

ELEC476

Advanced Systems Modelling & Control

Assignment 1 -

System Event-Driven PID Control for
DC Servo Drive & Model Reference Adaptive Control
in a DC Electrical Drive

Khanthapak Thaipakdee

Department of Electrical Engineering and Electronics,
University of Liverpool,
Brownlow Hill, Liverpool L69 3GJ, UK

Table of Contents

1. Chocolate Packaging System Simulation.	1
1.1 System Description.	1
1.2 PID tuning and DC servo simulation.	2
1.3 Design and implement the random event system.	7
1.4 The Overall System event driven and Continuous PID System.	10
2. Model Reference Adaptive Control for a DC Electrical Drive.	25
2.1 Model and Design of the MIT-based DC motor controller and Lyapunov-based DC motor controller.	25
2.2 Results and Analysis of the Simulation.	29
2.3 Comparative Analysis of MIT and Lyapunov-Based Model Reference Adaptive Controllers.	35
2.4 Conclusions.	36
3. References.	36

Table of Figures

Figure	Page
Figure 1 Chocolate packaging system.	1
Figure 2a Rotation arm and 2b picking and dropping arm.	1
Figure 3 Step response of PID DC motor with and without parameter K_p	3
Figure 4 Step response of PID DC motor with different value of K_i	4
Figure 5 Step response of PID DC motor with different value of K_d	4
Figure 6 Simulink scheme of PID controller.	5
Figure 7 Simulink scheme of DC motor for picking and dropping 0.5kg boxes (including rotating 90°). ..	6
Figure 8 Simulink scheme of DC motor for picking and dropping 1 kg boxes (including rotating 180°). ..	6
Figure 9 Simulink scheme of random event system.	7
Figure 10 Uniform random number parameters.	8
Figure 11 If action subsystem block for picking and dropping 0.5kg boxes (including rotating 90°).	9
Figure 12 If action subsystem block for picking and dropping 1kg boxes (including rotating 180°).	9
Figure 13 Velocity profile for picking and dropping.	10
Figure 14 Velocity profile for rotating 90°	10
Figure 15 Velocity profile for rotating 180°	11
Figure 16 Simulink scheme of velocity combination.	11
Figure 17 Combination of velocity profiles.	12
Figure 18 Velocity response compared by K_p	13
Figure 19 Current response compared by K_p	14
Figure 20 Current response compared by K_p	15
Figure 21 Velocity response compared by K_i	16
Figure 22 Current response compared by K_i	17
Figure 23 Velocity response compared by K_d	18
Figure 24 Current response compared by K_d	19
Figure 25 Experimental analysis of 0.5 kg box event: seed 16 (Time: 16 - 32 seconds)	21
Figure 26 Experimental analysis of 1 kg box event: seed 16 (Time: 80 - 96 seconds).	21
Figure 27 Experimental analysis of 0.5 kg box event: seed 16 (Time: 0 - 300 seconds).	22
Figure 28 Experimental analysis of 1 kg box event: seed 16 (Time: 0 - 300 seconds).	22
Figure 29 Experimental analysis of the missing box event: seed 16 (Time: 0 - 300 seconds).	23

Table of Figures (Cont.)

Figure	Page
Figure 30 Experimental analysis of 0.5 kg box event: seed 15 (Time: 0 - 300 seconds).	23
Figure 31 Experimental analysis of 1 kg box event: seed 15 (Time: 0 - 300 seconds).	24
Figure 32 Experimental analysis of the missing box event: seed 15 (Time: 0 - 300 seconds).	24
Figure 33 Block diagram of MRAS [1].	25
Figure 34 Simulink scheme of plant and reference model.....	25
Figure 35 Simulink scheme of MIT-based DC motor controller.....	28
Figure 36 Simulink scheme of Lyapunov-based DC motor controller	28
Figure 37 The output signals ($y(t)$, $y_m(t)$) obtained using the MIT rule.	29
Figure 38 The adjustment error signal ($e(t)$) progression obtained using the MIT rule.	29
Figure 39 The output signals ($y(t)$, $y_m(t)$) obtained using the Lyapunov stability theory.....	30
Figure 40 The adjustment error signal ($e(t)$) progression using the Lyapunov stability theory.	30
Figure 41 Simulink scheme of MIT-based DC motor controller with MRAS pattern.	31
Figure 42 Performance evaluation: MIT rule vs. MIT rule with MRAS pattern in DC motor control	32
Figure 43 Performance evaluation: MIT rule vs. Lyapunov stability theory in DC motor control ($\gamma=3$).	33
Figure 44 Performance evaluation: MIT rule vs. Lyapunov stability theory in DC motor control ($\gamma=5$).	33
Figure 45 Performance evaluation: MIT rule vs. Lyapunov stability theory in DC motor control ($\gamma=0.1$).	34
Figure 46 Performance evaluation: MIT rule vs. Lyapunov stability theory in DC motor control ($\gamma=0.01$).	34
Figure 47 Effect of α on time response curve for MIT rule [2].	35
Figure 48 Effect of α on time response curve for Lyapunov rule [2].	36

1. Chocolate Packaging System Simulation.

1.1 System Description.

This assignment involves simulating a robotic system for picking and dropping chocolate boxes based on their weights. The simulation covers the dynamics of the conveyor system, the robot's decision-making, and the movements of robot arms driven by DC servomotors. The chocolate packaging system was shown in Figure 1.

The robot plays a crucial role in handling chocolate boxes arriving every 16 seconds, with a 1-in-20 chance of a missing box due to malfunctions. An optical sensor detects faults, prompting the robot to wait for the next box.

Boxes weigh either 0.5 kg or 1 kg, with four times as many 0.5 kg boxes. A sensor detects box weight, guiding the robot to drop light boxes at location A (with a 90° rotation) and heavier boxes at location B (with a 180° rotation, as in Figure 2a). Robot arm 2 handles the picking and dropping, illustrated in Figure 2b.

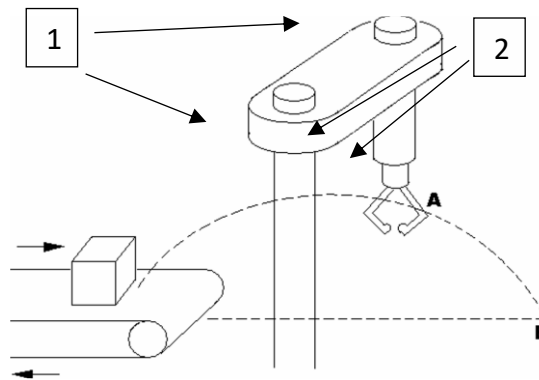


Figure 1 Chocolate packaging system.

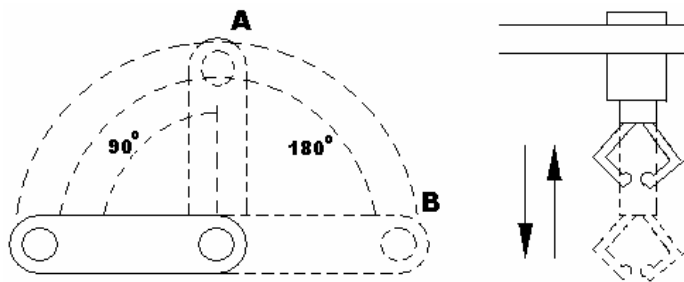


Figure 2a Rotation arm and 2b picking and dropping arm.

Both robot joints use DC servomotors, with picking up and dropping down movements representing transformations of motor rotations.

The motor drive systems can be described by two state-space models below:

- Picking and dropping 0.5kg boxes (including rotating 90°)

$$\dot{x}_1 = -10x_1 + 333.33x_2$$

$$\dot{x}_2 = -0.5x_1 - 10x_2 + 0.1v_a$$

$$\dot{x}_3 = x_1$$

- Picking and dropping 1kg boxes (including rotating 90°)

$$\dot{x}_1 = -7.3x_1 + 250x_2$$

$$\dot{x}_2 = -0.5x_1 - 10x_2 + 0.1v_a$$

$$\dot{x}_3 = x_1$$

where $x_1 = \omega$ (angular velocity in rad/s), $x_2 = i_a$ (armature current), and $x_3 = \theta$ (angular displacement in rad).

1.2 PID tuning and DC servo simulation.

A PID controller is adopted to control the servo motor which has the following structure:

$$v_a = K \left\{ K_p (\omega_{profile} - \omega) + K_i \int (\omega_{profile} - \omega) dt + K_d \frac{d(\omega_{profile} - \omega)}{dt} \right\}$$

where $\omega_{profile}$ is the desired velocity profile.

A PID (Proportional-Integral-Derivative) controller functions as a regulatory mechanism for a system, dynamically adjusting the control input in response to the disparity between the desired setpoint and the actual output of the system. The triad of components within a PID controller—proportional (P), integral (I), and derivative (D)—fulfils distinctive roles in shaping the dynamical characteristics of the system.

The proportional (P) term promptly responds to the prevailing error, the integral (I) term rectifies steady-state error through the integration of cumulative error over time, and the derivative (D) term addresses the rate of change of the error, thereby proactively anticipating and mitigating prospective errors.

The proportional gain (K_p) is a tuning parameter that determines the strength of the proportional action in the overall control output. Increasing K_p makes the controller respond more aggressively to the current error, leading to a faster initial correction. However, too high a value of K_p can result in overshooting, oscillations, or instability.

In this system, increasing the value of K_p results in a rapid rise time, while decreasing the value of K_p leads to a slow rise time and high overshoot, as shown in Figure 3.

The integral gain (K_i) in a PID controller serves to eliminate steady-state error and enhance system stability. Increasing K_i reduces persistent deviations from the setpoint, ensuring better accuracy. However, higher K_i values may slow the response to sudden changes, potentially leading to integrator windup and affecting transient response.

From Figure 4, there isn't much difference between the step responses of $K_i = 0.01$ and 0.1 . However, when increasing K_i to 1.0 , the response exhibits high overshoot, a fast rise time, and slow settling time.

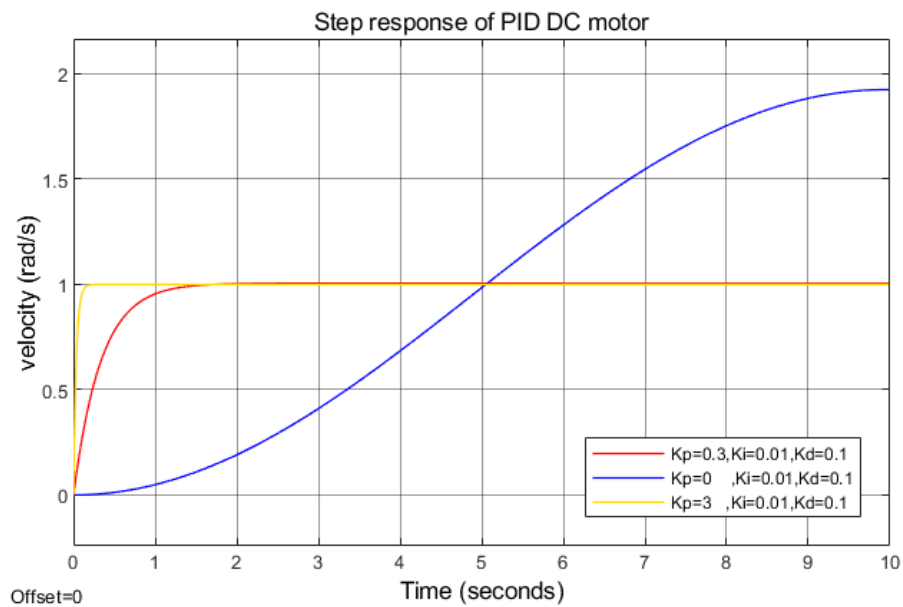


Figure 3 Step response of PID DC motor with and without parameter K_p .

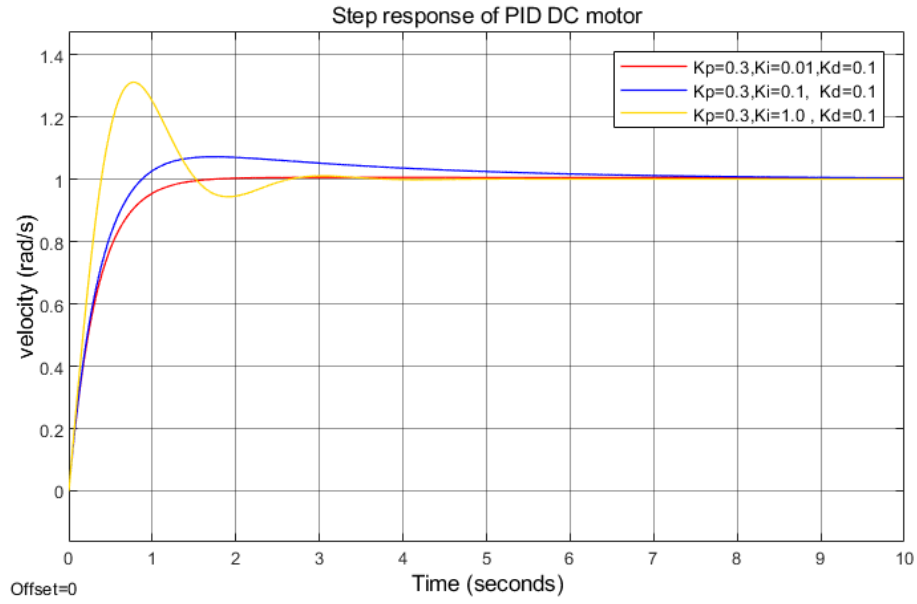


Figure 4 Step response of PID DC motor with different value of K_i .

The derivative gain (K_d) in a PID controller has several effects on the system's response. Higher K_d values contribute to damping oscillations, improving the transient response and reducing overshoot. This results in a faster settling time and enhanced stability.

According to Figure 5, it is evident that the PI controller alone is insufficient for this system, as oscillations occur at the beginning of the time (around 1 second). However, the introduction of a derivative term ($K_d=0.1$) proves effective in eliminating these oscillations. It is noteworthy, though, that a higher K_d value ($K_d=0.2$) results in increased overshoot and a longer settling time.

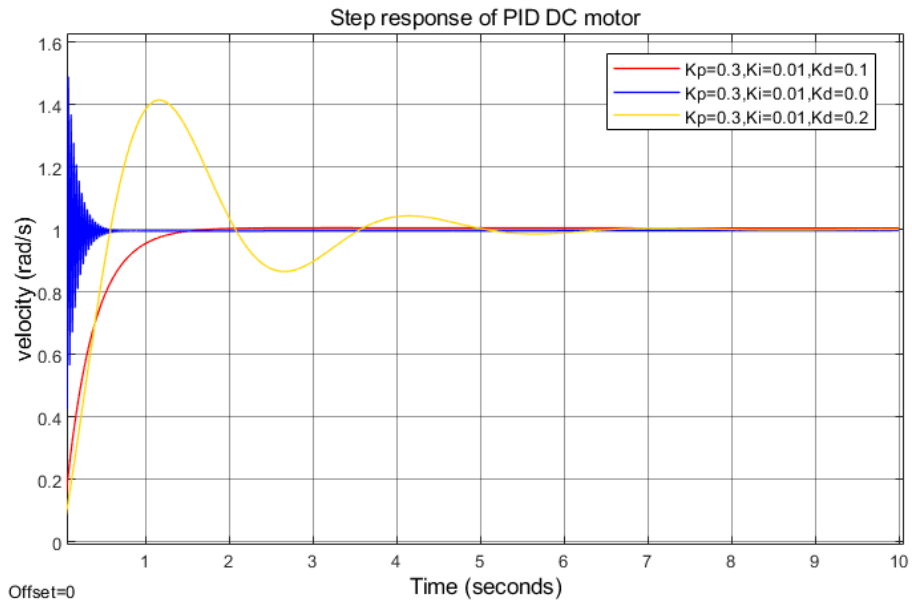


Figure 5 Step response of PID DC motor with different value of K_d .

In Figure 6, a Simulink scheme depicting a PID controller is showcased. This configuration commonly features a "velocity profile" block (v90/v180) to represent the setpoint or reference signal. This velocity profile block is connected to a "PID Controller" block, encompassing adjustable parameters such as K_p , K_i , and K_d . During step response measurement, the velocity profile block is substituted with a step input block to observe the system's response.

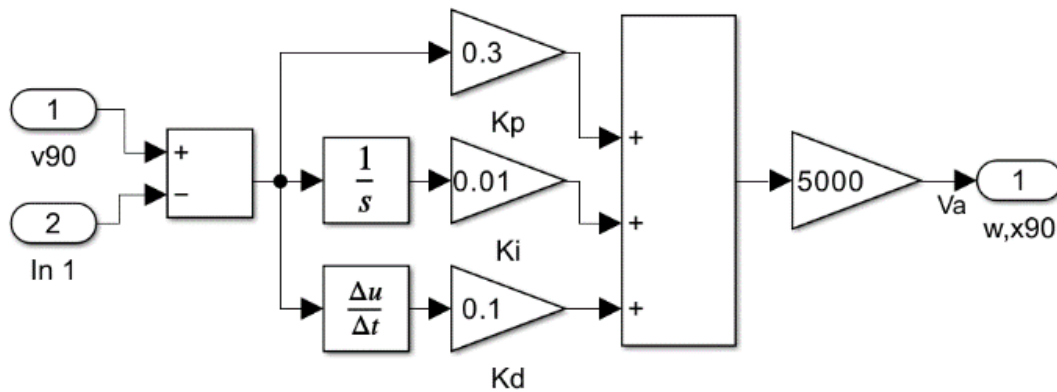


Figure 6 Simulink scheme of PID controller.

Figure 7 presents a detailed Simulink scheme illustrating the operation of a DC motor designed for the precise picking and dropping of 0.5kg boxes, complete with a 90° rotational mechanism. The inclusion of “w,x90” as a feedback block in its PID controller contributes to the system's stability and accuracy. Additionally, “In1” serves as an input port, representing the reference velocity profile tailored for this scenario.

In Figure 8, a parallel Simulink scheme is depicted for the DC motor, this time tasked with handling 1kg boxes and executing a 180° rotation. The “w,x180” feedback block within its PID controller enhances control precision and response. The corresponding input port, “In2”, accommodates the reference velocity profile tailored to the dynamics of 1kg box manipulation

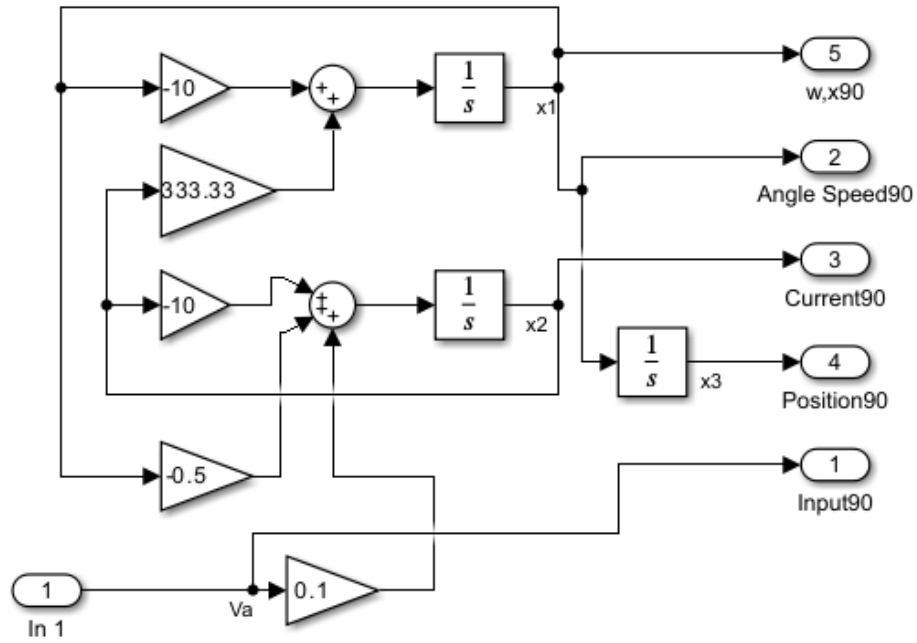


Figure 7 Simulink scheme of DC motor for picking and dropping 0.5kg boxes (including rotating 90°).

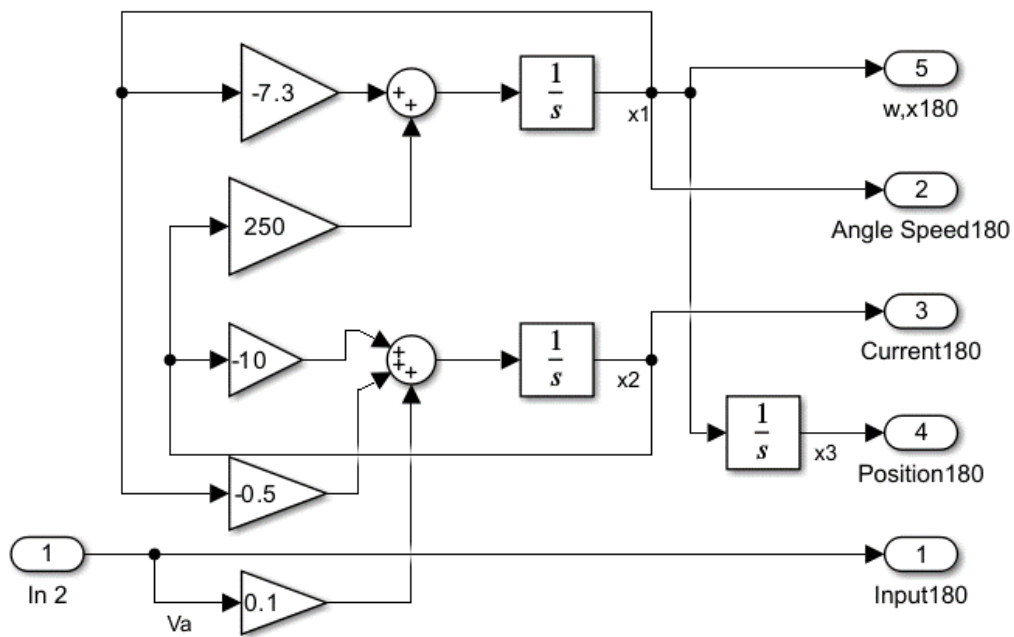


Figure 8 Simulink scheme of DC motor for picking and dropping 1 kg boxes (including rotating 180°).

1.3 Design and implement the random event system.

Figure 9 depicts a Simulink scheme modeling the stochastic dynamics of a random event system. The model uses a uniform random number block to introduce variability in generating probabilities for different box weights. Key components include the uniform random number block, responsible for generating random numbers. The If block evaluates specific conditions, while the If Action Subsystem block is employed within the If block to encapsulate events triggered when certain conditions are met.

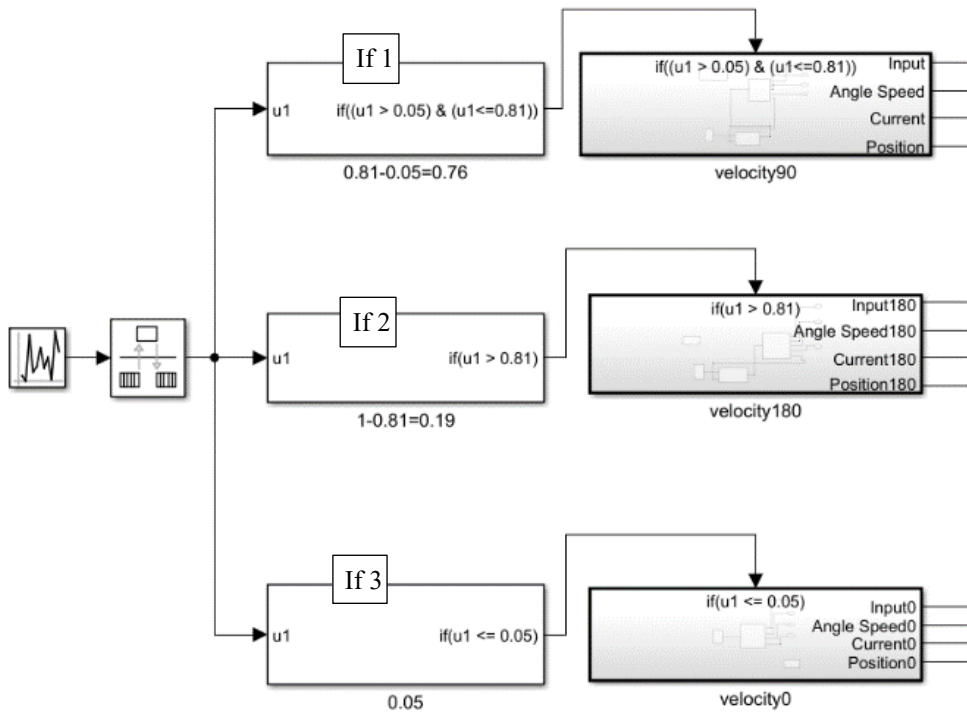


Figure 9 Simulink scheme of random event system.

In the system description, the probability of a missing box event is $1/20$, indicating a relatively rare occurrence. Consequently, the combined probability of events related to the 0.5 kg and 1 kg boxes is $19/20$. Within these events, boxes are classified based on weight, with a four-to-one ratio favouring 0.5 kg boxes. As a result, the probability of a 0.5 kg box event is $76/100$. Conversely, the probability of a 1 kg box event is $19/100$.

The "If1" block condition is determined by a random number between 0.05 and 0.81, representing a range that corresponds to 76% of the 0.5 kg box event. Similarly, the "If2" block condition is determined by a random number between 0.81 and 1.00, reflecting 19% of the 1 kg box event. The "If3" block condition is determined by a random number between 0.00 and 0.05, indicating 5% ($1/20$) probability of a missing box event.

In Figure 10, the adjustable parameters within the uniform random number block play a crucial role in shaping the system's behaviour. Considering the conditions set in the "If" block, where the random number interval should span from 0.00 to 1.00, the minimum is appropriately set to 0, and the maximum is set to 1.

Additionally, indicating a box transfer rate of 16 seconds, the sample time is configured to be 16. This setting ensures that only one event occurs during this period, effectively controlling the timing of box transfers within the system.

Furthermore, the optional seed parameter, when utilized, introduces a different sequence of events in subsequent simulations, providing variability and allowing for the exploration of diverse scenarios.

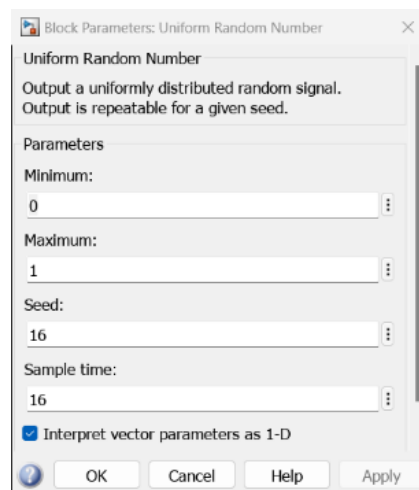


Figure 10 Uniform random number parameters.

In Figure 11, the If action subsystem block is dedicated to picking and dropping 0.5kg boxes, encompassing a 90° rotation. Simultaneously, Figure 12 represents the If action subsystem block tailored for picking and dropping 1kg boxes, involving a 180° rotation. Each If action subsystem block is composed of essential components, including its own DC motor subsystem block, PID controller subsystem block, profile velocity block, and an action port.

Within these subsystem blocks, the connections are meticulously structured: the desired profile velocity serves as the input for the PID controller, and feedback signals (w,x) from the DC motor are fed back into the PID controller. The adjusted velocity from the PID controller then acts as the input for the DC motor, ensuring a closed-loop control system.

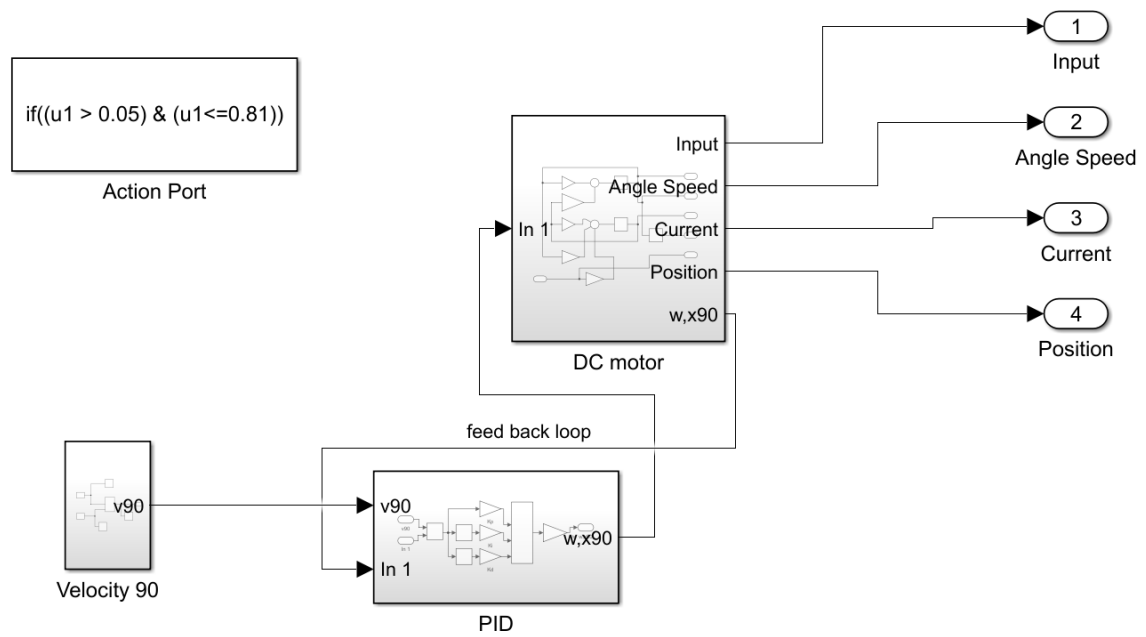


Figure 11 If action subsystem block for picking and dropping 0.5kg boxes (including rotating 90°).

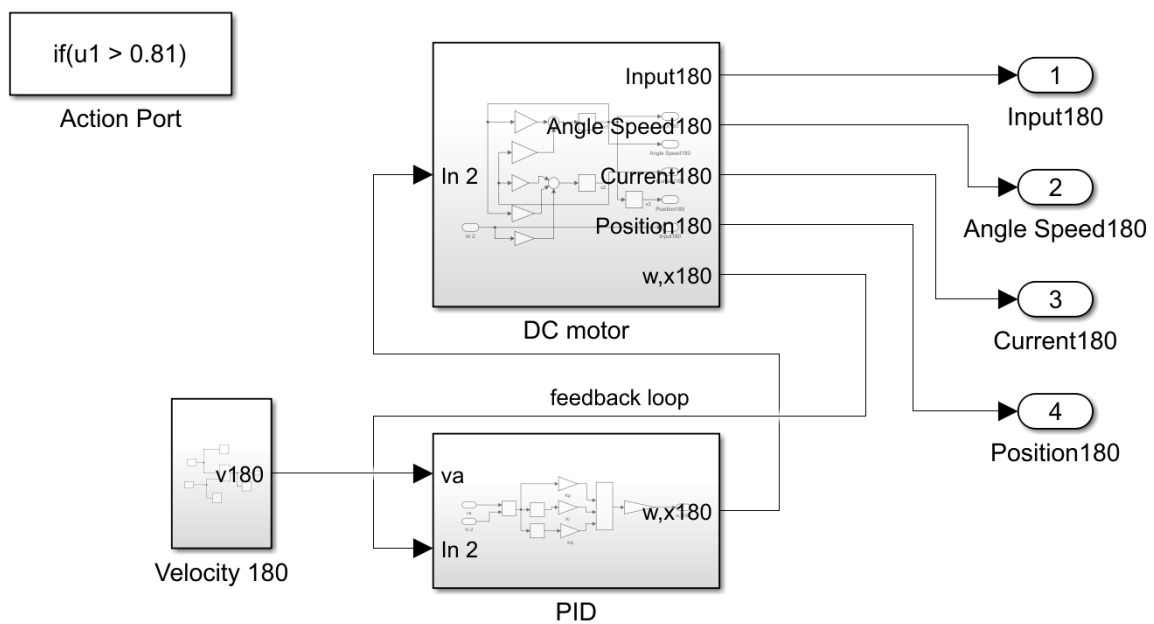


Figure 12 If action subsystem block for picking and dropping 1kg boxes (including rotating 180°).

1.4 The Overall System event driven and Continuous PID System.

To simulate the comprehensive behaviour of the system, which includes both event-driven processes and continuous PID-controlled operations, the reference velocities are combined using two repeating table blocks for each event. Figure 13 illustrates the velocity profile for the picking and dropping. Additionally, Figures 14 and 15 specify the velocity profiles for rotating 90° and rotating 180° , respectively. These profiles serve as crucial inputs to guide the system's movements.

Figure 16 presents the Simulink scheme for the velocity combination. In this scheme, the repeating table blocks are strategically utilized to superpose the reference velocities associated with different events.

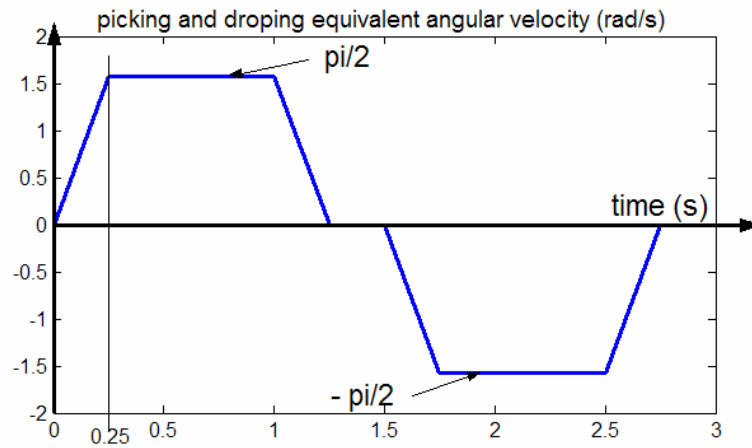


Figure 13 Velocity profile for picking and dropping.

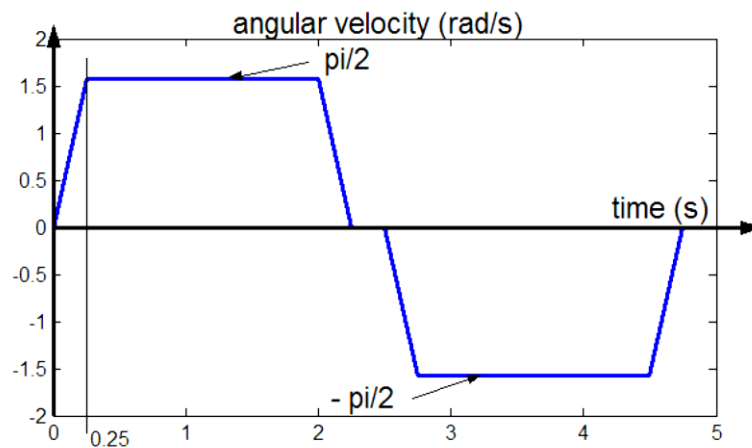


Figure 14 Velocity profile for rotating 90° .

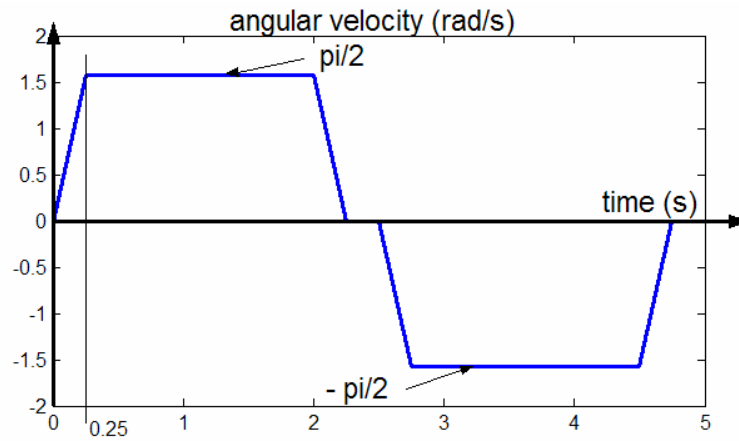


Figure 15 Velocity profile for rotating 180°.

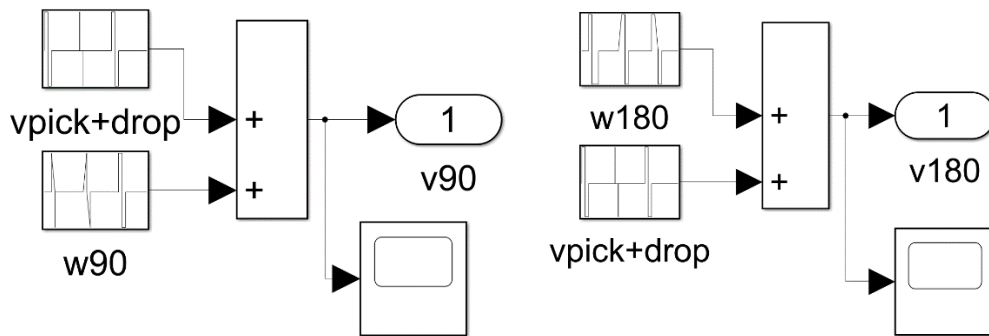


Figure 16 Simulink scheme of velocity combination.

With this sequence, we can superimpose these velocity profiles to obtain the overall profile, as illustrated in Figure 17. v_{90} is composed of the velocity profile for picking and dropping during time=0-3 sec, and the velocity profile for rotating 90° during time=3-6 sec. Meanwhile, v_{180} is composed of the velocity profile for picking and dropping during time=0-3 sec, and the velocity profile for rotating 180° during time=3-8 sec.

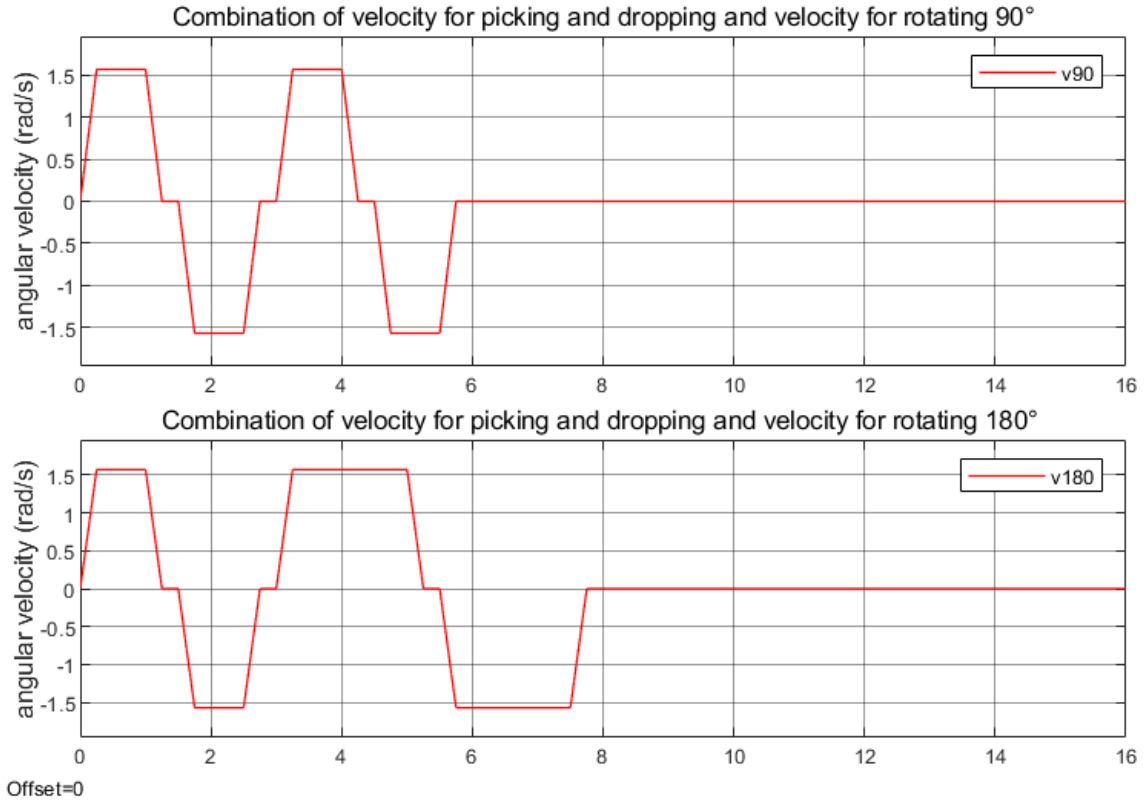


Figure 17 Combination of velocity profiles.

Figures 18, 21, and 23 illustrate the velocity response, while Figures 19, 20, 22, and 24 depict the current response under varying proportional (K_p), integral (K_i), and derivative (K_d) parameters. Each simulation modifies only one parameter at a time for clarity.

K_p is a crucial parameter, evident in Figure 18 and Figure 19. When K_p is set to 0, the angular velocity (at $t = 3.8023$ seconds) deviates by 0.7% from the desired value ($\pi/2$), and armature current spikes occur. Increasing K_p to 3 provides accurate angular velocity but introduces spikes in armature current, as shown in Figure 19. As indicated in part 1.2, increasing K_p from 0.3 to 3 enhances the step response's rise time. However, testing with reference velocity reveals that $K_p = 3$ leads to more oscillations in armature current compared to $K_p = 0.3$, as depicted in Figure 20.

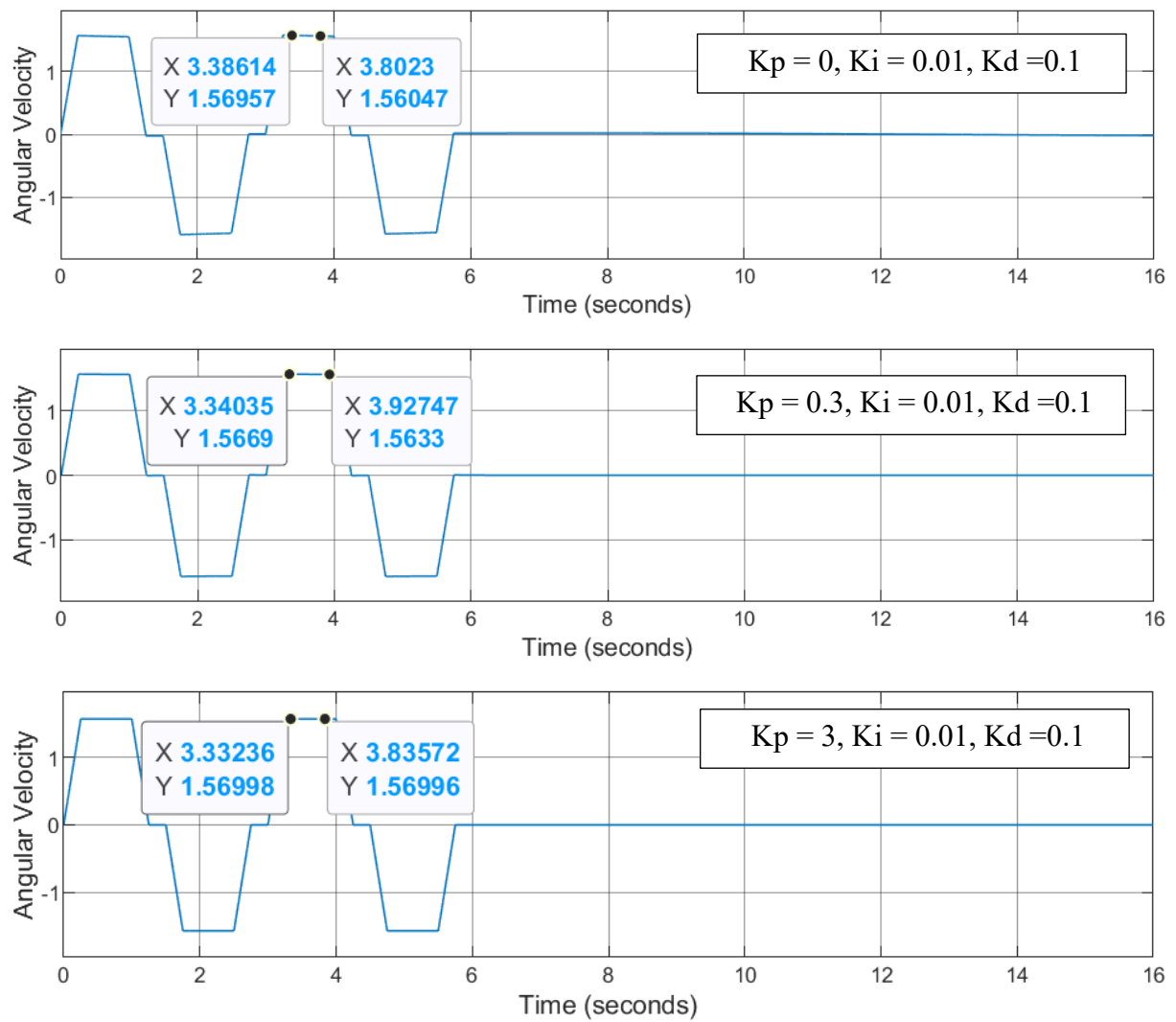


Figure 18 Velocity response compared by K_p .

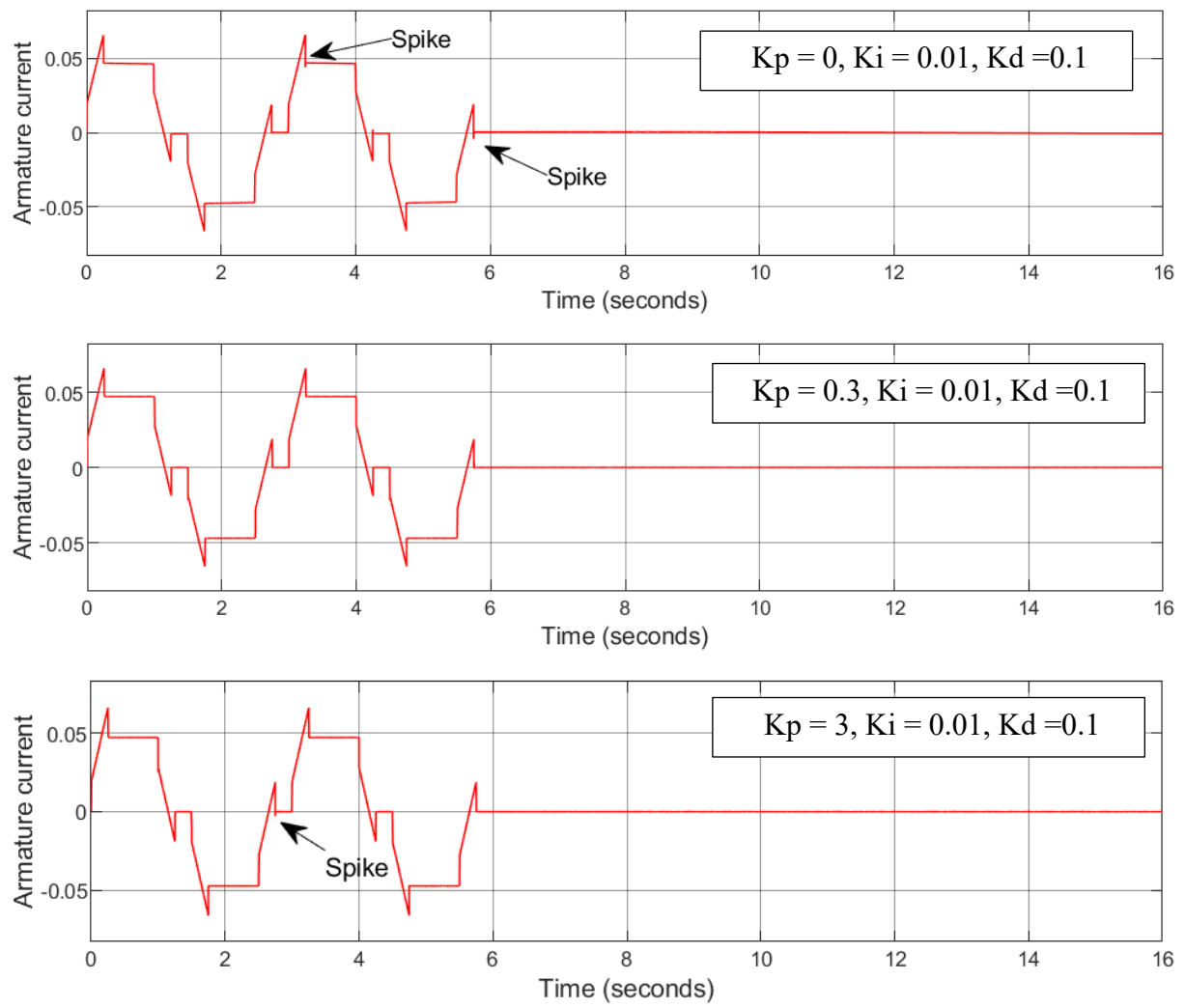


Figure 19 Current response compared by K_p .

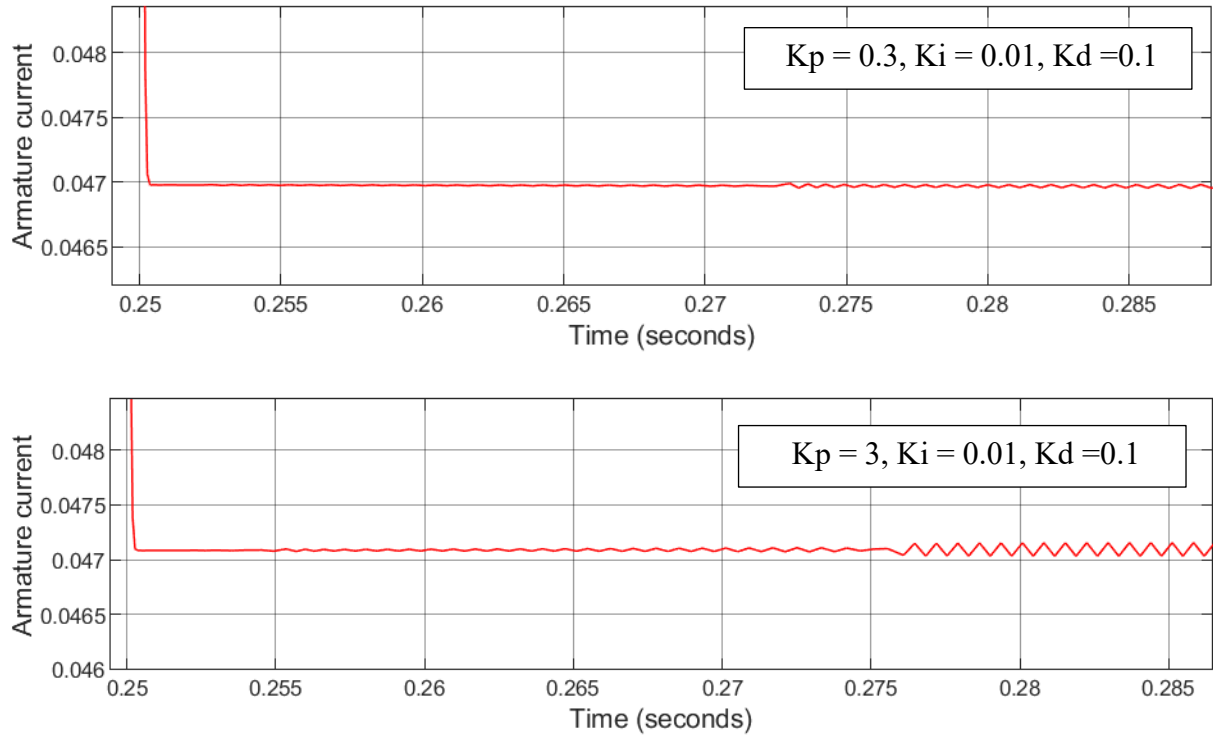


Figure 20 Current response compared by K_p .

Figure 21 shows that at K_i values of 0, 0.01, 1, or 3, the angular velocity error consistently remains below 0.3%, indicating a favorable performance. Increasing K_i leads to more precise velocity, akin to the step response where $K_i = 0.1$ or $K_i = 1$ exhibits faster rise times than $K_i = 0.01$. However, testing with profile velocity in Figure 22 reveals spikes at time = 0.25 seconds and time = 1 second for $K_i = 1$, presenting an undesirable characteristic. Additionally, without K_i , armature current still exhibits spikes.

In Figure 23, setting the derivative parameter (K_d) to 0 results in an acceptable angular velocity error of around 0.5%. However, the armature current deviates from the desired velocity, exhibiting oscillations throughout the waveform. This behavior aligns with observations in the step response, indicating it's unsuitable for the control system. Increasing K_d improves angular velocity precision, but spikes appear at $K_d = 0.2$ and $K_d = 0.5$, as depicted in Figure 24.

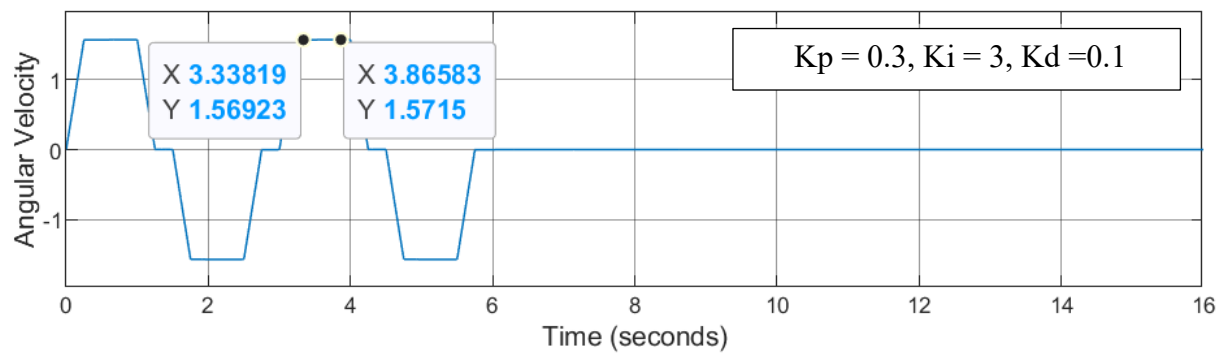
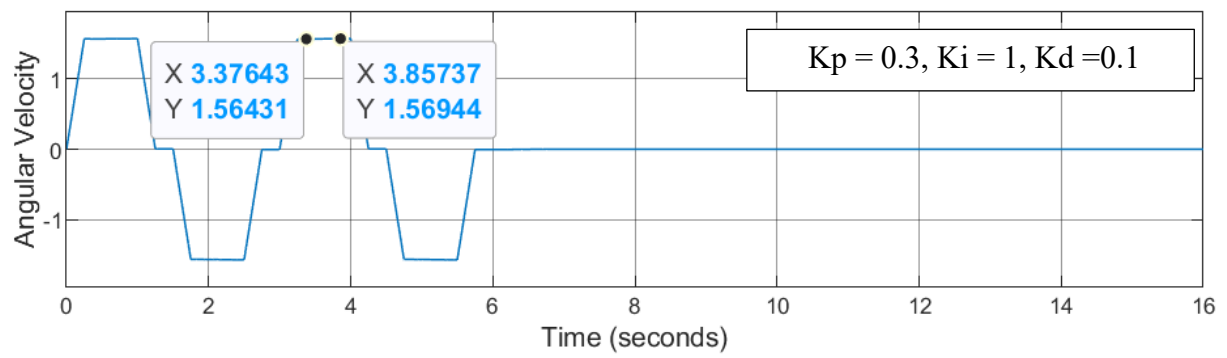
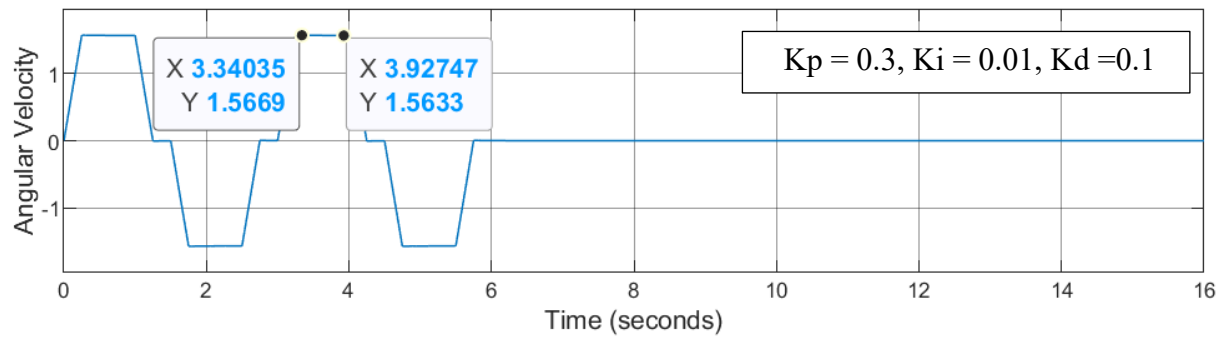
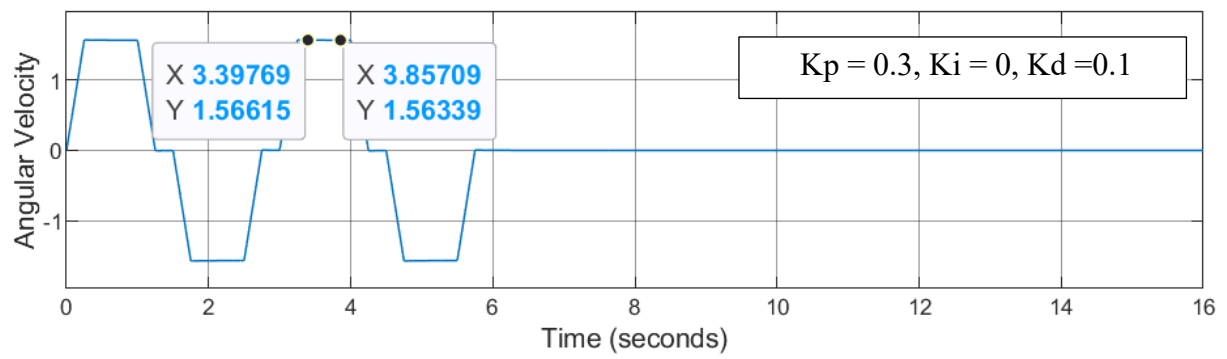


Figure 21 Velocity response compared by K_i .

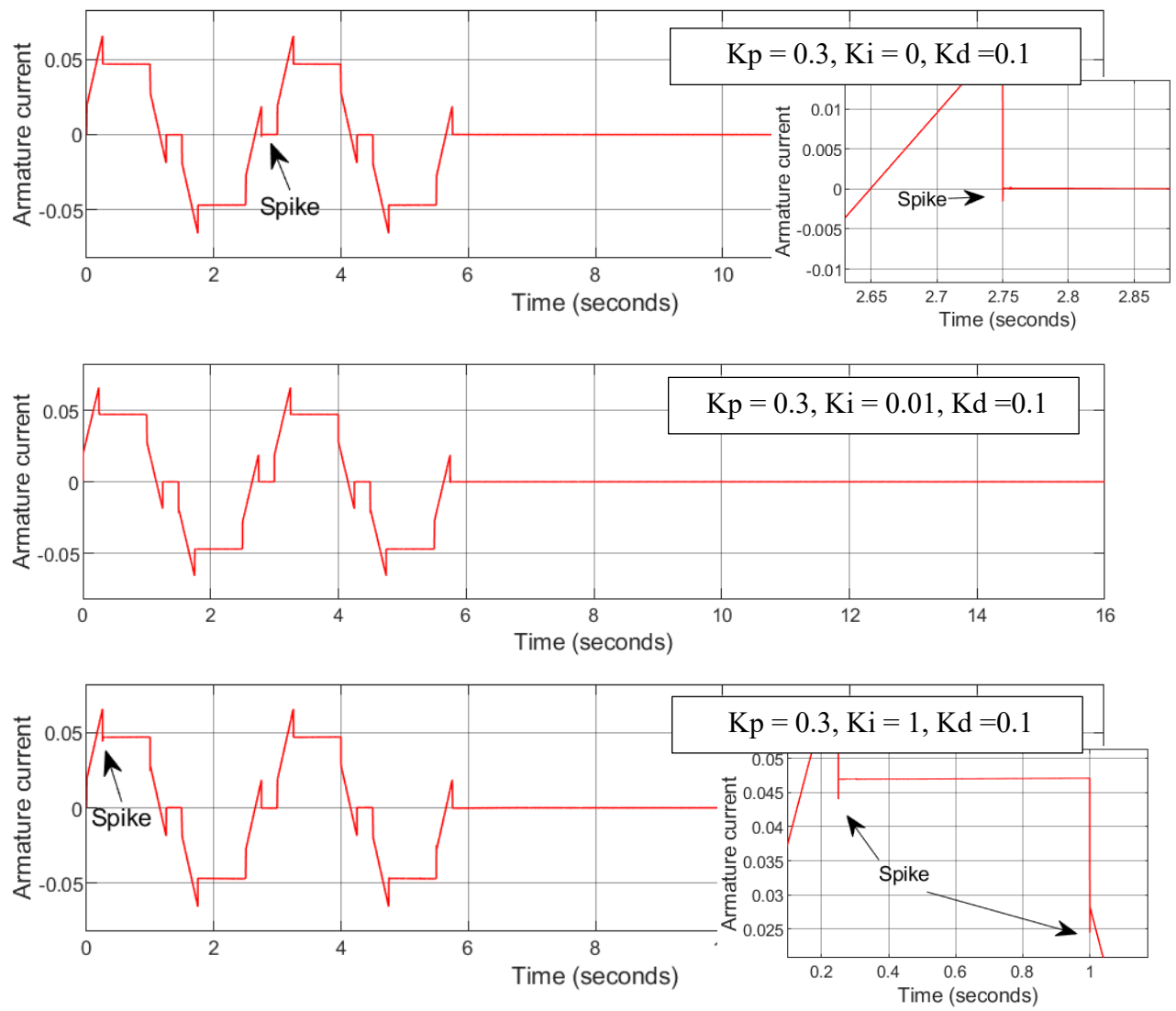


Figure 22 Current response compared by K_i .

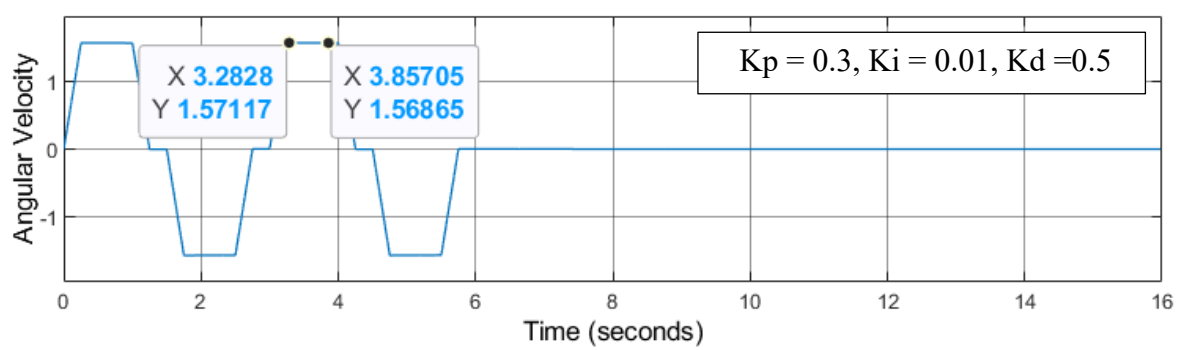
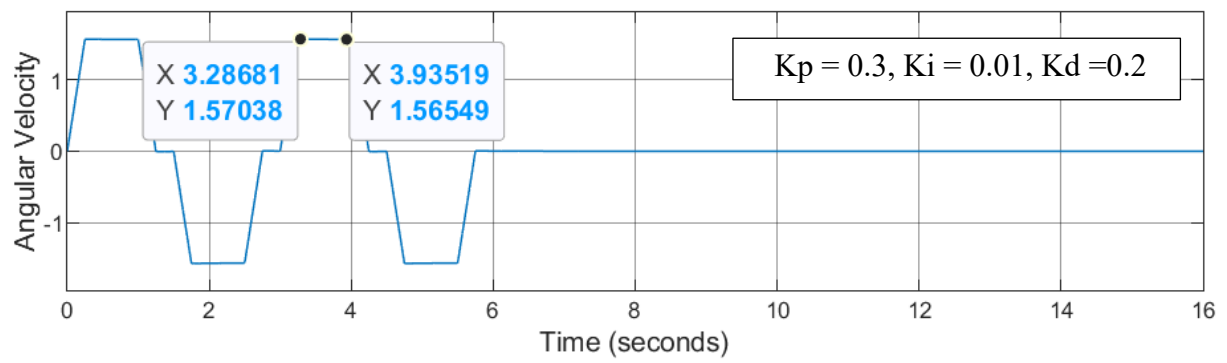
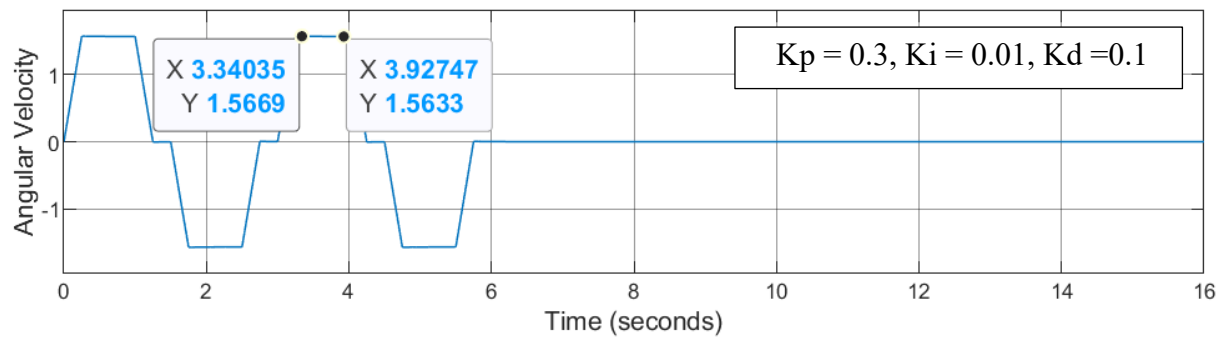
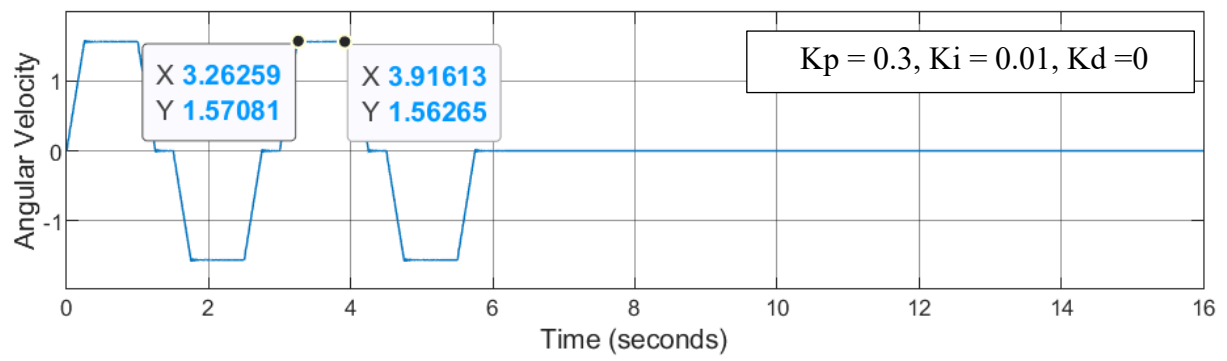


Figure 23 Velocity response compared by K_d .

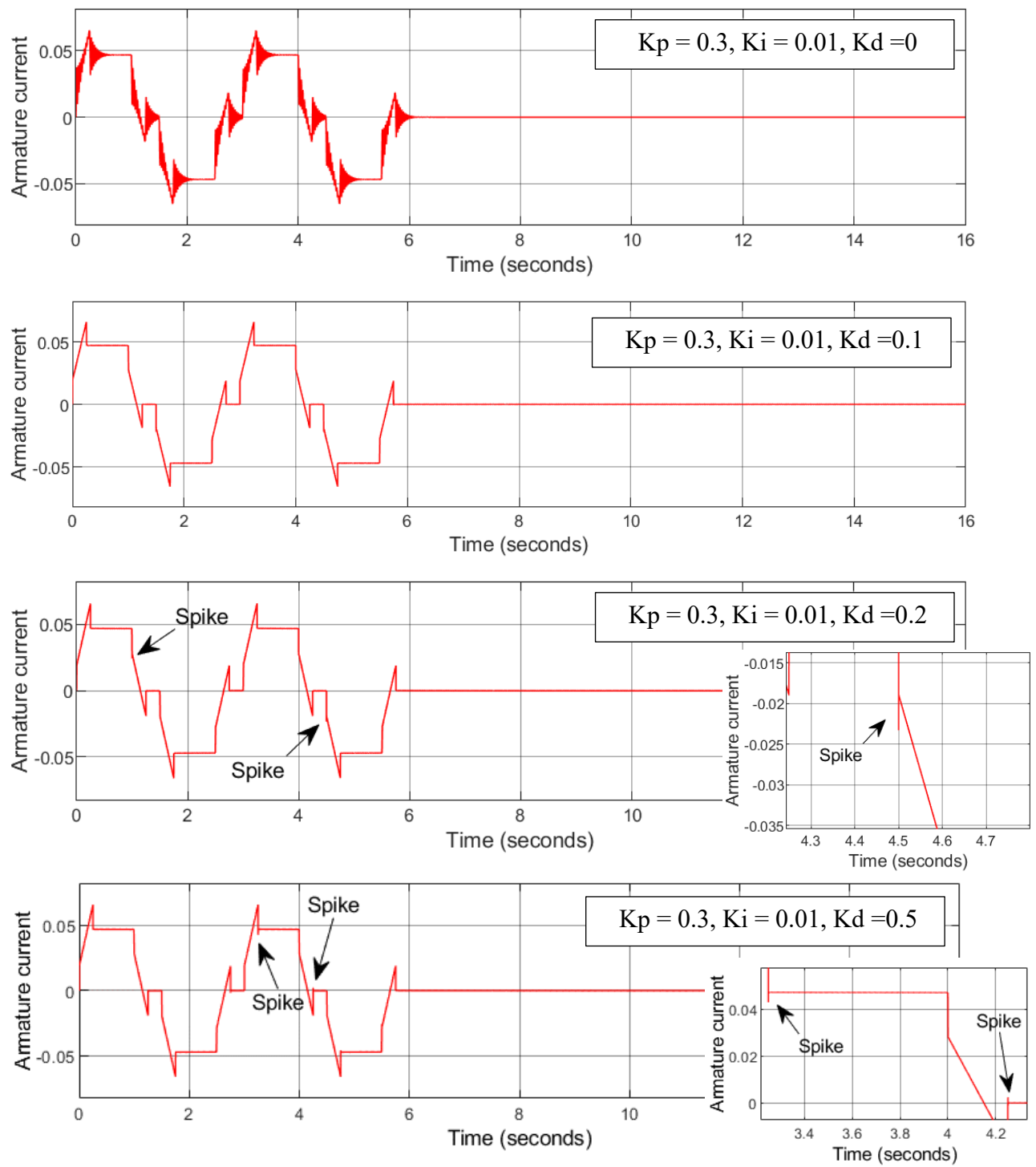


Figure 24 Current response compared by K_d .

Figure 25 showcases the angular velocity, and armature current dynamics during the 16-32 second interval for the 0.5 kg box occurrence, while Figure 26 illustrates the corresponding parameters for the 1 kg box instance observed between 80 and 96 seconds. It is apparent that the measured angular velocity for both scenarios aligns seamlessly with the overall profile, displaying no spikes or oscillations. Moreover, the armature current follows a similar smooth pattern as the velocity, underscoring its compatibility with the system.

Figures 27-29 depict the input, angular velocity, and armature current of a 0.5kg box, a 1kg box, and a missing box event when the seed is set to 16. In contrast, Figures 30-32 represent the same parameters for events with the seed set to 15. The results reveal that different seeds produce distinct event sequences. Specifically, during the first 16 seconds of the seed 15 event, a 0.5 kg box appears, followed by a 1 kg box in the subsequent 16 seconds. Conversely, in the sequence generated by seed 16, four boxes of 0.5 kg each arrive before a 1 kg box appears at time = 64 seconds.

Figures 29 and 32 show no current and velocity readings during missing box events, which is expected as there's no box for pickup. Table 1 displays the number of events for various seed values over 320 seconds, with an expected 20 events due to a box arriving every 16 seconds. Seeds 15 and 16 accurately match the expected number of boxes, although the average percentage of events across all seeds doesn't meet the desired conditions.

Table 1 Number of events in 320 seconds for various seed values.

seed		0.5 kg box	1 kg box	missing box
1		18	0	2
2		12	5	3
3		18	1	1
4		14	4	2
5		17	2	1
6		15	3	2
7		13	2	5
8		17	2	1
9		13	4	3
10		17	2	1
11		14	4	2
12		17	2	1
13		15	3	2
14		14	2	4
15		15	4	1
16		15	4	1
		76.25%	13.75%	10.00%

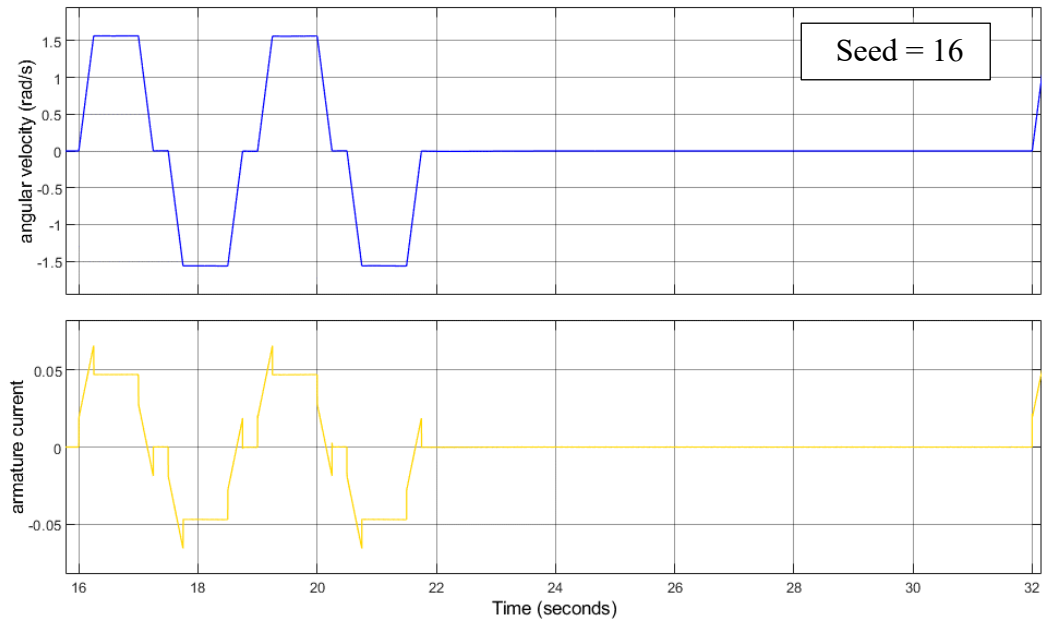


Figure 25 Experimental analysis of 0.5 kg box event: seed 16 (Time: 16 - 32 seconds)

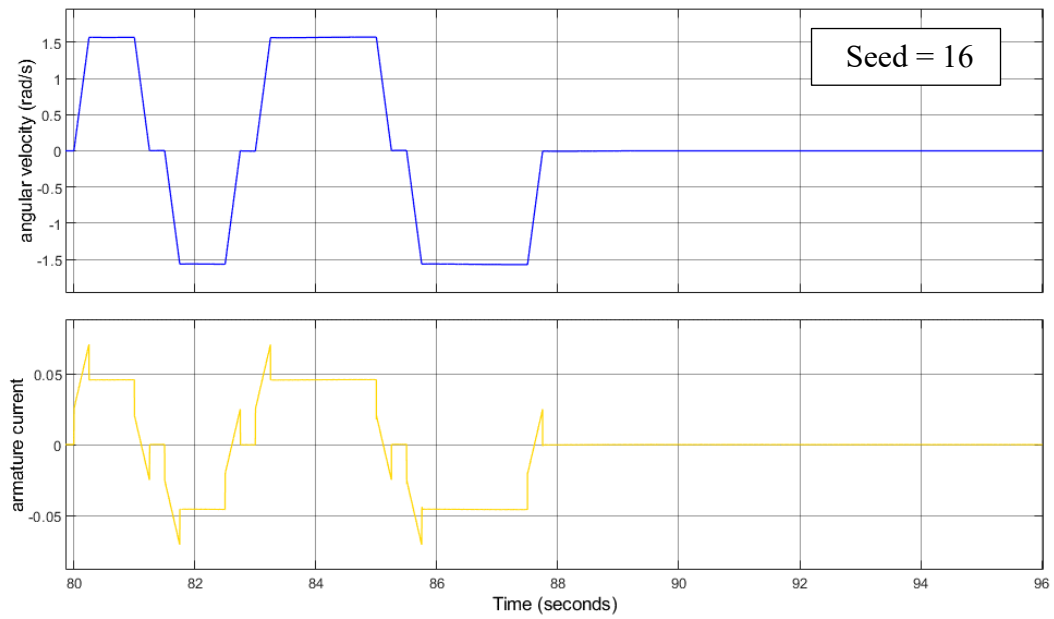


Figure 26 Experimental analysis of 1 kg box event: seed 16 (Time: 80 - 96 seconds).

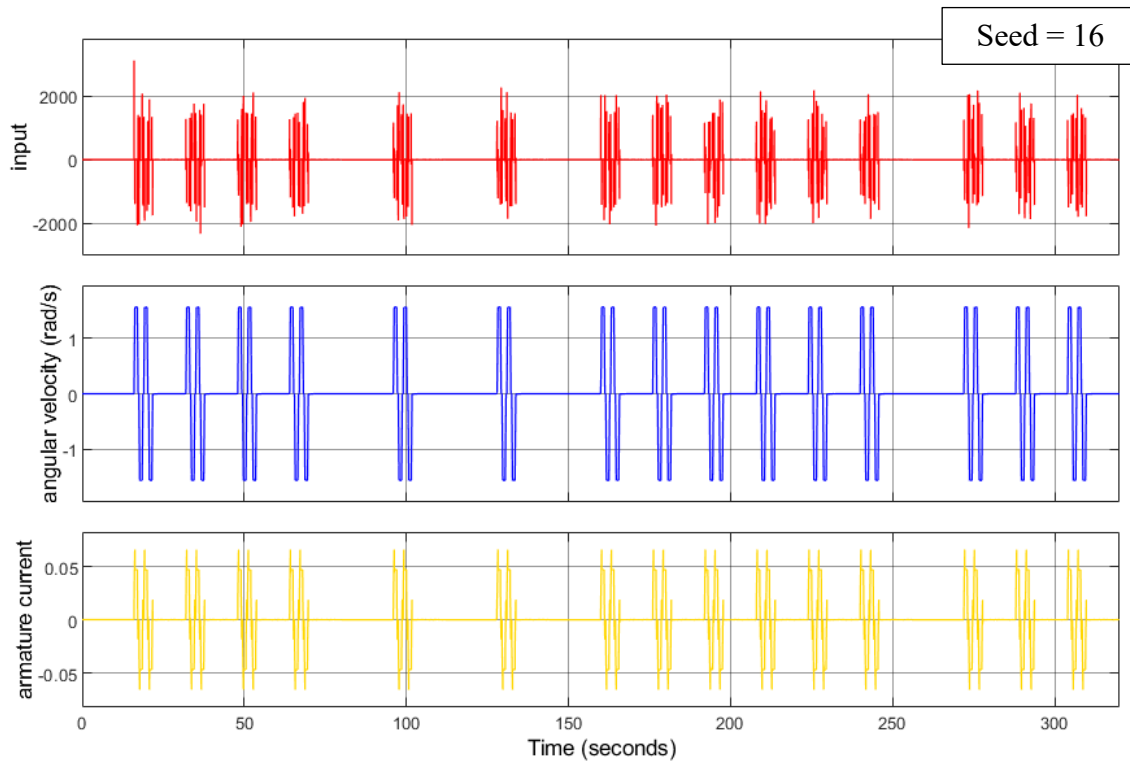


Figure 27 Experimental analysis of 0.5 kg box event: seed 16 (Time: 0 - 300 seconds).

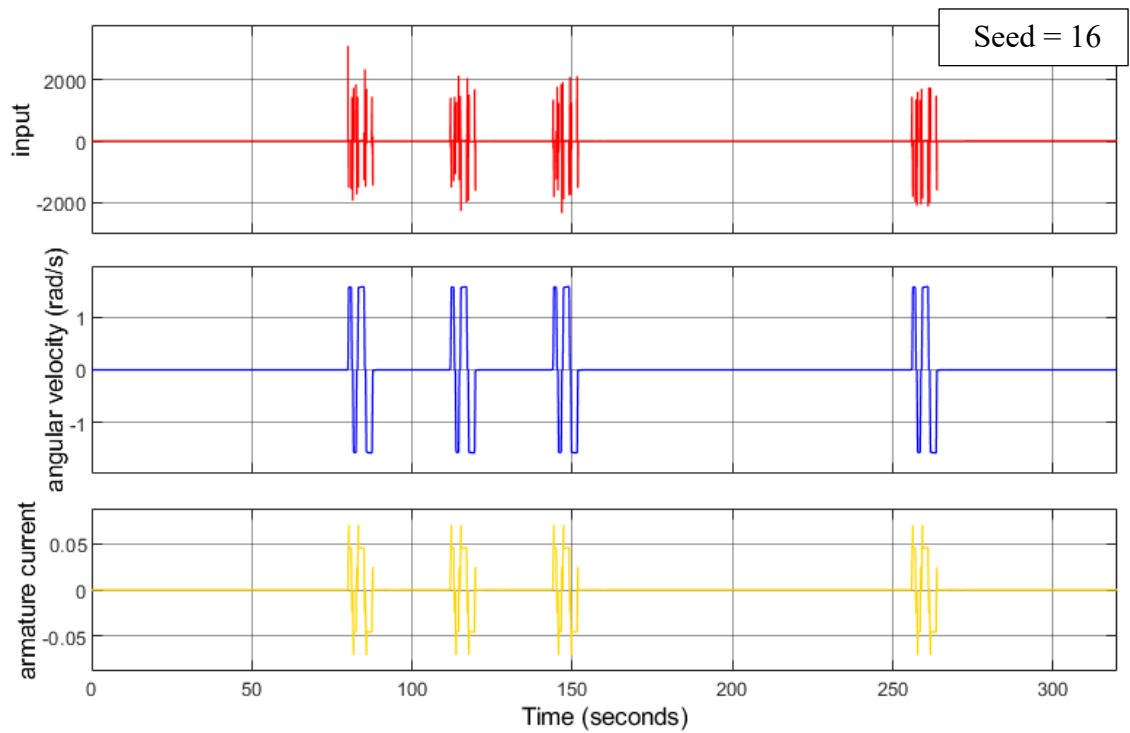


Figure 28 Experimental analysis of 1 kg box event: seed 16 (Time: 0 - 300 seconds).

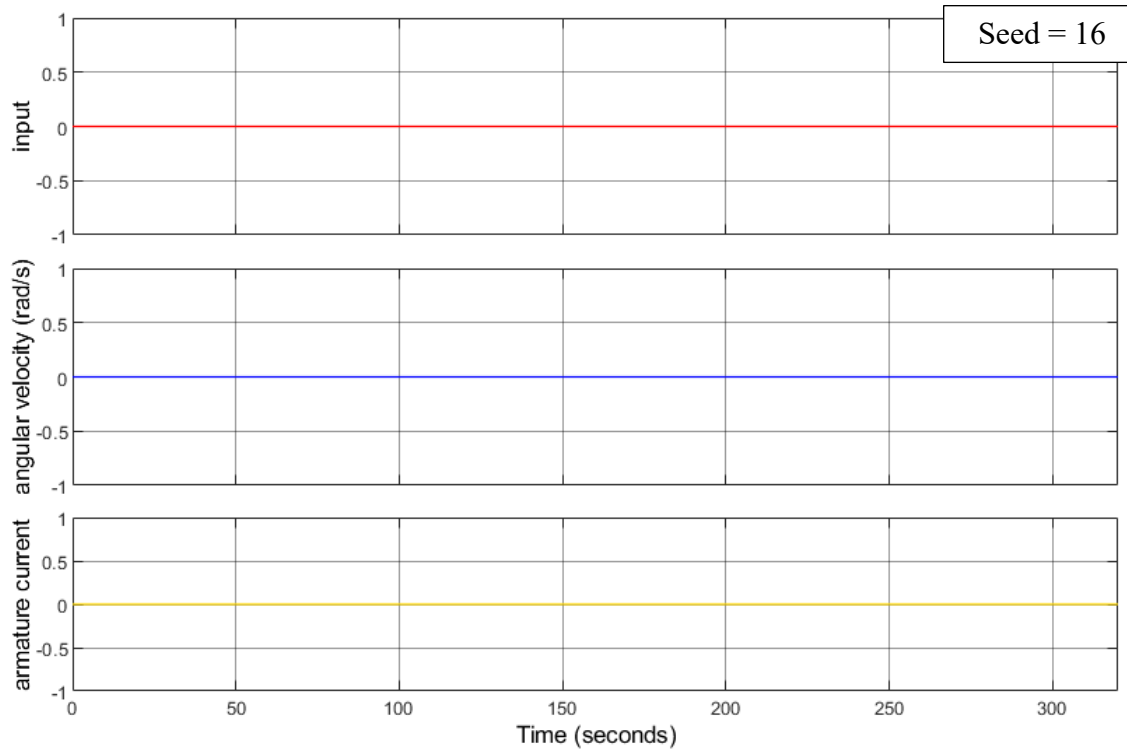


Figure 29 Experimental analysis of the missing box event: seed 16 (Time: 0 - 300 seconds).

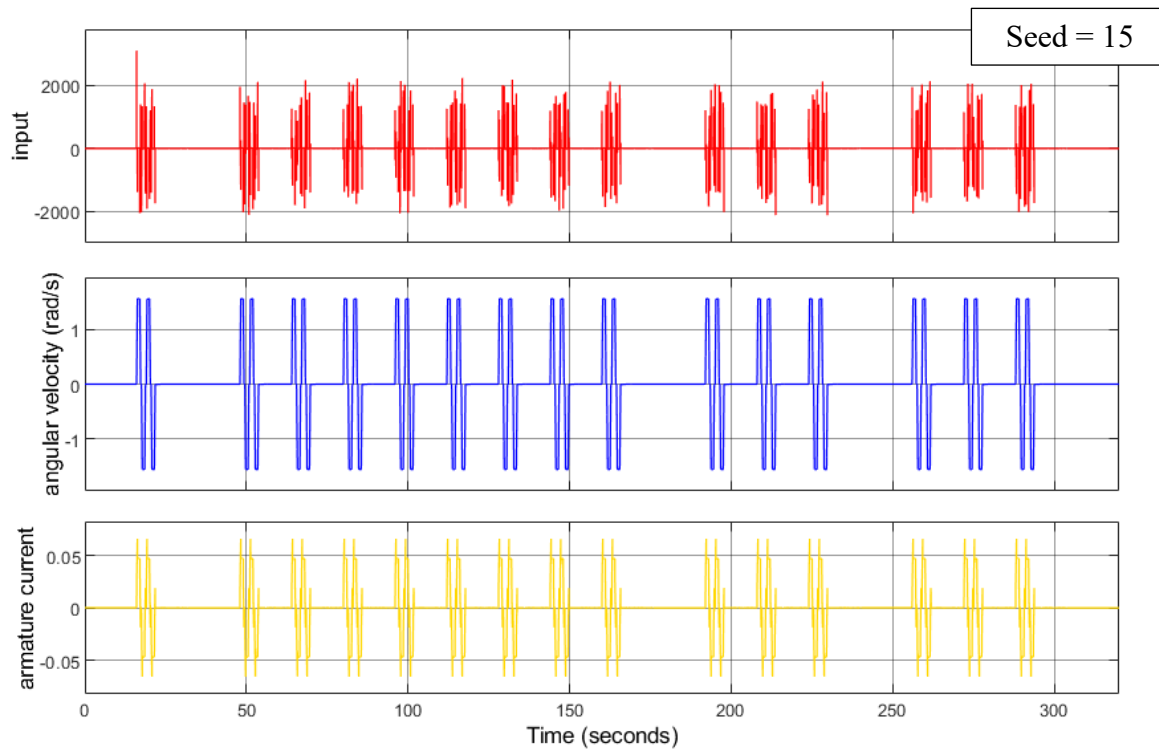


Figure 30 Experimental analysis of 0.5 kg box event: seed 15 (Time: 0 - 300 seconds).

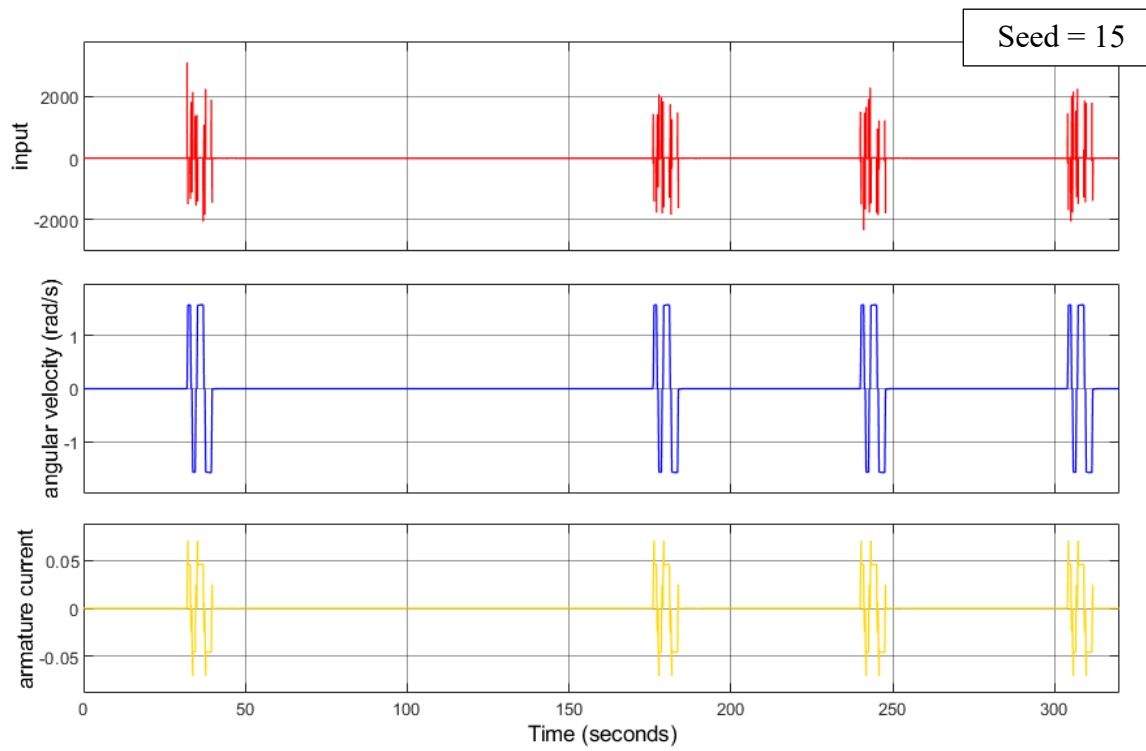


Figure 31 Experimental analysis of 1 kg box event: seed 15 (Time: 0 - 300 seconds).

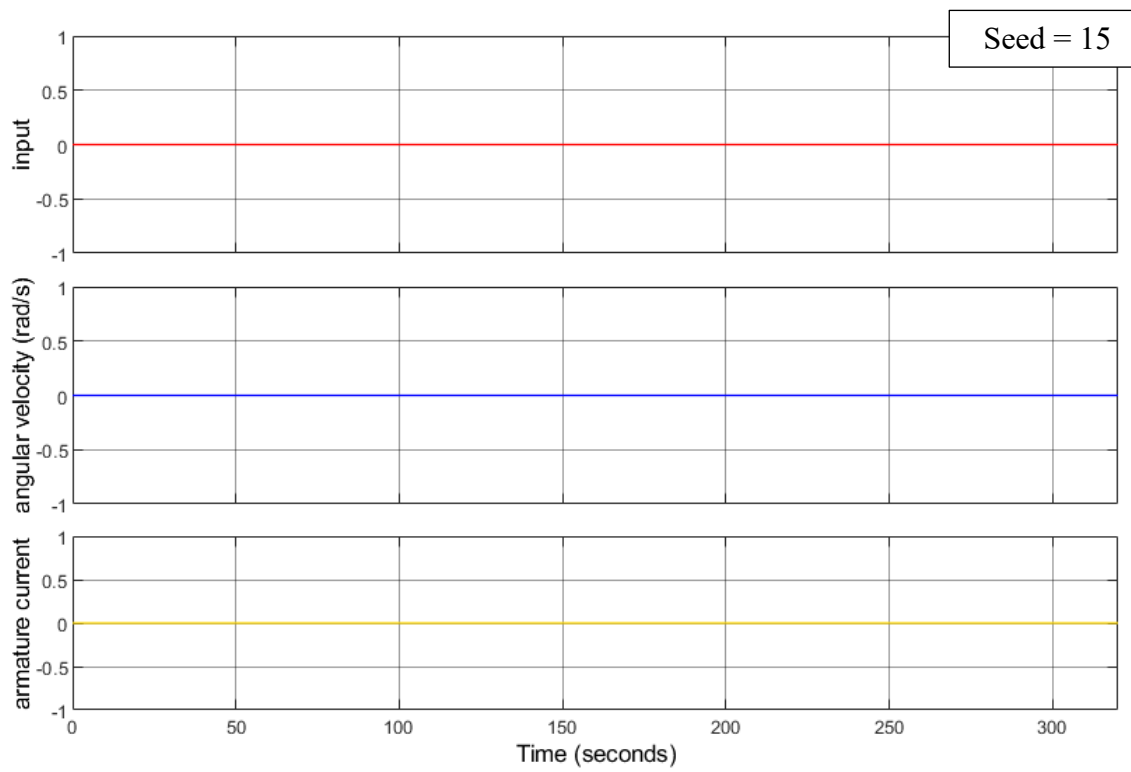


Figure 32 Experimental analysis of the missing box event: seed 15 (Time: 0 - 300 seconds).

2. Model Reference Adaptive Control for a DC Electrical Drive.

In control systems engineering, Model Reference Adaptive Control (MRAC) optimizes physical systems like DC motors. MRAC uses an idealized motor as a reference model, mirroring desired behaviour. This model guides a customized control law for the actual motor, aiming to minimize the error between their outputs. Through adept adjustments, the system strives for negligible error, aligning the controlled motor's output seamlessly with the desired behaviour. This adaptive approach aids in navigating uncertainties, variations, and unknown parameters, ensuring precise and responsive control in the dynamic realm of DC motor systems.

2.1 Model and Design of the MIT-based DC motor controller and Lyapunov-based DC motor controller.

In this system as shown in Figure 33, the reference signal (r) serves as the desired input, guiding the adjustable input (u) of the plant, representing the DC motor. The system aims to make the actual output (y) closely track the reference output (y_m), which embodies the desired behaviour of the DC motor. The control is implemented through two distinct laws: the MIT rule and Lyapunov stability theory, each providing strategies for adjusting the controller's parameters [1]. The variables r , u , y_m , and y play crucial roles in achieving precise control, with the reference model serving as a benchmark. The dynamic interaction between these elements is orchestrated to minimize the disparity between the actual and desired outputs, ensuring the effective performance of the DC motor under varying conditions. The Simulink scheme of plant and reference model is Figure 34.

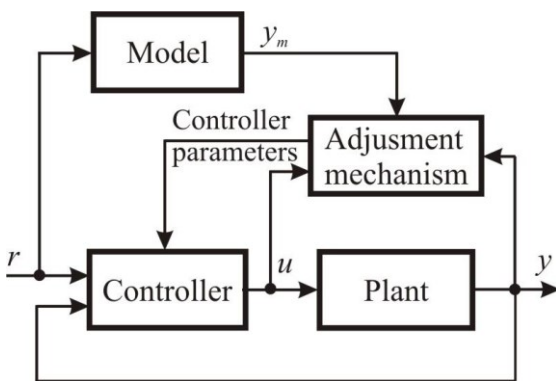


Figure 33 Block diagram of MRAS [1].

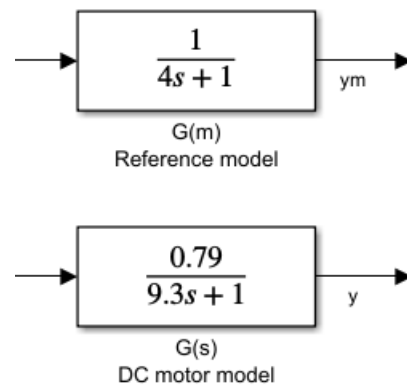


Figure 34 Simulink scheme of plant and reference model.

In this scenario, the DC motor model is simplified to a first-order ordinary differential equation (ODE)

$$\dot{y} = -ay(t) + bu(t) \quad (2.1)$$

Employing the Laplace transform, the corresponding transfer function for the model is derived as

$$G(s) = \frac{Y(s)}{U(s)} = \frac{b}{s + a} = \frac{0.79}{9.3s + 1} \quad (2.2)$$

Simultaneously, the reference model for the DC motor is characterized by

$$\dot{y}_m = -a_my_m(t) + b_mr(t) \quad (2.3)$$

with its Laplace-transformed transfer function being

$$G(s) = \frac{Y(s)}{U(s)} = \frac{b}{s + a} = \frac{1}{4s + 1} \quad (2.4)$$

The control law is applied as

$$u(t) = t_0r(t) - s_0y(t) \quad (2.5)$$

leading to the definition of the error

$$e(t) = y(t) - y_m(t) \quad (2.6)$$

To apply the MIT rule, Equation (2.5) is introduced in Equation (2.3) as below

$$\begin{aligned} \dot{y} &= -ay(t) + bu(t) \\ py(t) &= -ay(t) + bu(t) \\ py(t) &= -ay(t) + bt_0r(t) - bs_0y(t) \\ (p + a(t) + bs_0)y(t) &= bt_0r(t) \\ y(t) &= \frac{bt_0}{p + a + bs_0}r(t) \end{aligned} \quad (2.7)$$

where p is the differential operator. Hence, the error will be

$$e(t) = \left(\frac{bt_0}{p + a + bs_0} - \frac{b_m}{p + a_m} \right) r(t) \quad (2.8)$$

The sensitivity derivatives are obtained by calculating the partial derivatives of the error with respect to the controller parameters.

$$\begin{aligned}\frac{\partial e(t)}{\partial t_0} &= \left(\frac{b}{p + a + bs_0} \right) r(t) \\ \frac{\partial e(t)}{\partial s_0} &= \left(\frac{b}{p + a + bs_0} \right) y(t)\end{aligned}\tag{2.9}$$

From the expression

$$\frac{d\theta}{dt} = -\gamma \frac{\partial J}{\partial \theta} = -\gamma e \frac{\partial e}{\partial \theta}\tag{2.10}$$

the parameters adjustment on the MIT rule can be obtained

$$\begin{aligned}\frac{dt_0(t)}{dt} &= -\gamma \left(\frac{1}{p + a_m} r(t) \right) e(t) \\ \frac{ds_0(t)}{dt} &= \gamma \left(\frac{1}{p + a_m} y(t) \right) e(t)\end{aligned}\tag{2.11}$$

where γ is a tunable parameter, and p is the differential operator.

To drive the parameters t_0 and s_0 for the Lyapunov theory, the Lyapunov function will be

$$\frac{dV(t)}{dt} = e \frac{de(t)}{dt} + \frac{1}{\gamma} (bs_0 + a - a_m) \frac{ds_0(t)}{dt} + \frac{1}{\gamma} (bt_0 - b_m) \frac{dt_0(t)}{dt}\tag{2.12}$$

The parameters are updated as

$$\begin{aligned}\frac{dt_0(t)}{dt} &= -\gamma r(t) e(t) \\ \frac{ds_0(t)}{dt} &= \gamma y(t) e(t)\end{aligned}\tag{2.13}$$

2.2 Results and Analysis of the Simulation.

A step input was applied to the simulated system shown in Figure 37 for 100 seconds. Figure 38 depicts the evolution of the adjustment error signal ($e(t)$) obtained through the MIT rule. Comparing this with the step response of the Lyapunov-based DC motor controller in Figure 39, it is observed that the Lyapunov stability theory results in less oscillation and faster settling time. Similarly, the error signals in Figure 40 show that the Lyapunov error signal experiences less oscillation than the MIT error signal. Additionally, it is noteworthy that the maximum error derived from the Lyapunov stability theory is less than the maximum error resulting from the MIT rule.

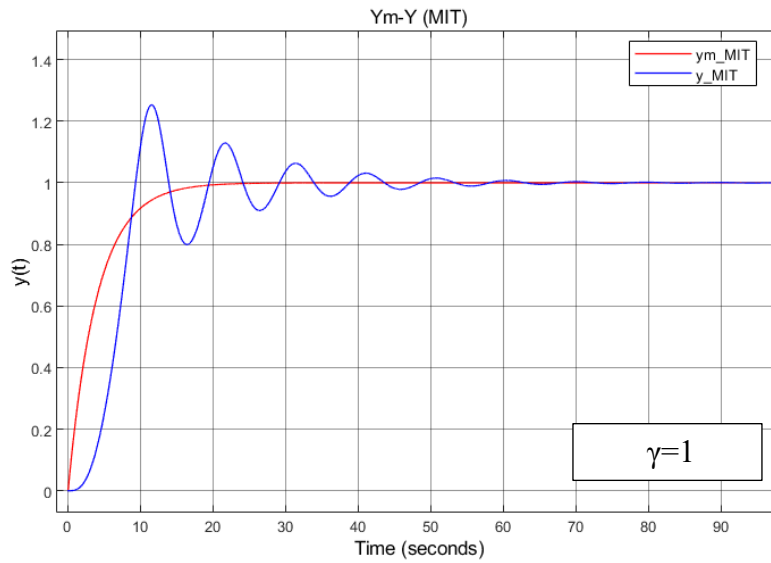


Figure 37 The output signals ($y(t)$, $y_m(t)$) obtained using the MIT rule.

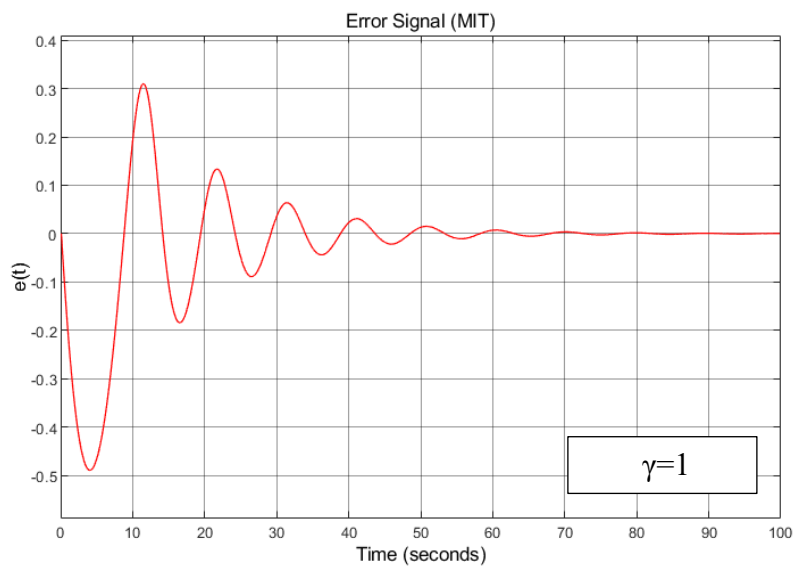


Figure 38 The adjustment error signal ($e(t)$) progression obtained using the MIT rule.

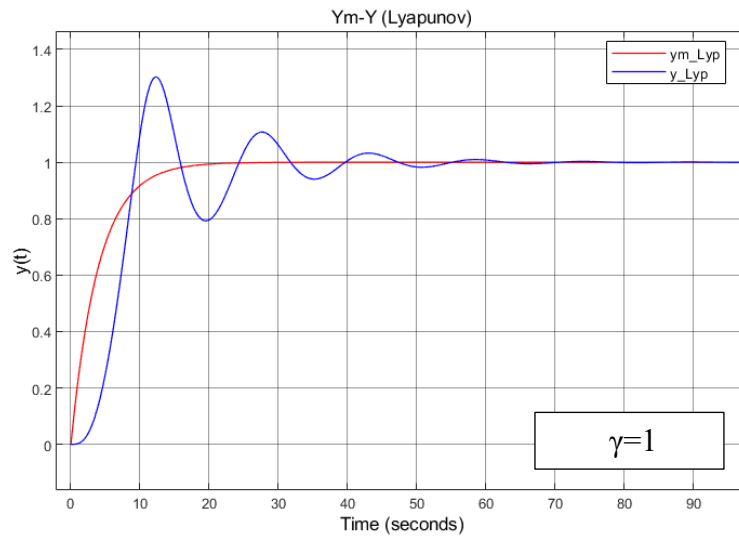


Figure 39 The output signals ($y(t)$, $y_m(t)$) obtained using the Lyapunov stability theory.

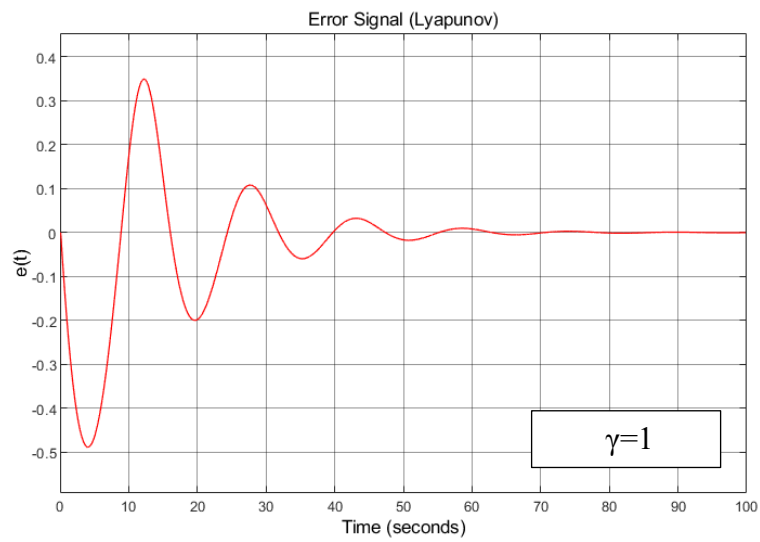


Figure 40 The adjustment error signal ($e(t)$) progression using the Lyapunov stability theory.

Because of

$$u(t) = t_0 r(t) - s_0 y(t)$$

$$r(t) = \frac{u(t) + s_0 y(t)}{t_0}$$

Therefore, we can replace $r(t)$ with above equation. Figure 38 illustrates Simulink scheme of MIT-based DC motor controller utilizing the expression $r(t) = \frac{u(t) + s_0 y(t)}{t_0}$ within adjustment mechanism subsystem block. This alternative simulation approach offers a nuanced perspective, allowing us to configure the system in a manner that aligns closely with the MRAS theoretical framework. Figure 41 compares the y and y_m outputs of the MIT-based DC motor controller Simulink scheme in Figure 35 with that of Figure 41. The results in Figure 42 show no differences, affirming the simulation to the theory.

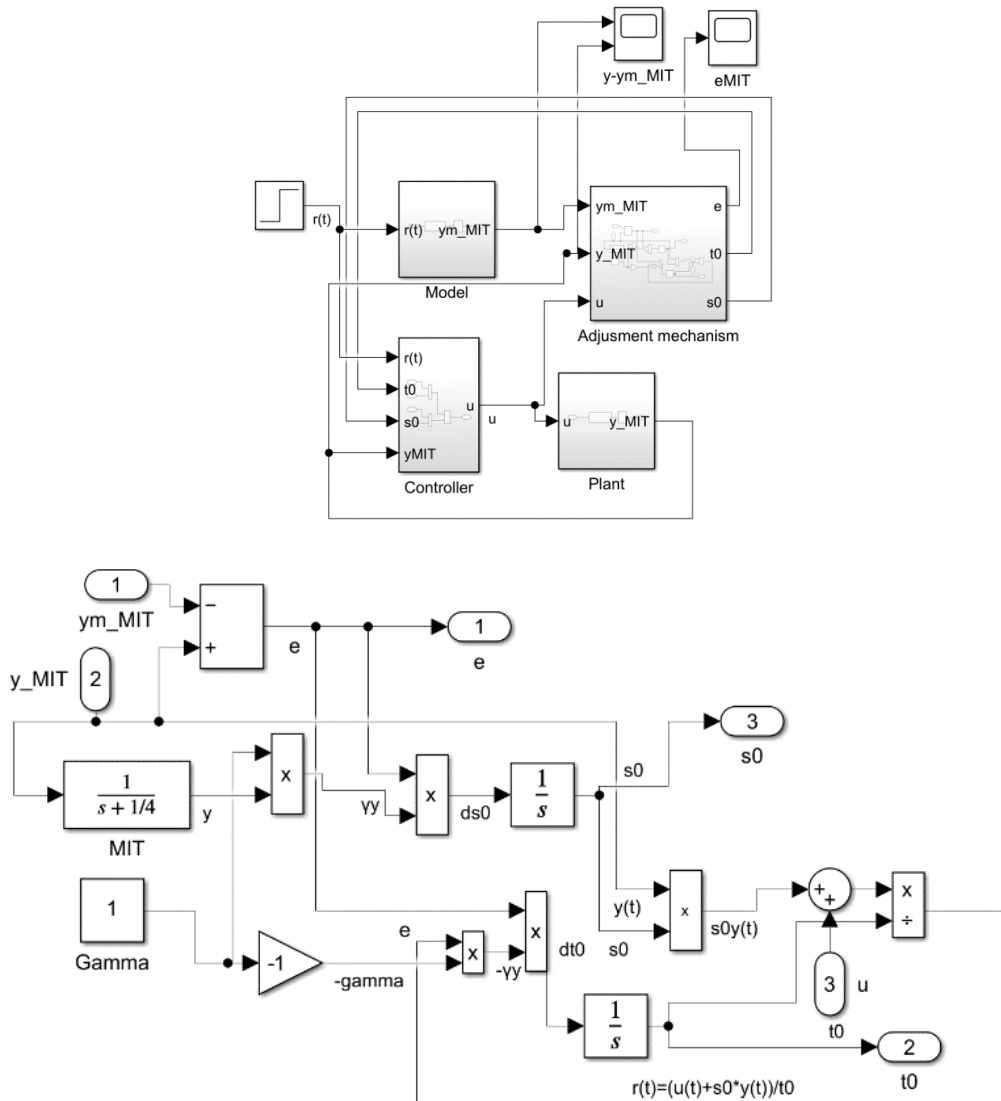


Figure 41 Simulink scheme of MIT-based DC motor controller with MRAS pattern.

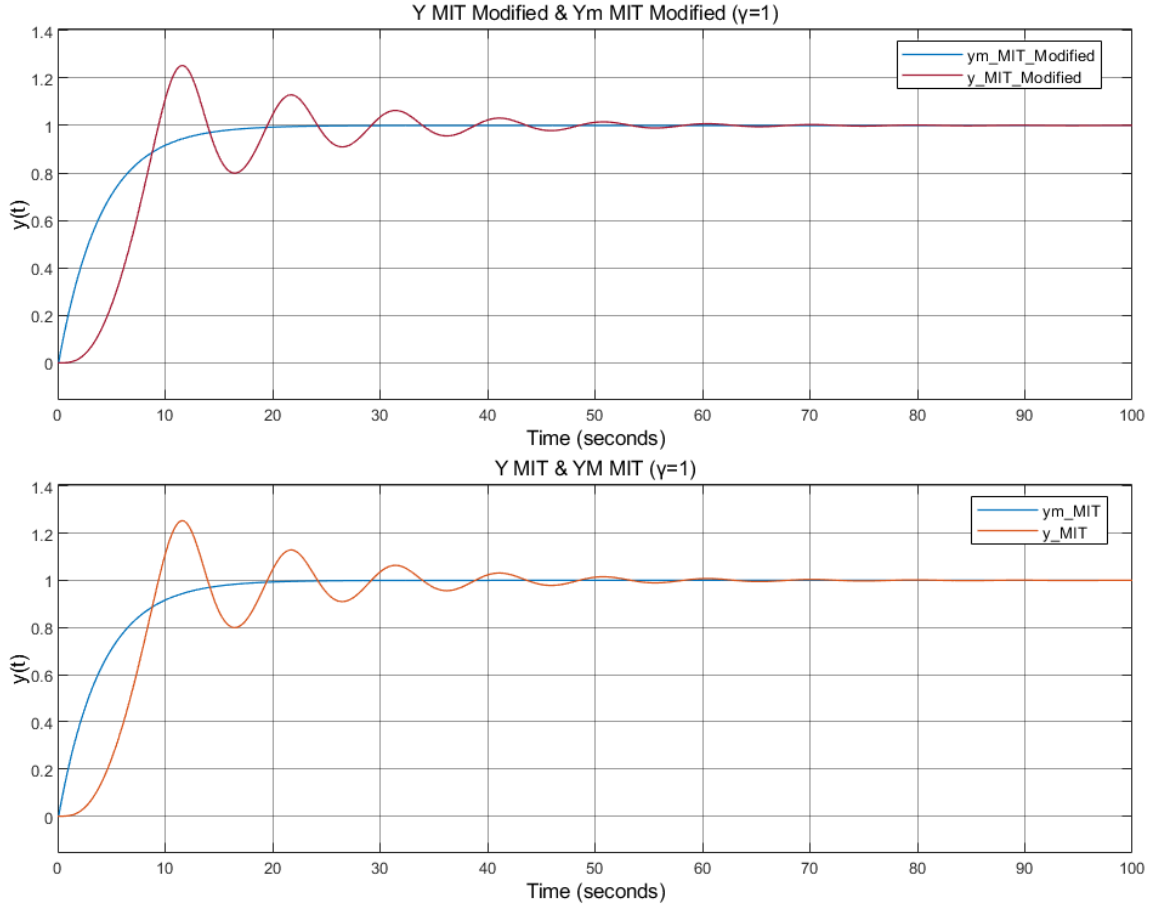


Figure 42 Performance evaluation: MIT rule vs. MIT rule with MRAS pattern in DC motor control

Figures 43, 44, 45, and 46 illustrate the performance of output and error progression employing both MIT rule and Lyapunov stability theory, with varying values of gamma (γ). Observations from the figures indicate that the larger value of gamma ($\gamma=3$ and $\gamma=5$) results in faster adaptation, while the smaller value of gamma ($\gamma=0.1$ and $\gamma=0.01$) leads to slower adaptation. Additionally, an increase in the value of gamma eventually leads to oscillations and, in some cases, instability.

The influence of the reference model becomes evident at $\gamma=3$, where the settling time of the Lyapunov output outshines that of the MIT output, although the overshoots exhibit minimal divergence. Upon increasing γ to 5, the Lyapunov output maintains a performance similar to that at $\gamma=3$, while the MIT output experiences heightened oscillations, and its settling time exceeds the simulated duration of 100 seconds. Similarly, with smaller γ values, the Lyapunov output demonstrates a quicker settling time compared to the MIT output for both γ values. Moreover, the overshoot in the Lyapunov output is marginally less than that in the MIT output.

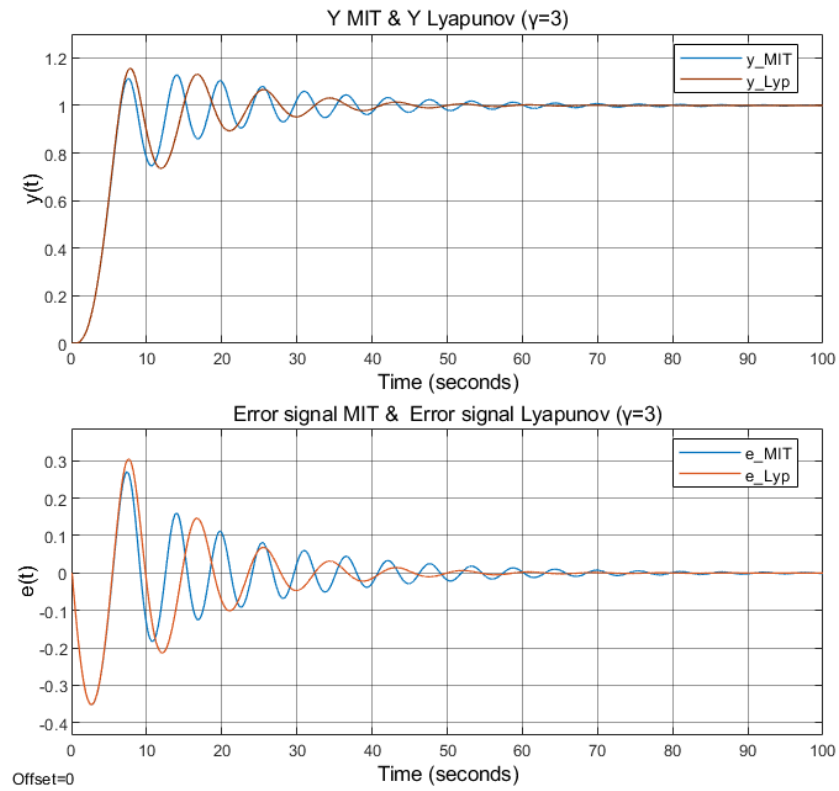


Figure 43 Performance evaluation: MIT rule vs. Lyapunov stability theory in DC motor control ($\gamma=3$).

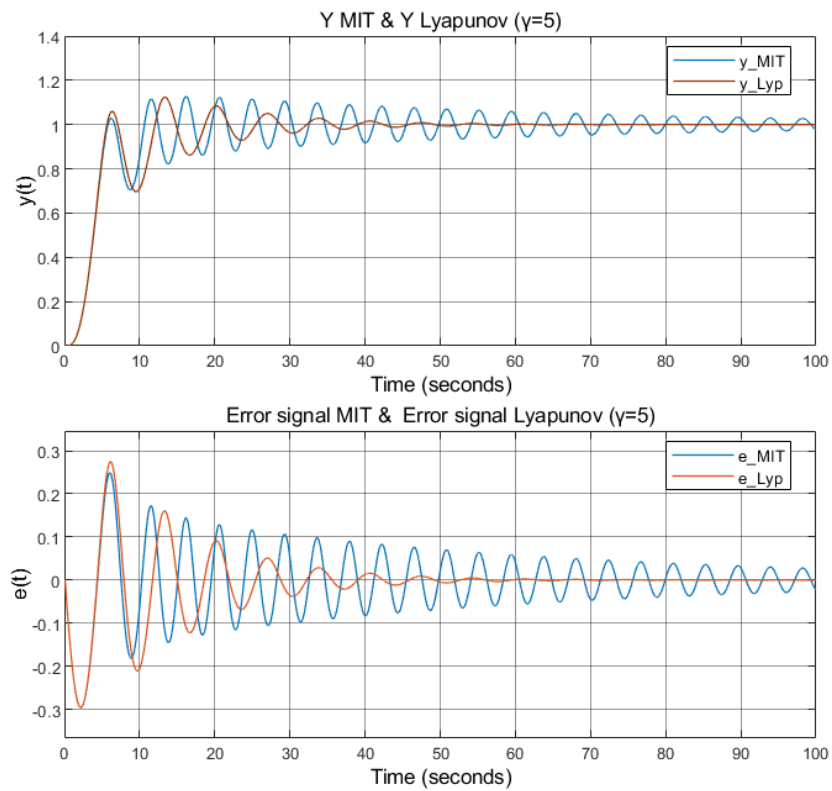


Figure 44 Performance evaluation: MIT rule vs. Lyapunov stability theory in DC motor control ($\gamma=5$).

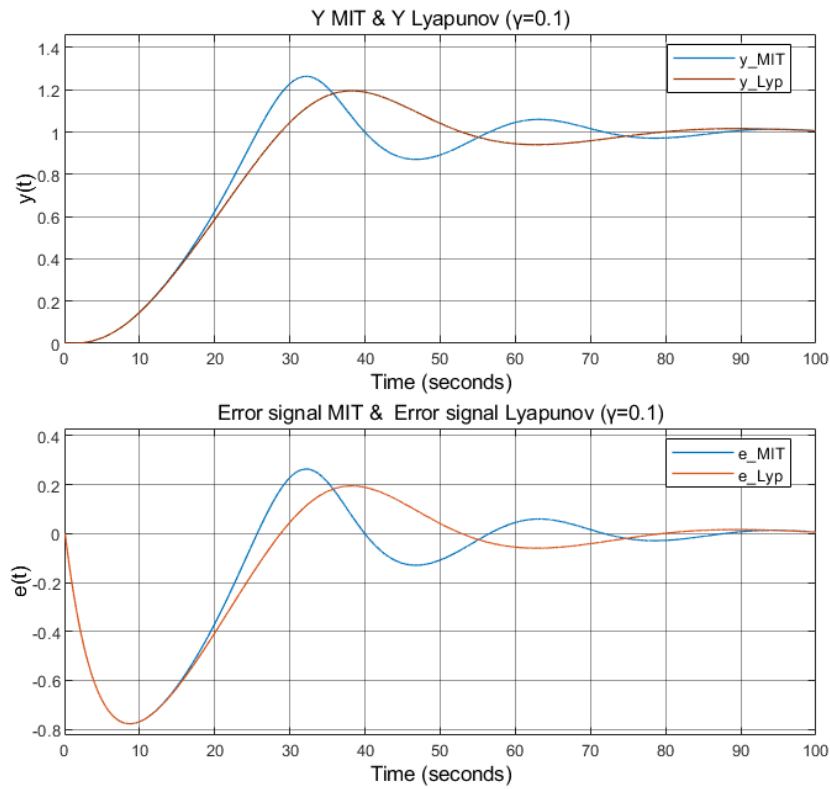


Figure 45 Performance evaluation: MIT rule vs. Lyapunov stability theory in DC motor control ($\gamma=0.1$).

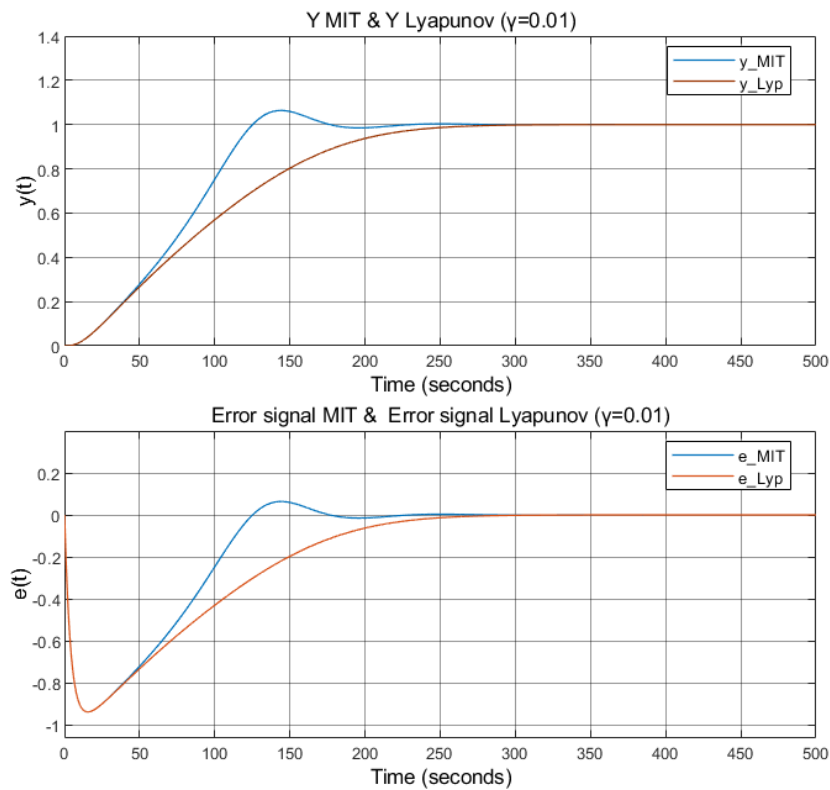


Figure 46 Performance evaluation: MIT rule vs. Lyapunov stability theory in DC motor control ($\gamma=0.01$).

2.3 Comparative Analysis of MIT and Lyapunov-Based Model Reference Adaptive Controllers.

The paper titled "Comparative Analysis of MIT Rule and Lyapunov Rule in Model Reference Adaptive Control Scheme" by Swarnkar Pankaj, Jain Shailendra Kumar, and Nema R.K presents a detailed comparison between MIT rule and Lyapunov rule in the context of MRAC for a first-order system. MIT rule is noted for its historical significance and automatic adjustment capability but is criticized for its sensitivity to adaptation gain. On the other hand, Lyapunov stability methods are recognized for their robustness but are considered more complex. The effectiveness of both methods depends on factors such as the choice of adaptation gains and system characteristics.

A noticeable trend in the study indicates that when the adaptation gain (α) is set to smaller values, the system response exhibits sluggish behaviour without oscillations. As α gradually increases, a trade-off becomes apparent: the maximum overshoot in the response grows, but concurrently, the system's speed improves, leading to a reduction in settling time. Interestingly, the analysis suggests that the enhancement in system performance is more rapid when employing the Lyapunov rule as the adaptation gain increases, as depicted in Figure 47 and Figure 48 [2].

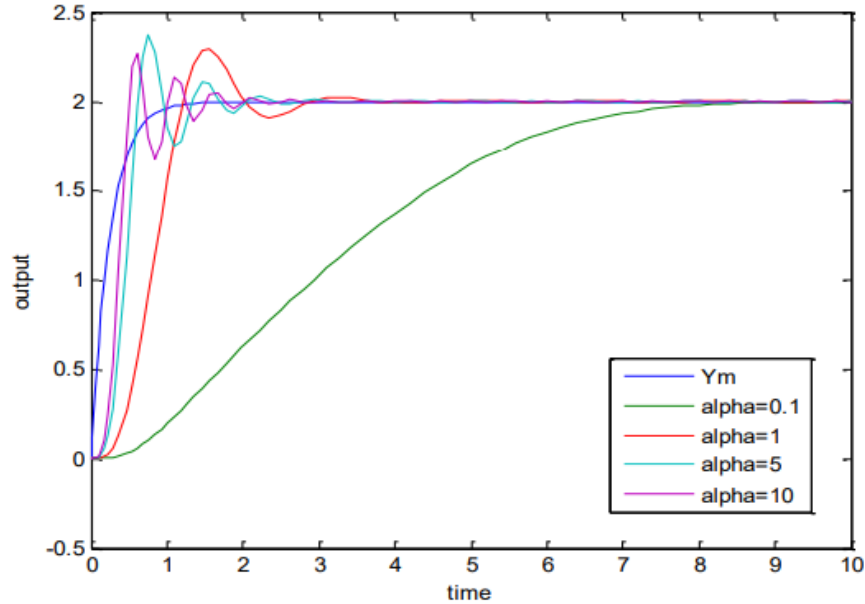


Figure 47 Effect of α on time response curve for MIT rule [2].

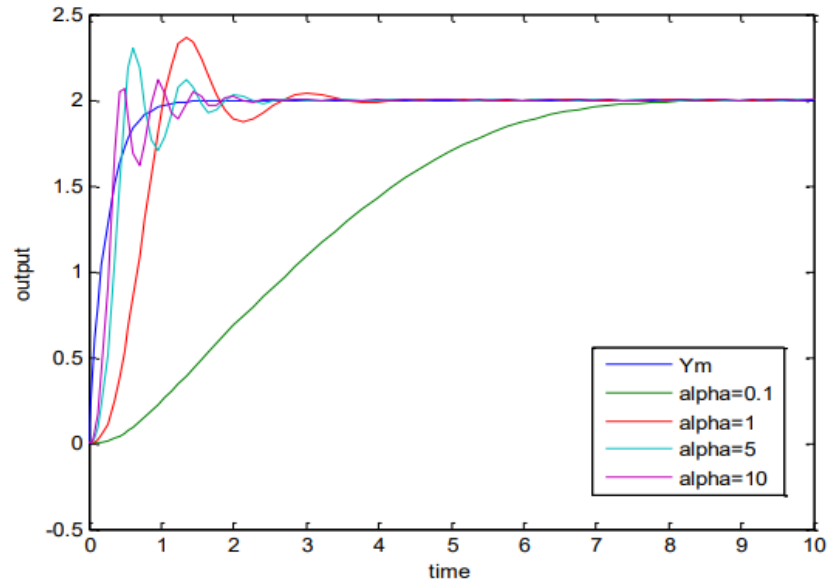


Figure 48 Effect of α on time response curve for Lyapunov rule [2].

2.4 Conclusions

To summarize, The MIT rule is often simpler to implement mathematically, making it easier to apply in certain cases while Lyapunov-based methods might involve more complex mathematical derivations, making them more challenging to implement and understand. Though the choice of adaptation gain in the MIT rule can be critical, and its value may depend on signal levels, which can make it challenging to determine the optimal gain.

Furthermore, the convergence rate of parameters under the MIT rule may be comparatively slower than that of the Lyapunov rule. Notably, Lyapunov stability theory provides a robust framework, ensuring the stability of the adaptive control system. Consequently, the Lyapunov method is favoured over the MIT rule, offering enhanced stability and potentially more rapid convergence.

3. References

- [1] Coman S., B. C. (2013). Model reference adaptive control for a dc electrical drive. *Engineering Sciences*.
- [2] Pankaj S., K. J. (2011). Comparative Analysis of MIT Rule and Lyapunov Rule in Model Reference Adaptive Control Scheme. *Innovative Systems Design and Engineering* 2, 154.