

Article type : Research Article

Date Received : 11/07/2021

Date Accepted : 08/08/2021

Date published : 01/09/2021



: www.minarjournal.com

<http://dx.doi.org/10.47832/2717-8234.3-3.14>



EFFECT OF MOVING AVERAGE FILTER WITH PID CONTROLLER USING HIL TECHNIQUE

Bilal A. MUBDIR¹

Abstract

Designing fast response controller that can achieving steady state with minimum error is big challenge in control application. Proportional Integral Derivate (PID) controllers are conventional controllers for their simplicity and wide range of applications in industrial. Many efforts have been made in improving the PID controller performance by enhancing the controller with different techniques for tuning its operation or filtering the feedback signal. In this paper, the effect of using Moving Average filter with the PID controller for controlling the speed of Permanent Magnet DC (PMDC) Motor with noisy speed sensor is discussed. PID controller and the moving average filter are modeled in MATLAB/Simulink software, where the Hardware in the Loop (HIL) technique with aid of microcontroller was used for the controlled plant which is a PMDC motor in this study. Comprehensive study on the available techniques used to modeling the PID controller is done as well. Furthermore, full practical tests prove that the controller behavior is more stable when moving average filter used and its effect minimized time needed to reach the steady state point clearly by 76%.

Keywords: HIL, PID, Moving Average, Microcontroller, MATLAB, Speed Control.

¹ Sulaimani Polytechnic University, Iraq, bilal.mubdir@spu.edu.iq, <https://orcid.org/0000-0001-8096-8027>

Introduction

One of the most important things to pay attention to when designing a control system is the stability of the system and obtaining the lowest error rate with respect to the set point of the system. Real time control is widely used in many industrial applications where it could be achieved by different approaches and methods.

Among all available techniques, Proportional-integral-derivative (PID) control are the traditional controls that used since more than 70 years in control engineering science [1]. Nowadays, more than 90% of the applied consumers and industrial control systems are built based PID controllers. Different PID software, patents, and hardware modules are discussed and the differences between the industrial and academic research practice are bring out in [2]. Furthermore, in NASA rocket lunch systems the control process is commonly performed by PID controller as reported by Y. Eisenberg and M. Dupuis in [3].

PID controllers in the past decades are developed by using analog circuits that has its own limitations. With emerge of computer technology and the continuous development in embedded processors technology in speed of performance and the cost, implementing PID controllers currently is done easily by deploying the PID algorithm into a single microcontroller unit. Moreover, in academic research many software packages provide an easy way to simulate the mathematical models of the PID controller such as MATLAB and LabView software application. Out of many examples and applications for developing PID controllers, controlling the speed of a DC motor is the most used application due its simplicity and easy of setup.

C. Agarwal and A. Gupta have used FPGA to develop a speed controller of PMDC motor with the aid of encoders for speed feedback. They implemented PID controller model under MATLAB with predefined parameters, where the model has been simulated to test its effectiveness then uploaded to the FPGA. Their developed speed controller proves high accuracy in controlling the speed compared to the traditional microcontroller. The tested the model n different types of FPGA boards and they compared the performance for each one[4].

The authors in [5] have designed and tested a PID controller to achieve a predefined threshold for the rise time, settling time, and the steady state error-based microcontroller. Primarily, their design has been configured under Simulink and with simulated PMDC motor. The practical they have tested the PID controller on an Atmega328 microcontroller under Proteus Software simulation application. The microcontroller has been connected to a PMDC motor and rotary encoder to feed the speed back to the microcontroller. They reported that their aims have been achieved under this design.

The power of using simulation software such as MATLAB offers a great method to simulate PID controller or any other types of controllers and test their performance. Not only simulating the controller, also the controlled plant may be simulated which the DC motor in which this paper is focused on as explained in [6], where Ling Xu et al., have simulated the equivalent model of the brushless type of DC motor in Simulink and they used double loop technique in their control mode to ensure the performance and the speed stability of the motor.

In 2017, K. Sharma and D. K. Palwalia have proposed a PID controller tuned by Fuzzy logic control. In their design, the PID controller parameters have been controlled by the Fuzzy logic control in order to provide better control over the speed of the DC motor. The work has been performed under MATLAB/Simulink software and the performance of the Fuzzy logic tuned PID (FPID) controlled has been compared to the conventional PID controller. As a result, the proposed FPID achieved higher performance than the conventional PID in terms of speed, steady state error, and overshoot [7].

Another control approach was used by N. Pati and his friend under MATLAB/Simulink software. The researchers have simulated the layout of Youla Parameterization approach that depend on the plant model inversion to control separately excited PMDC Motor and then compare it to the classic PID controller without filter for the same study case. The aim of their research is to present the performance analysis of the closed loop control for the motor in two different approaches [8].

G. Shahgholian et al. has focused on the adaptive model control and their advantages over the conventional non-adaptive control models when it used to control the speed of the motors. They aimed to increase the stability of control system and reduce the consequences of load disturbances and changes. They simulated the in MATLAB/Simulink software the model of two adaptive controller type, Fuzzy and reference one. The two-controlapproached, and non-adaptive control model have been studied and compared to each other. The authors have concludedafter testing that the performance of the system is better when using adaptive controller rather than

conventional type. Also, the found that the adaptive Fuzzy type provide superior performance over the other adaptive types.[9].

Controlling the speed of a DC motor is not necessary to be in a closed loop control system as in PID control criteria. For example, S. N. Thanh and his colleagues have been designed a sensor-less control system which is applied separately excited DC motor. Their system has been designed and simulated under Simulink and it technically reveals that its performance much better than the linear PI controller[10].

1. Hardware In The Loop (Hil)

In this paper, convention PID controller is used under MATLAB/Simulink software and Atmega328 microcontroller unit were utilized as a hardware in the loop (HIL) of the control system to control a physical Permanent Magnet DC (PMDC) Motor.

Regardless the issues in interfacing the hardware to the software environment and limitation of processing the time capabilities that reported in [11], HIL technique is used to test the designed controller, where it represents a type of real time simulation that allow the designer to test the designed controller, evaluate its performance, and its validity in real time as show in Figure (1) [12, 13].

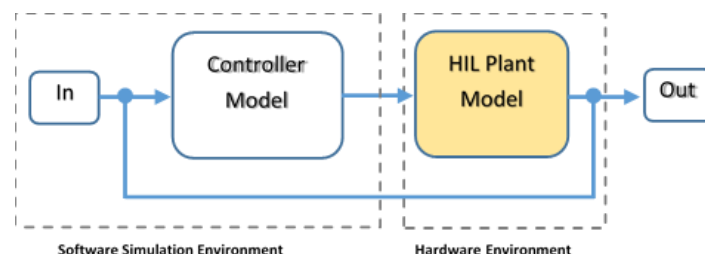


Figure 1: Hardware in the loop technique in a closed loop control

Practically, HIL technique has been used to develop a closed loop PI controller for PMDC motor in reference [14]. H. Khan and his colleague have used the 8-bit Atmel 8051 microcontroller as a hardware in the loop of the controller. The algorithm of the PI controller was deployed into the microcontroller, while the DC motor as the controlled plant has been simulated in the MATLAB/Simulink software. The embedded processor Atmel 8051 feed the control signal to the plant in Simulink, and the output of the plant is feed to the microcontroller again. Both feeding is carried out from/to the microcontroller via serial communication port.

Literature [15] described a method for characteristics analysis and comparison of two control methods for the permanent magnet synchronous motor (PMSM). The researchers have used high efficiency PSMS development method according to rapid control prototype (RCP). Hardware in the loop simulation model have been developed by using hardware and software environment system, the simulation carried out by MATLAB and CCS softwares. The hardware such as the PMSM, driving circuit, encoders and other circuitry have been interfaced to the real time control TMS320F28335 32-bit microcontroller unit. Their results show the hardware in the loop simulation performed very well as expected in the mathematical models with the same PID controller parameters

In [16], H. Hameed has simulated the brushless DC motor in MATLAB/Simulink and designed a PID controller to regulate the BLDC motor's speed. The designed controller has been compared to the Proportional (P), and Proportional Integral (PI) controllers under the MATLAB simulation environment. His results show that the designed PID controller performed better than the other types. The author also has used the Hardware in the loop technique to test the output signal of the controller on the oscilloscope which is in the form of PWM by using Arduino development board.

Likewise, M. Gavran and his colleges have implemented a closed-loop controller by using Arduino development board that contain Atmega328 microcontroller from Atmel and PI controller built in MATLAB/Simulink installed on personal computer with help of speed sensor that works as a negative feedback to the controller. They used a ready to use software developed by MathWorks

Company to access the pins and the features of the microcontroller from the Simulink. They tested the system practically on a DC motor loaded practically with generator [17]. Using HIL is not limited only to test and validate controllers. For example, Y. C. Chou et al. have present HIL simulation to test autonomous underwater vehicles (AUV) and developing its control system. They have used HIL between the AUV computer and the Simulation server. They concluded that developing control system for the AUVs could be more effective if the HIL simulation is used [18].

2. Moving Average Filter

In closed-loop control system, using proper filter may leads to significant improvement on stability of the system and its performance [19]. Sudden or abnormal disturbances appeared on the feedback of the closed-loop controller, will be reflected proportionally on the output of the controller and so the behavior of the controlled plant. The usage of the filter must be targeting that sudden only to eliminate its effect of the controlled plant. Usually, some filters are specified to work with control systems, such as First Order Lag, Second Order, Butterworth Second order, or Averaging filter [20].

Among above mentioned types, Averaging or Moving Average (MA) filter, or sometimes named "Rolling Average" filter, considered to be the simplest one. It is window-based filter in which the average of specific number of samples called Segment or "window" is calculated [21, 22].

Moving average classified into three main types, they are either Simple, Cumulative, or Exponential Moving Average. The Simple Moving Average work on current data with time series and neglect the old data, whereas the Cumulative type is considering the old data in the averaging calculation. On other hand, Exponential type moving average is working on the data pattern and give precedence to the fresh data in data series or signal and it is used mainly for forecasting purposes due to its efficiency and higher effect [22, 23].

In this paper, the Simple Moving Average (SMA) filter is used to filter out the noise in the feedback of the closed-loop controller with different window's sizes. The SMA is governed by the mathematical expression of equation (1), knowing that n is the number of the previous data points or the window size [24].

$$SMA = \frac{P_M + P_{M-1} + P_{M-2} + \dots + P_{M-(n-1)}}{n} \quad (1)$$

Where:

SMA: Simple Moving Average value

PM: point value at time M

n: window size or number of data points

3. Problem Statement

The control process of an electromechanical system sometimes involved with noise generated by mechanical vibrations or electronics components interference and other causes. In any embedded system, sensors are the most part affected by noise, and it could result in an error on its output. Considering closed-loop controller controlling DC motor possess the specifications outlined in Table (1), with a noisy sensor used to measure the motor speed as return feedback for PID type controller. The noise generate by the sensor will affect the behavior of the speed response of the DC motor and develop serious delay in reaching the steady state level of the desired reference speed of the DC motor.

Table 1: The specifications of the DC motor used in experiments

Brand	Graupner Jumbo 550
Voltage	9V
Maximum Current	1.9A
Gearbox Ratio	1.5:1
Maximum Speed	13500 rpm

4. Control Model Configuration

In MATLAB/Simulink, the default block for the PID controller in discrete time has been used to build the closed loop control system. The model includes a step input that represents the desired speed in which the DC motor has been tested in addition to another required blocks in the same control loop to send and receive data from the external hardware, which is act as Hardware in the Loop (HIL) as shown in Figure (2).

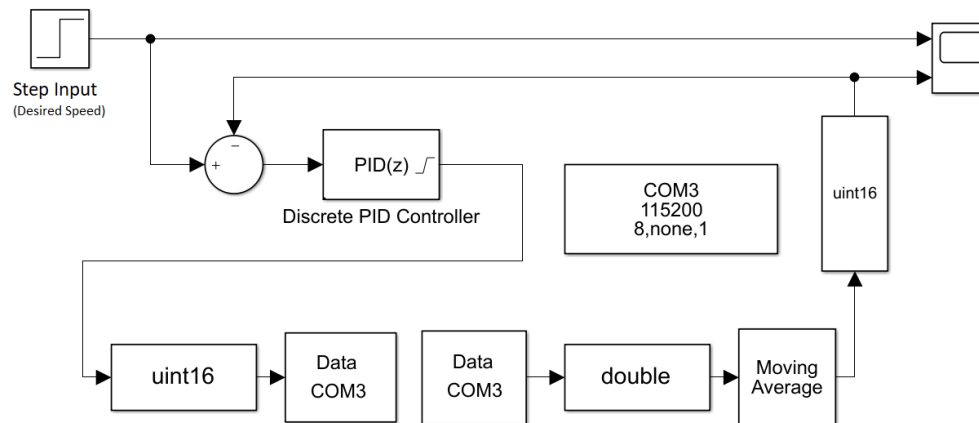


Figure 2: The simulate model for the closed-loop control system with HIL.

The HIL has been interfaced over Serial communication Protocol with aid of an external UART circuit. The physical stage here is the Atmega328P microcontroller unit (MCU) from Atmel [25]. According to Figure (3), the MCU is connected to the motor driving circuit and speed measurement circuit as well.

The motor driving circuit is designed using MOSFET operated as switch to control the voltage of the motor by varying duty cycle PWM signal from the MCU, thus controlling its speed as shown in Figure (4).

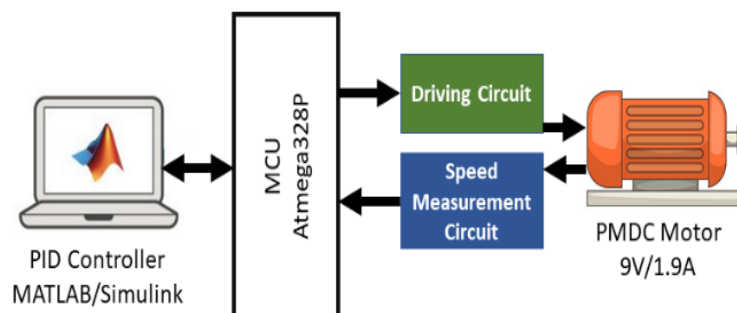


Figure 3: MCU including motor driving and speed measuring circuits

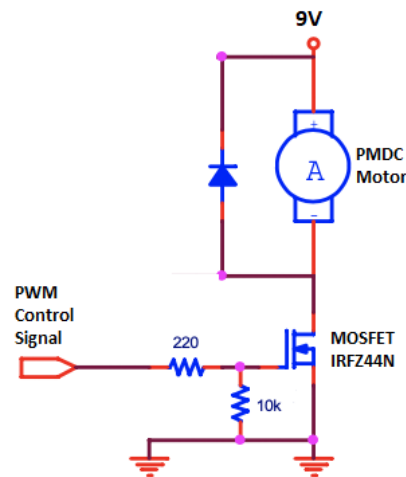


Figure 4: PMDC motor driving circuit

On the other hand, simple encoder circuit was used to measure the speed of the motor with the aid of plastic wheel attached to the motor's rotational shaft. The encoder circuit built using a pair of IR transmitter and receiver as illustrated in Figure (5), to generate a pulse every single complete rotation. The MCU has been programmed to accumulate the pulses and calculate the instantaneous revolutions per minute (RPM) and driving the MOSFET continuously according to the received signal from the PID controller in MATLAB.

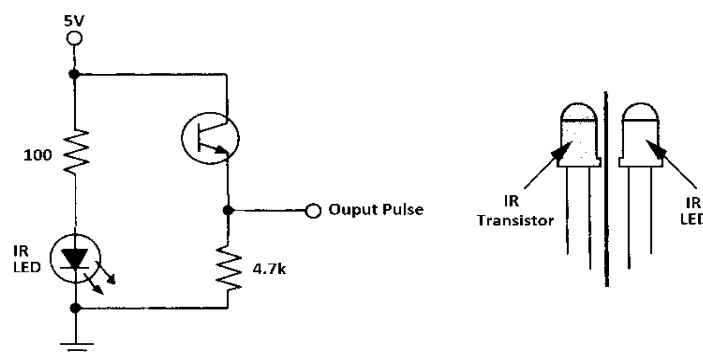


Figure 5: Motor Speed measurement circuit

The data are sent and received every 50ms over serial communication protocol with a baud rate of 115200 bps. The while Simulink model has been configured to work on a sampling rate of 50ms to be compatible with the HIL. The motor used in the experiments is Brushed DC motor with technical specifications as explained in Table (1). The motor operated by separated 9V power supply with a total of 1.9A current consumption on rated speed. Moreover, The Simulink model has other blocks for data collection purposes and exporting waveforms drawings. Moving average block with different window sizes has been used in five tests and disabled on others by commenting it through.

5. Results & Discussion

The tests are carried out for duration of 20 sec with a desired speed of 5000 RPM as a step input to the PID controller and the response of the motor and its speed has been monitored. The PID controller's parameters used, shown in Table (2), are pre-selected and tested to maintain stable control response when the speed measurement sensor is accurate. Then, the sensor has been replaced with noisy one, that measure the speed with ± 50 RPM and speed response of the motor has been obtained. The motor response in both cases, when the sensor speed measurement sensor is accurate and noisy are shown in Figure (6).

Table 2: PID controller parameters used in the control system

Parameter	Value
P	2.5
I	1
D	0.5

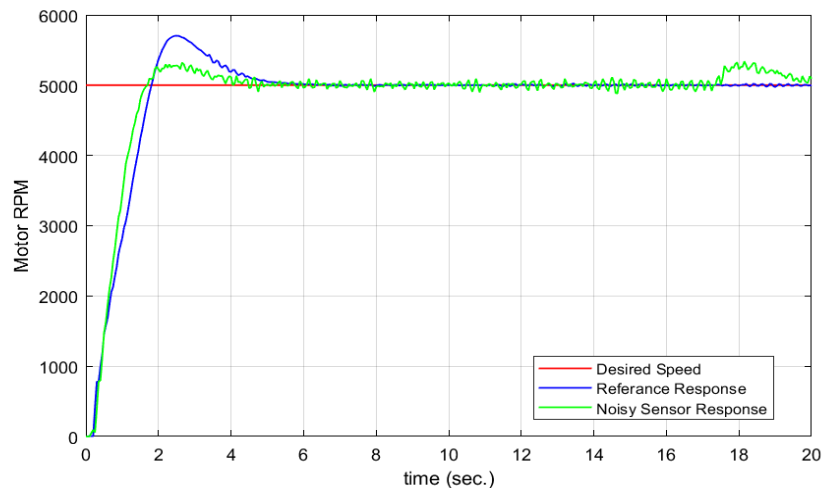


Figure 6: Motor's speed response when sensor is accurate and noisy

The resultant response in Figure (6) shows that when the sensor is accurate the overshoot is 14% and the settling time is 4.544 sec, this is the case where it is considered as the reference case for the controller in this research. It is clearly seen in Figure (6) that noisy sensor results in less overshoot 6.78% but with longer settling time 19.987 sec. Longer settling time with continuous fluctuation could lead to unstable performance, which is not preferable for critical applications.

At this point, moving average filter with different window sizes has been embedded in the control loop on Simulink and the step response tested to obtain the speed response. The tests are carried by set the moving average's window size to 5, 10, 15, 20, 30, and 50 to evaluate the performance in each case. The speed response is shown in Figures (7) to (12) for all each case 5, 10, 15, 20, 30, and 50 respectively.

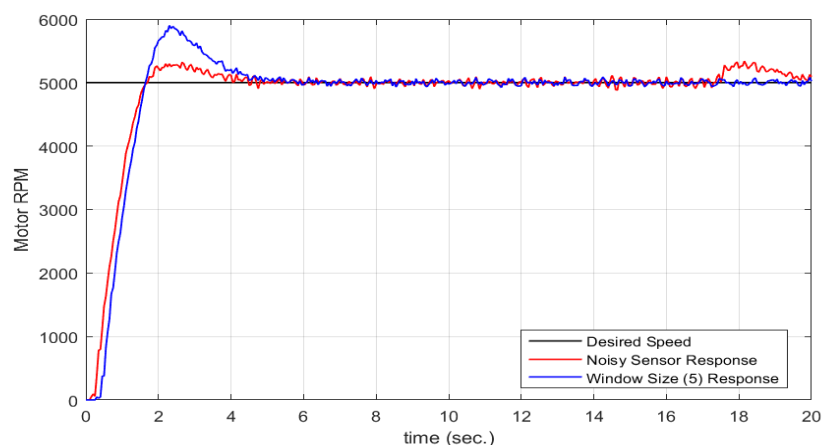


Figure 7: Motor's speed response with Moving Average Filter used (window size = 5).

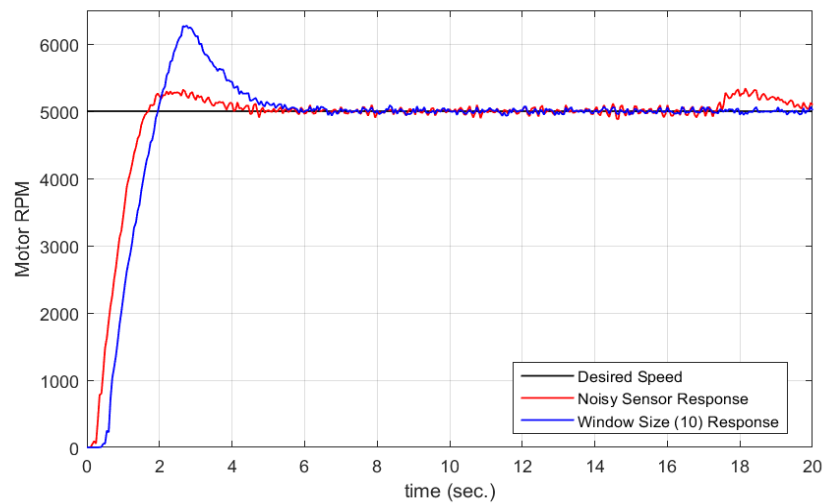


Figure 8: Motor's speed response with Moving Average Filter (window size = 10).

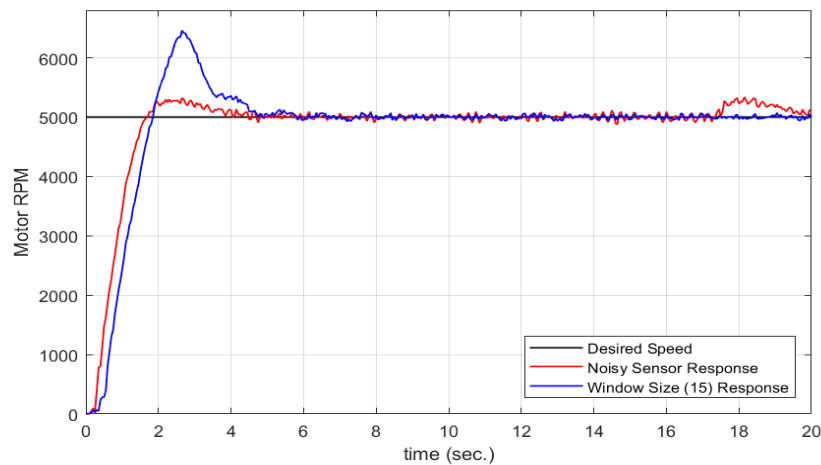


Figure 9: Motor's speed response with Moving Average Filter used (window size = 15).

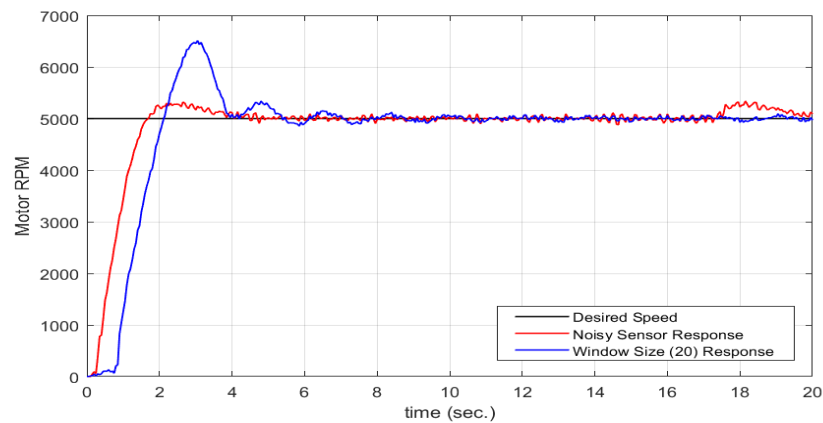


Figure 10: Motor's speed response with Moving Average Filter used (window size = 20).

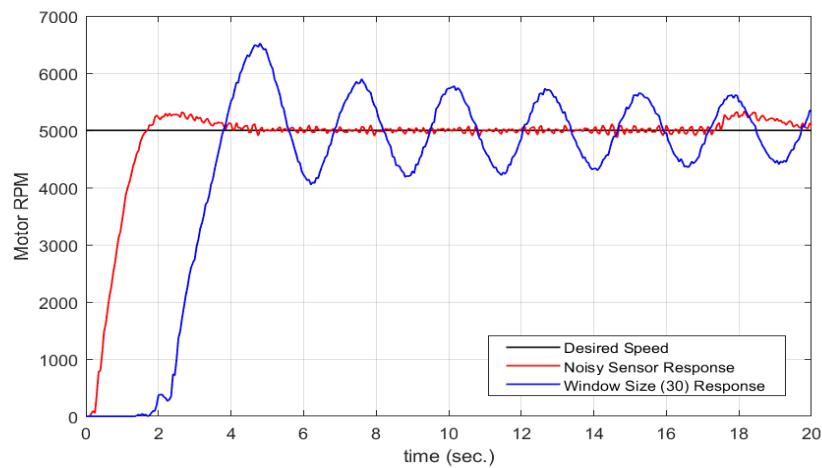


Figure 11: Motor's speed response with Moving Average Filter used (window size = 30).

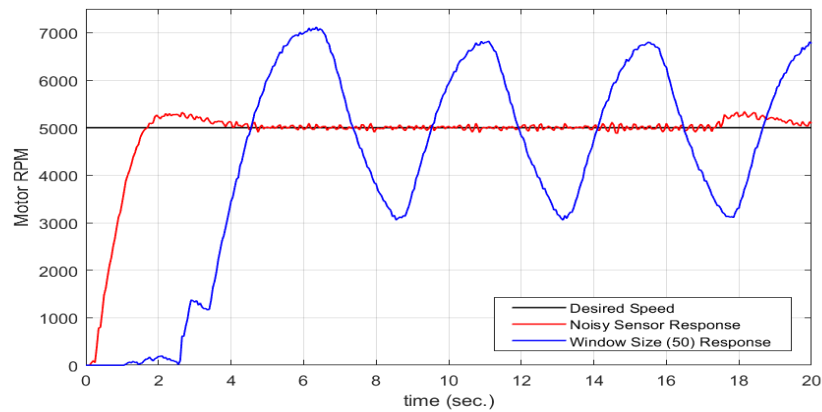


Figure 12: Motor's speed response with Moving Average Filter used (window size = 50).

With quick look at the resultant responses, it is obvious the change in the overshoot, settling time, and thus the stability. Table (3) summarized the step response characteristics for each case in addition to the reference case. As the moving average's size of the window increases, the noise eliminated but the response tends to be unstable. With lower window's size, the stability was enhanced better than larger size.

Table 3: Summary of all experimental tests

	Reference Response	Response with noisy sensor	Response with Moving Average Filter					
			Size of window = 5	Size of window = 10	Size of window = 15	Size of window = 20	Size of window = 30	Size of window = 50
RiseTime (sec.)	1.337	1.0505	0.9631	1.113	1.0731	1.0496	1.2429	1.7184
SettlingTime (sec.)	4.544	19.987	4.802	5.353	5.47	8.4108	NaN	NaN
SettlingMin (rpm)	4505	4593	4589	4530	4543	4552	4051	3054
SettlingMax (rpm)	5703	5339	5899	6272	6457	6510	6524	7119
Overshoot (%)	14.06	6.78	17.98	25.44	29.14	30.2	30.48	42.38
Undershoot (%)	0	0	0	0	0	0	0	0
Peak (rpm)	5703	5339	5899	6272	6457	6510	6524	7119
PeakTime (sec.)	2.5	18.15	2.3	2.75	2.65	3.05	4.8	6.35
Improvement (sec.)	-	-	15.185	14.634	14.517	11.577	-	-
Improvement (%)	-	-	75.97	73.21	72.63	57.92	-	-
Delay to reference (sec.)	-	-	0.2579	0.8094	0.9259	3.8667	-	-

It's clearly seen from Table (3) that when the window size is more than 20, the system is completely became unstable and never reach to the steady state. At the lowest window size, the performance of the step response for the noisy sensor was improved by about 76% in terms of Settling time, and it reaches the steady state level just 257ms later to the reference response shown in Table (3) but with larger overshoot. When the size of the window doubled to 10, the settling time increases again and became 5.353 sec but still acceptable in terms of stability with 73.2% improvement compared to the noisy sensor case step response. Furthermore, the settling time is 5.47 sec and 8.41sec when the moving average filter's window size is 15 and 20 respectively with an improvement of 72.6% and 57.9% compared to the noisy sensor case step response.

5. Conclusion

To sum up, Using the HIL technique provide an easy method to test and verify the control system by integrating the facilities of the computer-based applications like MATLAB with the real physical plant. Improving the closed-loop control system can be performed using different techniques, extra tuning for the controller parameters, or using proper filters to achieve better performance. The results proved that using even a simple moving average filter with minimum window size leads to a significant improvement in stability in case of using a noisy sensor in the feedback of the closed-loop control system. Where the time needed for reaching the steady state level is reduced by 15.18 sec with an overall improvement of 76%. However, an increase in overshoot along with the improvement in reaching the steady state level is observed, where it could be not preferable in some cases which must be considered for specific applications.

Reference

- A. O'Dwyer, "PI and PID controller tuning rules: an overview and personal perspective," 2006 IET Irish Signals and Systems Conference, pp. 161-166, 2006.
- Anumandla Kiran Kumar et al, "Moving average method based air pollution monitoring system using (IoT) platform," IOP Publishing, Journal of Physics: Conference Series, vol 1706, 2020.
- Atmel Corporation, "8-bit AVR Microcontroller with 4/8/16/32K Bytes In-System Programmable Flash," ATmega 328P datasheet, 2019.
- C. Agarwal and A. Gupta, "Modeling, simulation-based DC motor speed control by implementing PID controller on FPGA," Confluence 2013: The Next Generation Information Technology Summit (4th International Conference), pp. 467-471, 2013. doi: 10.1049/cp.2013.2358.
- Chou, Yu-Cheng, et al., "A Hardware-in-the-Loop Simulation Platform for Development of AUV Control Systems." 2019 IEEE Underwater Technology (UT), pp. 1-6, 2019.
- Crenna, Francesco, Giovanni B. Rossi, and Marta Berardengo, "Filtering Biomechanical Signals in Movement Analysis," SensorsJournal, no. 13: 4580, 2021. <https://doi.org/10.3390/s21134580>
- de Jong, Erik & Vaessen, Peter & Graaff, Roald, "The role of hardware in the loop in validation and testing," International Journal of Distributed Energy Resources and Smart Grids, vol. 9, pp1-6, March 2013.
- G. Shahgholian, M. Maghsoodi, et al., "Analysis of speed control in DC motor drive by using fuzzy control based on model reference adaptive control," 2016 13th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), pp. 1-6, 2016. doi: 10.1109/ECTICon.2016.7561239.
- G. Zhao and S. Zhang, "Development of Permanent Magnet Synchronous Motor RCP Based on Hardware-in-the-loop Simulation," 2018 IEEE International Conference on Mechatronics and Automation (ICMA), pp. 264-269, 2018. doi: 10.1109/ICMA.2018.8484509.
- H. S. Hameed, "Brushless DC motor controller design using MATLAB applications," 2018 1st International Scientific Conference of Engineering Sciences - 3rd Scientific Conference of Engineering Science (ISCES), pp. 44-49, 2018. doi: 10.1109/ISCES.2018.8340526.
- H. S. Khan and M. B. Kadri, "DC motor speed control by embedded PI controller with hardware-in-loop simulation," 2013 3rd IEEE International Conference on Computer, Control and Communication (IC4), pp. 1-4, 2013. doi: 10.1109/IC4.2013.6653742.
- K. Sharma and D. K. Palwalia, "A modified PID control with adaptive fuzzy controller applied to DC motor," 2017 International Conference on Information, Communication, Instrumentation and Control (ICICIC), pp. 1-6, 2017. doi: 10.1109/ICOMICON.2017.8279151.
- Li, Y. and Ang, K.H. and Chong, G.C.Y., "Patents, software and hardware for PID control: an overview and analysis of the current art," IEEE Control Systems Magazine, vol. 26, no.1, pp. 42-54, 2006.
- Ling Xu, Jian-Guo Song and Qiang-Qiang Lin, "Brushless DC motor speed control system Simulink simulation," 2016 IEEE International Conference on Power and Renewable Energy (ICPRE), pp. 62-66, 2016. doi: 10.1109/ICPRE.2016.7871173.

- M. Gavran, M. Fruk and G. Vujisić, "PI controller for DC motor speed realized with Arduino and Simulink," 2017 40th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), pp. 1557-1561, 2017. doi: 10.23919/MIPRO.2017.7973669.
- Mathworks. (R2021a). "Basics of Hardware-In-The-Loop simulation," Accessed on July 25, 2021, available: <https://www.mathworks.com/help/phymod/simscape/ug/what-is-hardware-in-the-loop-simulation.html>
- Michel Ruel P.E., "Using Filtering to Improve Performance," ISA Expo: The Instrumentation, Systems and Automation Society, 2003.
- N. M. A. Mohamed et al., "Implementation of a PID control system on microcontroller (DC motor case study)," 2017 International Conference on Communication, Control, Computing and Electronics Engineering (ICCCCEE), pp. 1-5, 2017. doi: 10.1109/ICCCCEE.2017.7866088.
- N. Pati and N. Swain, "Design and study of speed control of DC motor using Youla parameterization and PID controller," 2017 IEEE Calcutta Conference (CALCON), pp. 433-437, 2017. doi: 10.1109/CALCON.2017.8280770.
- National Instruments Corp, "What Is Hardware-in-the-Loop?", updated on Dec 17, 2020, available: <https://www.ni.com/en-lb/innovations/white-papers/17/what-is-hardware-in-the-loop-.html>
- S. Hansun, "A new approach of moving average method in time series analysis," 2013 Conference on New Media Studies (CoNMedia), pp. 1-4, 2013. doi: 10.1109/CoNMedia.2013.6708545.
- S. Nguyen Thanh, C. Nguyen, and H. Ha Xuan, "Improved performance of a sensorless DC motor control using fuzzy logic," 2014 5th International Conference on Intelligent and Advanced Systems (ICIAS), pp. 1-6, 2014. doi: 10.1109/ICIAS.2014.6869506.
- Tore Hägglund, "Signal Filtering in PID Control," IFAC Proceedings Volumes, vol. 45, no. 3, pp. 1-10, 2012.
- Y. Eisenberg, and M. Dupuis, "Adaptive Control Law for PID," Grantee Report, NASA Internships and Fellowships (NIF), April 12, 2019.
- Yu, Jaehong, Seoung B. Kim, Jinli Bai, and Sung W. Han, "Comparative Study on Exponentially Weighted Moving Average Approaches for the Self-Starting Forecasting," Applied Sciences, vol. 10, no. 20: 7351, 2020. <https://doi.org/10.3390/app10207351>