a i_{ai}(t) + L_a \frac{di_{ai}(t)}{dt} + e_{ai}(t) \\ v_{ai}(t) &= R_a i_{ai}(t) + L_a \frac{di_{ai}(t)}{dt} + K_b \dot{\theta}_{mi}(t) \end{aligned} \quad (3)$$

Where $v_{ai}(t)$ is the input voltage applied to the armature, R_a is armature winding resistance, $i_{ai}(t)$ is armature current and L_a is armature winding inductance. From equation (3) the rate of change of armature current $i'_{ai}(t)$ could be on the following form

$$i'_{ai}(t) = \frac{-R_a}{L_a} i_a(t) - \frac{K_b}{L_a} \dot{\theta}_{mi}(t) + \frac{1}{L_a} v_{ai}(t) \quad (4)$$

By taking all torques on rotor shaft and substituting from equation (2) with motor developed torque equation, then motor torque and armature current are as follows

$$\begin{aligned} T_{mi}(t) &= T_{Li}(t) + b\dot{\theta}_{mi}(t) + J\ddot{\theta}_{mi}(t) \\ i_{ai}(t) &= \frac{1}{K_m} T_{Li}(t) + \frac{b}{K_m} \dot{\theta}_{mi}(t) + \frac{J}{K_m} \ddot{\theta}_{mi}(t) \end{aligned} \quad (4)$$

Where T_{L_i} is load torque, $\ddot{\theta}_{mi}(t)$ is motor angular acceleration, b is viscous damping coefficient and J is total effective inertia affecting motor. Then from equation (5) the motor angular acceleration $\ddot{\theta}_{mi}(t)$ could be on the following form

$$\ddot{\theta}_{mi}(t) = \frac{K_m}{J} i_{ai}(t) - \frac{b}{J} \dot{\theta}_{mi}(t) - \frac{1}{J} T_{Li}(t) \quad (6)$$

By setting the states to be motor angular coordinate, motor angular velocity and armature current the DC motor state space model could be on the following form

$$\begin{bmatrix} \dot{\theta}_{mi}(t) \\ \ddot{\theta}_{mi}(t) \\ i'_{ai}(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & \frac{-b}{J} & \frac{k_m}{J} \\ 0 & \frac{-k_b}{L_a} & \frac{-R_a}{L_a} \end{bmatrix} \begin{bmatrix} \theta_{mi}(t) \\ \dot{\theta}_{mi}(t) \\ i_{ai}(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \frac{1}{L_a} \end{bmatrix} [v_{ai}(t)] + \begin{bmatrix} 0 \\ \frac{-1}{J} \\ 0 \end{bmatrix} [T_{Li}(t)] \quad (7)$$

$$\begin{bmatrix} \theta_{mi}(t) \\ \dot{\theta}_{mi}(t) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \theta_{mi}(t) \\ \dot{\theta}_{mi}(t) \\ i_{ai}(t) \end{bmatrix} + [0][v_{ai}(t)] \quad (8)$$

The effect of lead screw is represented in converting the rotary motion into linear motion. Figure (3) shows the relation between rotary motion and linear motion.

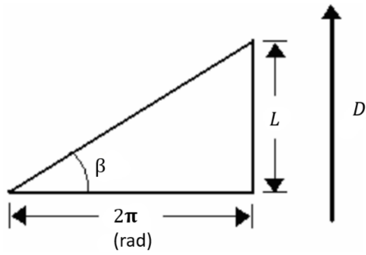

Figure 3. Relation between rotary and linear motion.

From the figure (3) the angle of lead β and the pitch of lead P could be calculated as follows

$$\begin{aligned} \beta &= tan^{-1}\left(\frac{L}{2\pi}\right) \\ P &= \frac{L}{2\pi} \end{aligned} \quad (9)$$

Where L is the displaced step of the lead screw per one revolution. The electric linear actuator displacement $D_i(t)$ and velocity $v_i(t)$ are calculated as follows

$$\begin{aligned} D_i(t) &= P\theta_{mi}(t) \\ v_i(t) &= P\dot{\theta}_{mi}(t) \end{aligned} \quad (10)$$

For the holding load torque

$$T_{Li}(t)=\frac{F_{\text{acti}}\,(t)\times L\times\eta}{2\pi}\ (11)$$

Where $F_{\text{acti}}\,(t)$ is the force acting on the linear actuator and η is lead screw efficiency. The total effective inertia J is calculated as follows

$$J=J_a+J_L\ (12)$$

By substitution from equations (10), (11) and (12) into the state space model derived in equations (7) and (8) the state space mathematical model of the electric linear actuator would be on the following form

$$\begin{bmatrix}\dot{\theta}_{mi}(t)\\ \ddot{\theta}_{mi}(t)\\ i'_{ai}(t)\end{bmatrix}=\begin{bmatrix}0 & 1 & 0\\ 0 & \frac{-b}{J_a+J_L} & \frac{k_m}{J_a+J_L}\\ 0 & \frac{-k_b}{L_a} & \frac{-R_a}{L_a}\end{bmatrix}\begin{bmatrix}\theta_{mi}(t)\\ \dot{\theta}_{mi}(t)\\ i_{ai}(t)\end{bmatrix}+\begin{bmatrix}0\\ 0\\ \frac{1}{L_a}\end{bmatrix}[v_{ai}(t)]+\begin{bmatrix}0\\ \frac{-L\eta}{2\pi(J_a+J_L)}\\ 0\end{bmatrix}[F_{\text{acti}}\,(t)]\ (13)$$

$$\begin{bmatrix}D_i(t)\\ v_i(t)\end{bmatrix}=\begin{bmatrix}P & 0 & 0\\ 0 & P & 0\end{bmatrix}\begin{bmatrix}\theta_{mi}(t)\\ \dot{\theta}_{mi}(t)\\ i_{ai}(t)\end{bmatrix}+[0][v_{ai}(t)]\ (14)$$

The system has one input (armature voltage) and two outputs (linear travel displacement and linear travel speed). The load force is considered as disturbance force and could be neglected in no load case. Based on the previous state space model, the following block control diagram shown in figure (4) could be concluded for the electric linear actuator.

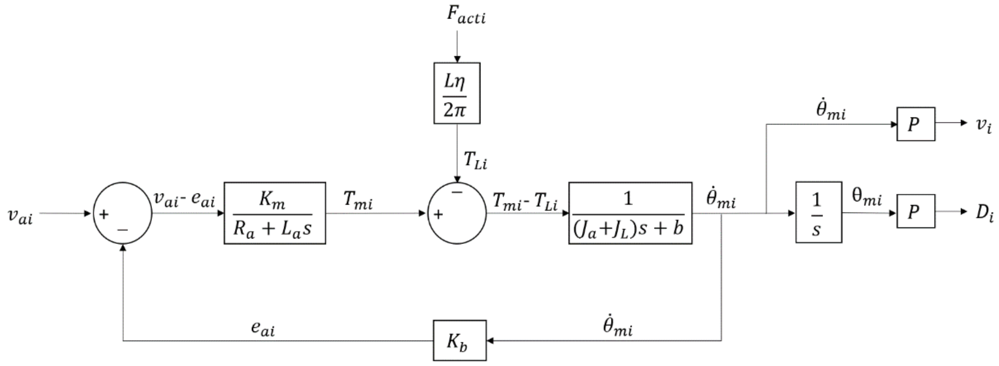

Figure 4. Control model of electric linear actuator.

In order to derive the model transfer function which represents the relation between input volt and travel displacement, assume no load case for simplification.

$$\frac{D_i}{v_{ai}}=\frac{P\,K_m/J\,L_a}{s^3+\left(\frac{JR_a+bL_a}{JL_a}\right)s^2+\left(\frac{bR_a+K_bK_m}{JL_a}\right)s}\ (15)$$

The direct kinematic equation for a parallel manipulator like Stewart platform can be expressed as

$$Q=f(L_i)\ (16)$$

Where Q is the position and orientation of the upper platform, L_i is the vector of linear actuators lengths. The travel displacement of each linear actuator is calculated as

$$D_i=\delta l_i\ (17)$$

Where l_i is the ith linear actuator length. To achieve the desired position and orientation of the upper platform, the desired travel displacement should be achieved accurately and appropriately.

3. Controller design and simulation

To write the equation of the electric linear actuator transfer function, both DC motor parameters and lead screw specifications should be specified. Table (1) shows the parameters of the 12V DC motor used to drive the electric linear actuator and the lead specifications.

Table 1. DC motor parameters and lead screw specifications.

Parameter	Value
R_a	$0.9\,\Omega$
J_a	$4.6841 \times 10^{-8}\ Kg \cdot m^2$
J_L	$5.46406 \times 10^{-7}\ Kg \cdot m^2$
L	$0.008\ m$

To determine the remaining parameters of a DC motor that are necessary for its control, it is necessary to obtain the relationship between the measured DC motor angular speed (rpm) and voltage (V).

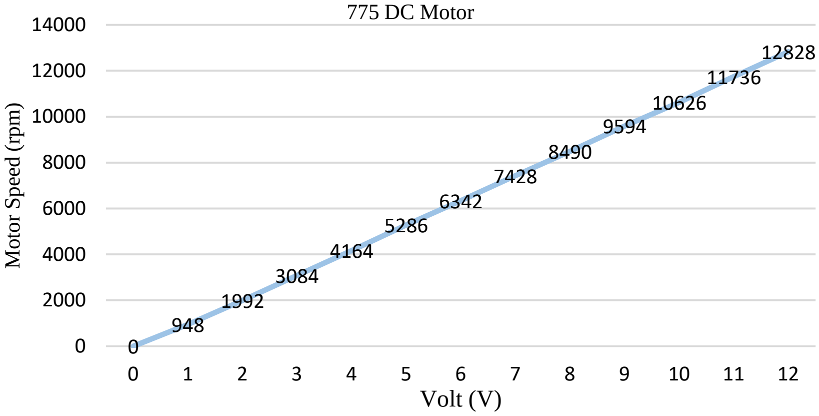

Figure 5. DC motor angular speed (rpm) and voltage (V).

The values that have been estimated using the parameter estimation tool in Simulink, specifically by the nonlinear least squares method, are as shown in table (2)

Table 2. DC motor estimated parameters.

Parameter	Value
L_a	$0.374\ H$
K_m	$8.314 \times 10^{-3}\ N \cdot m/A$
K_b	$7.185 \times 10^{-3}\ \ V/ms^{-1}$
b	$1.71878 \times 10^{-5}\ N \cdot ms$

The electric linear actuator open loop transfer function $G(s)$ could be calculated as follows

$$G(s) = \frac{D_i}{v_{ai}} = \frac{604}{s^3 + 369.3s^2 + 4291s} \quad (18)$$

3.1. PID controller design and simulation

Although this controller type is conventional one, but till now it is widely used [6],[7]. Generally, as shown in figure (6), the desired position (displacement) is the control system input. The error between desired and measured signals is the input for controller. based on error, controller generates the control signal to the driver and in turn it generates the output voltage to the actuator to execute the desired displacement. The output displacement is measured through the displacement sensor and then feeds the control system with the actual measured position. The PID control signal $u(t)$ comes on the following form

$$u(t) = K_P e(t) + K_I \int_0^t e(t)dt + K_D \frac{de(t)}{dt} \quad (19)$$

Where K_P is the proportional tuning gain, K_I is the integral tuning gain, K_D is the derivative tuning gain and $e(t)$ is the error signal. The error could be calculated as follows

$$e(t) = R(t) - y(t) \quad (20)$$

Where $R(t)$ is desired reference displacement and $y(t)$ is actual measured displacement. The PID controller block diagram could be on the following form as shown in figure (6).

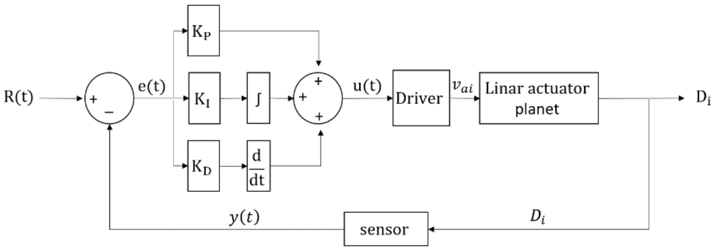

Figure 6. Closed loop PID controller block diagram.

The design procedure starts with assigning the desired system percent overshot $P.O$ and desired settling time t_s.

$$\begin{aligned} P.O &= 5\% \\ t_s &= 0.1 \text{ s} \end{aligned} \quad (21)$$

The dynamic damping ratio ζ could be calculated as follows

$$\zeta = \frac{-ln\left(\frac{PO}{100}\right)}{\sqrt{\pi^2 + ln^2\left(\frac{PO}{100}\right)}} \approx 0.92417 \; (22)$$

The system is under damped system, so the system will produce two complex conjugate poles. Based on the settling time and the dynamic damping ratio the system natural frequency ω_n is calculated from the following approximated formula

$$\omega_n = \frac{4}{\zeta t_s} \approx 43.282 \, rad/s \;\; (23)$$

The system desired poles $P_{1,2}$ are calculated as follows.

$$\begin{aligned} P_{1,2} &= -\zeta\omega_n \pm \omega_n\sqrt{\zeta^2 - 1} \\ P_1 &= -40 + 16.5327i \quad (24) \\ P_2 &= -40 - 16.5327i \end{aligned}$$

Based on plotting the root locus of the linear actuator transfer function $G(s)$ as shown in figure (7), the system poles are $Pole_1 = 0,\ Pole_2 = -357.3363$ and $Pole_3 = -12.008587$. After calculating both desired and system poles and plotting system root locus, the angles between them are calculated as shown in figure (8).

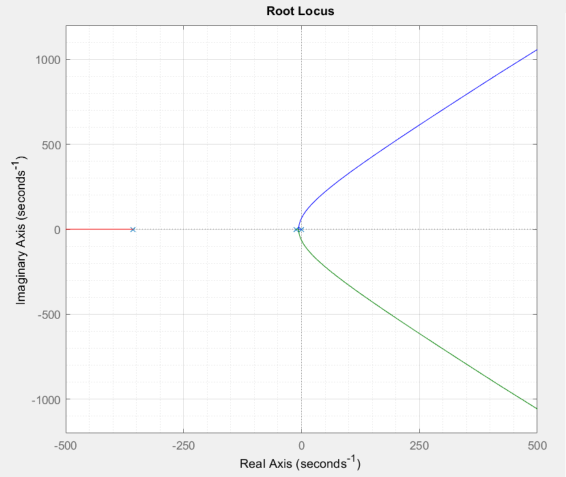

Figure 7. Root locus of linear actuator open loop transfer function.

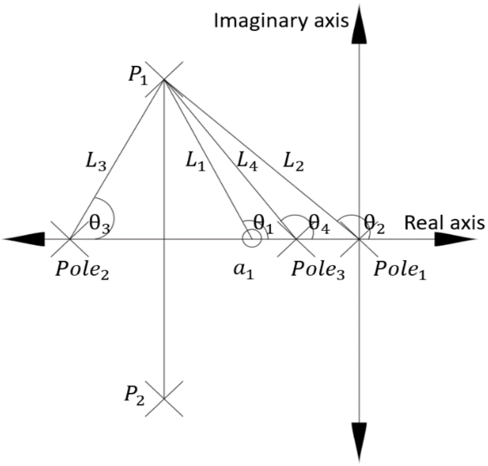

Figure 8. Controller gain calculation using Root locus.

$$\begin{aligned} \theta_2 &= 180^\circ - arctan\left(\frac{imag\ (P_1)}{|(real\ (P_1) -\ pole\,1)|}\right) \approx 157.5438^\circ \\ \theta_3 &= arctan\left(\frac{imag\ (P_1)}{|(\ pole\,2 - real\ (P_1))|}\right) \approx 2.9823^\circ \\ \theta_4 &= 180^\circ - arctan\left(\frac{imag\ (P_1)}{|(real\ (P_1) -\ pole\,3)|}\right) \approx 149.4324^\circ \end{aligned} \tag{25}$$

Where θ_2, θ_3 and θ_4 are angles between $Pole_1$, $Pole_2$ and $Pole_3$ and P_1 respectively. In order to compute the controller constant a_1 which is considered like introducing zero to the system, the constant angle θ_1 should be calculated first.

$$\begin{gathered} \sum \theta_Z - \sum \theta_P = -180^\circ \\ \theta_1 = -180^\circ + (\theta_2 + \theta_3 + \theta_4) \\ \theta_1 \approx 129.9585^\circ \end{gathered} \tag{26}$$

Where $\sum\theta_Z$ is the total summation of all zeroes angles of the system and $\sum\theta_P$ is the total summation of all poles angles of the system. The controller constant will be calculated as follows

$$\begin{gathered} tan\ (\theta_1) = \frac{imag\ (P_1)}{|(real\ (P_1) - a_1)|} \\ a_1 \approx 26.1478 \end{gathered} \tag{27}$$

To measure the gain of controller K_1 graphically to meet the required design specifications $(\zeta, \omega_n, P.O)$, simply the root locus of modified system transfer function $G_1(s)$ will be plotted as shown in figure (9) then the point at which the desired system poles and specifications are met shows that $K_1 \approx 34.3$.

$$G_1(s) = (s + a_1)G(s) \tag{28}$$

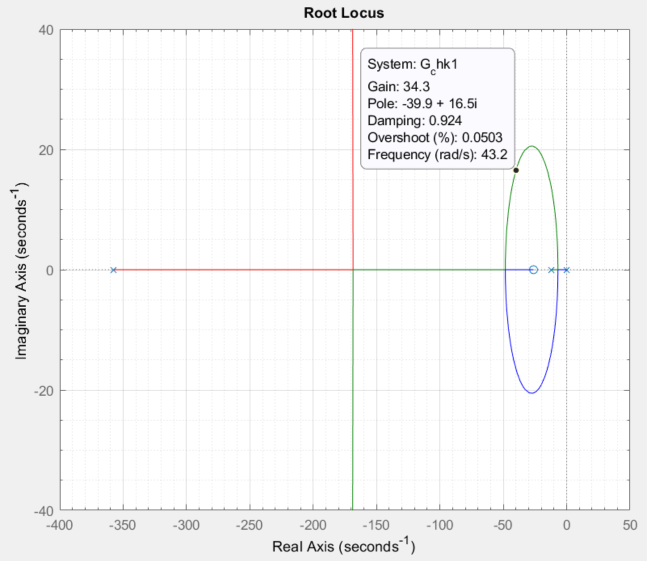

Figure 9. Root locus of modified system transfer function.

The other method used to obtain the controller gain K_1 is calculating it directly from the overall controller gain K_{overall} . To calculate K_{overall} poles constants L_2, L_3 and L_4 and zero constant L_1 would be calculated first based on Pythagorean theory.

$$\begin{gathered} L_1 \approx 21.5688 \\ L_2 \approx 43.282 \\ L_3 \approx 317.76667 \\ L_4 \approx 32.509 \\ K_{\text{overall}} = \frac{L_2 \times L_3 \times L_4}{L_1} \approx 20729.8179 \end{gathered} \tag{29}$$

So, the controller gain will be

$$K_1 = \frac{K_{\text{overall}}}{604} \approx 34.32094 \ (30)$$

Both the graphically measured value and the calculated value are the same for the controller gain K_1. The cascade PID controller is given on the following formula

$$G_{PID}(s) = \frac{K_1(s+a_1)(s+b_1)}{s} \ (31)$$

Where b_1 is controller zero which is chosen close to the origin on the basis of pole zero cancellation to cancel the pole at zero $(pole\ _1)$.

$$b_1 = 0.05 \ (32)$$

The controller gains (K_P, K_I and K_D) are calculated as follows

$$\begin{gathered} K_P = K_1(a_1+b_1) \approx 899.1326 \\ K_I = K_1 a_1 b_1 \approx 44.8708 \ (33) \\ K_D = K_1 \approx 34.32094 \end{gathered}$$

The open loop transfer function G_{open} which represents the transfer function of the cascade controller and the plant.

$$G_{\text{open}} = G_{PID}(s)G(s) \ (34)$$

The closed loop transfer function with unity feedback G_{closed} which represents the overall transfer function of the system.

$$G_{\text{closed}}(s) = \frac{G_{\text{open}}(s)}{1 + G_{\text{open}}(s)} \ (35)$$

The poles of open loop transfer function are $Pole_2$ and $Pole_3$, while the poles of the closed loop transfer function are the desired poles P_1 and P_2 and a neglected non dominant pole $\left(P_{\text{neglected}} = 289.36\right)$ as it has very tiny effect on the system response. The plotting of both open and closed loops root locus explains the previously discussed results clearly as shown in figures (10) and (11).

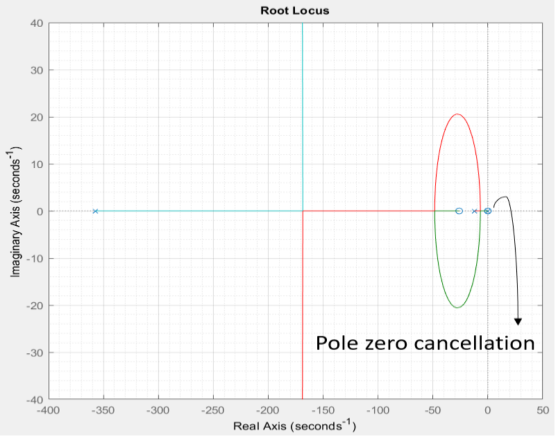

Figure 10. Root locus of G_{open} .

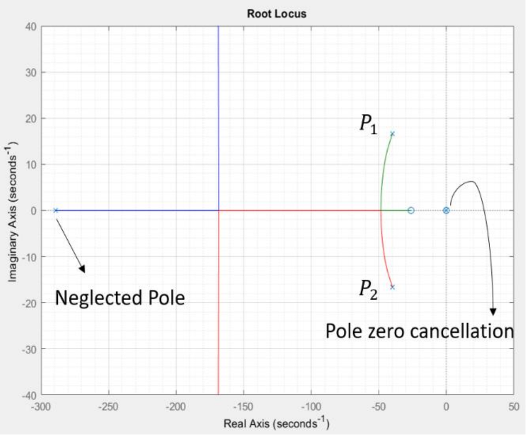

Figure 11. Root locus of G_{closed} .

The actuator model has been structured on Simulink to study and simulate the effect of the PID controller and also to help in studying other different tuning methods as shown in figure (12).

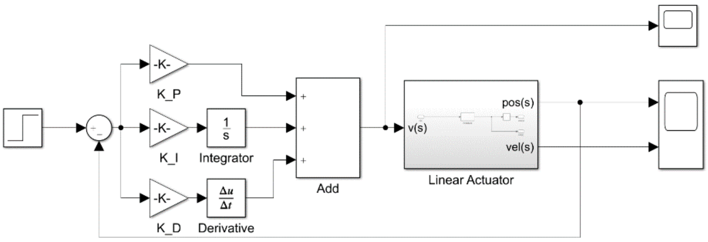

Figure 12. Simulink model of the linear actuator.

The step response of the closed loop transfer function has been simulated as shown in figure (13) and it is characterized by settling in 0.1211 seconds with 9.906% over shoot and 0.0224 seconds rise time. The obtained values of overshot and settling time are just above the desired system specifications due to the approximated formula (23) used to determine the desired natural frequency of the system.

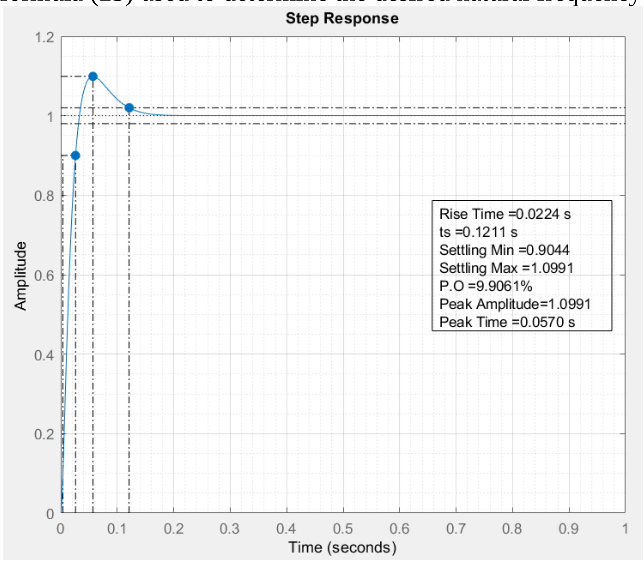

Figure 13. Simulation results of PID controller step response.

The evaluation of the pid controller using simulation must be validated with noise and disturbance. The designed controller is integrated with limiter as shown in figure (14) to limit the control signal and prevent overshooting or oscillations in the system, which can lead to instability. This technique is able to reject load disturbances and minimize the impact of noise on the output signal. By analysing the simulation results shown in figure (15) the controller showed its ability to maintain the desired output despite the presence of disturbances and noise. The step response settles in 1.2046 seconds with 15.392% over shoot and 0.0956 seconds rise time.

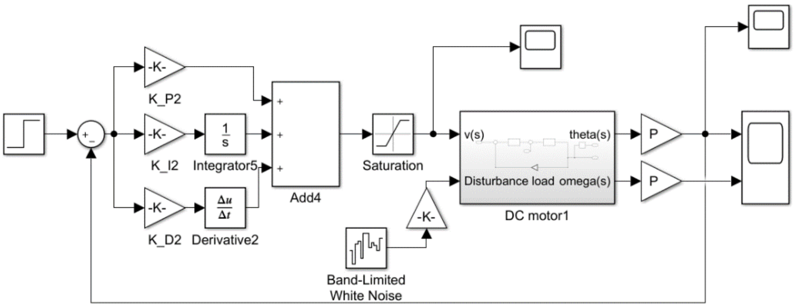

Figure 14. PID controller noise rejection technique block diagram.

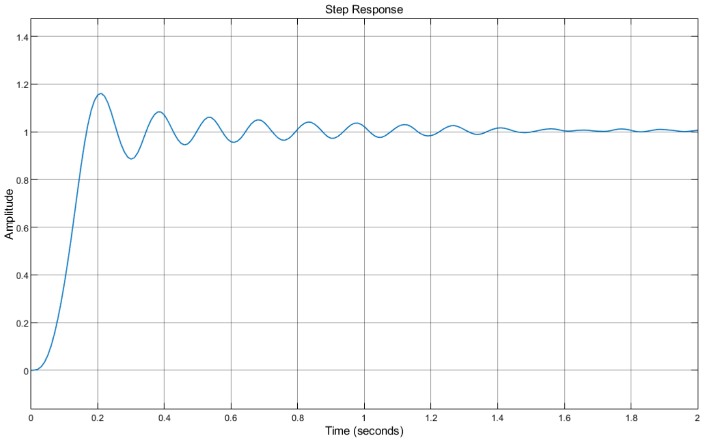

Figure 15. Simulation results of PID controller step response with noise effect**.**

The Bode plot shown in figure (16) shows that $G(s)$ and $G_{\text{open}}\ (s)$ phase margin of 89.3 and 68.2 degrees respectively which means that both are stable, but they are close to the stability limit. The open loop transfer function $G(s)$ can tolerate a gain increase of gain up to 68.4 dB before becoming unstable at 65.5 rad/s. A peak gain of 0.806 dB at a frequency of 26.1 rad/s, indicates that $G_{\text{closed}}\ (s)$ has a resonant frequency at this point. Also, the phase angle of $G_{\text{closed}}\ (s)$ remains below -180 degrees for all frequencies up to and including 0 rad/s indicates that designed control system can handle large time delays without becoming unstable. $G_{\text{closed}}\ (s)$ has a delay margin of 0.0492 seconds, indicates that it can tolerate small time delays at 48.1 rad/s frequency without becoming unstable.

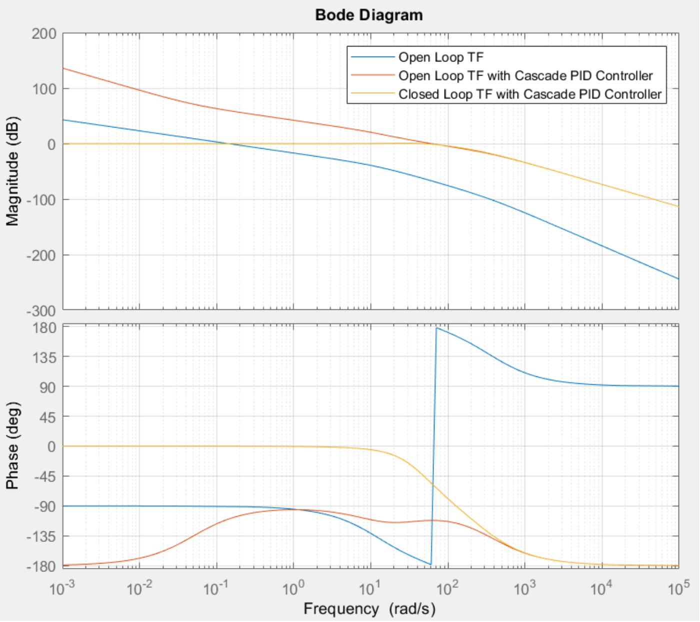

Figure 16. Bode plot of control system.

3.2. Ziegler-Nichols closed loop tuning method

This method is known as continuous cycling method and also may be mentioned as ultimate gain method [8],[9]. The continuous cycling method is one of the best-known strategies of closed loop systems [10],[11],[12]. The PID tuning parameters are calculated as function of both ultimate (critical) proportional gain PU and period TU. The PID tuning parameters are represented in table (3) [13].

Table 3. PID tuning parameters based on Ziegler-Nichols closed loop tuning method.

K_P	K_I	K_D
$0.6K_U$	$(6K_U)/(5T_U)$	$(3/40)K_UT_U$

Based on Simulink model the closed loop ultimate (critical) proportional gain at which the system reaches its sustained oscillation was evaluated as $K_U = 2624$ and the ultimate critical period T_U was graphically measured as shown in figure (17) from the periodic time of five successive peaks $T = 480.556$ ms. The controller parameters are calculated as follows

$$\begin{aligned} T &= 5T_U \\ T_U &= 96.1116 \text{ ms} \end{aligned} \quad (36)$$

Table 4. Values of PID tuning parameters based on Ziegler-Nichols closed loop tuning method.

K_P	K_I	K_D
1574.4	32761.91427	18.91476

The step response of the closed loop transfer function using Ziegler-Nichols tuning method was simulated as shown in figure (18) The response settles in 0.4631 seconds with 76.4360% over shoot and 0.0214 seconds rise time.

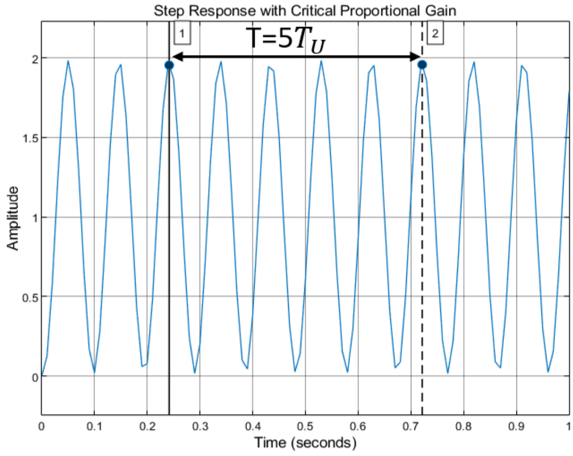

Figure 17. Closed loop step response of with critical proportional gain.

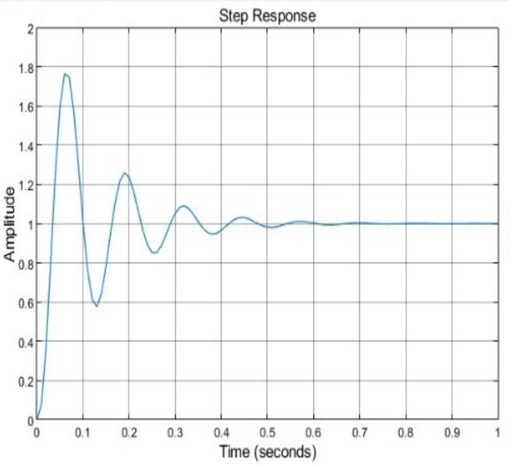

Figure 18. Closed loop step response using Ziegler-Nichols tuning method.

Based on previous response results of the PID controller, the design of controller using root locus is more superior than using Ziegler-Nichols tuning method. The desired system settling time and overshoot has been achieved more accurate by using root locus method. The root locus tuning method enables the control system to effectively reject load disturbances while also achieving improved performance characteristics. Additionally, the integration of a control signal limiter can further enhance the performance of the system.

4. Experimental implementation

The PID controller parameters has been tested using Arduino Mega to generate the control signal for the driver. The L298 dual H-Bridge driver allows controlling the direction and position of the electric linear actuator. The ultrasonic sensor reads the output displacement then feedback the control system with the actual measured displacement. Figures (19) and (20) show the electric linear actuator wiring and implementation.

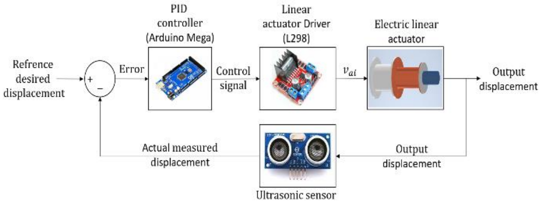
Refrence desired displacement
+
−
Error
PID controller (Arduino Mega)
Control signal
Linear actuator Driver (L298)
v_{ai}
Electric linear actuator
Output displacement
Actual measured displacement
Ultrasonic sensor
Output displacement

Figure 19. Wiring of electric linear actuator position control.

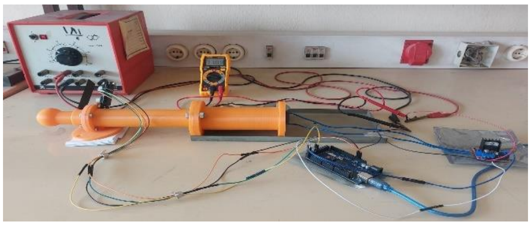

Figure 20. Electric linear actuator experimental model.

The experimental model response shown in figure (21) has shown vividly accurate results in comparison of the simulated PID response of the linear actuator model. When a positive step displacement (D_i=1cm) is set to the model, it has reached the steady state response in 0.32 seconds with overshoot approximately equal to 17%. The designed controller has a stable and consistent response over time. It is able to quickly and accurately respond to changes in the input or set point, without delay.

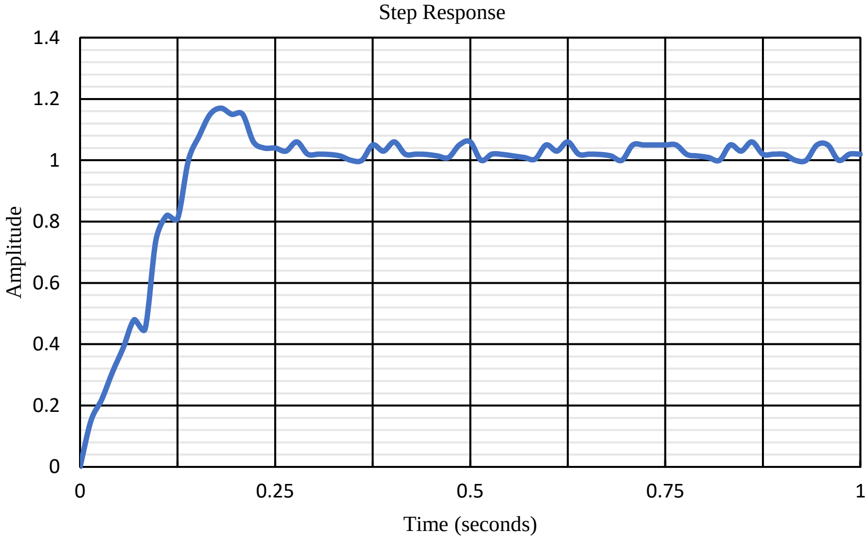

Figure 21. Electric linear actuator experimental step response using PID controller.

5. Conclusion

Parallel manipulators are positional controlled systems. Linear actuators are able to make push or pull effect. In this paper, the electric linear actuator mathematical model, transfer function and the control block diagram were obtained. PID controller could be designed on the basis of root locus method or by using tuning method like Ziegler-Nichols. The control parameters of the DC motor have been estimated using nonlinear least squares method. The simulation results of the step responses for the linear actuator have showed that the response obtained by designing the controller using root locus is more superior than using the tuning method. The tuning method reached the settling response in 0.46 seconds with 76.43% overshoot. By using root locus method, the settling response were reached in 0.12 second with 9.9% overshoot. With presence of load disturbances, the actuator settles in 1.2046 seconds with 15.392% over shoot and 0.0956 seconds rise time. The experimental results of the linear actuator have showed that the designed PID controller has vividly accurate results as the step response has settled in 0.32 seconds with 17% overshoot. The analysis of bode diagram has showed that the open loop transfer functions are close to the stability limit. There is a peak gain of 0.806 dB at a frequency of 26.1 rad/s for the controlled closed loop transfer function. The designed controller is stable and can handle large time delays without becoming unstable and can tolerate small time delays at 48.1 rad/s frequency without becoming unstable.

References

[1] Lekkala, K.K. and Mittal, V.K., 2014, July. PID controlled 2D precision robot. In *2014 International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT)* (pp. 1141-1145). IEEE.

[2] Nayak, B. and Sahu, S., 2019. Parameter estimation of DC motor through whale optimization algorithm. International Journal of Power Electronics and Drive Systems, **10(1)**, p.83.

[3] Xie, J., Yang, R., Gooi, H.B. and Nguyen, H.D., 2023. PID-based CNN-LSTM for accuracy-boosted virtual sensor in battery thermal management system. Applied Energy, **331**, p.120424.

[4] Rao, P.G.K., Subramanyam, M.V. and Satyaprasad, K., 2014, July. Study on PID controller design and performance based on tuning techniques. In 2014 International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT) (pp. 1411-1417). IEEE.

[5] Maung, M.M., Latt, M.M. and Nwe, C.M., 2018. DC motor angular position control using PID controller with friction compensation. International journal of scientific and research publications, **8(11)**, p.149.

[6] Barman, A., Dutta, S., Tiwari, K., Roy, S. and Pain, S., 2022, February. Genetic Algorithm Based Adaptive PID Tuning of Time Delay Process. In International Symposium on Artificial Intelligence (pp. 64-75). Cham: Springer Nature Switzerland.

[7] Senthil Kumar, S. and Anitha, G., 2021. A novel self-tuning fuzzy logic-based PID controllers for two-axis gimbal stabilization in a missile seeker. International Journal of Aerospace Engineering, **2021**, pp.1-12.

[8] Ziegler, J.G. and Nichols, N.B., 1942. Optimum settings for automatic controllers. Transactions of the American society of mechanical engineers, **64(8)**, pp.759-765.

[9] Hemamalini, B., Lakshmi, P. and Navabharathi, S., 2023. Implementation of AI Techniques for Tuning of Controller Parameters in a Nonlinear System. In AI Techniques for Renewable Source Integration and Battery Charging Methods in Electric Vehicle Applications (pp. 243-260). IGI Global.

[10] Melinda, M., Khatir, R., Ibina, A.R.P., Mufti, A., Syahyadi, R. and Hasanuddin, I., 2023. Implementation of PID Controller on Hohenheim Tunnel Dryer Using Ziegler-Nichols Approach Method. Jurnal Polimesin, **21(1)**, pp.101-107.

[11] Murugesan, D., Jagatheesan, K., Kulkarni, A.J. and Shah, P., 2023. A Socio Inspired Technique in Nuclear Power Plant for Load Frequency Control by Using Cohort Intelligence Optimization-Based PID Controller. In Renewable Energy Optimization, Planning and Control: Proceedings of ICRTE 2022 (pp. 1-12). Singapore: Springer Nature Singapore.

[12] Ambroziak, A. and Chojecki, A., 2023. The PID controller optimisation module using Fuzzy Self-Tuning PSO for Air Handling Unit in continuous operation. Engineering Applications of Artificial Intelligence, **117**, p.105485.

[13] Ray, S.K. and Paul, D., 2010. Performance Comparison of Electronic Printwheel System by PI and PID Controller Using Genetic Algorithms. International Journal of Computer Science & Emerging Technologies, **1(4)**.