**MODELATED SYSTEM**

To model the transfer function represented by the DC motor speed system shown in Figure 1, it was necessary to take measurements of the system's input and output signal, contemplating the fact that with the input variable it was generated a step type signal simulating a PWM change of 10 to 60% to obtain a response from the system and describe the speed change in RPM with respect to time.



**Figure 1.** Image of the physical system.

The data obtained experimentally were entered into MATLAB to perform the modeling of the system represented by the data in Figure 2 and Figure 3. The code entered in MATLAB is shown in Figure 4.

**Figure 2.** System input (variation of the PWM signal from 10 to 60% simulating a step).

**Figure 3.** System output (system response in RPM before a step input).

close all

clear

clc

time=[0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,48,49,50,51,52,53,54,55,56,57,58,59,60,61,62,63,64,65,66,67,68,69,70,71, 72,73,74,75,76,77,78,79,80,81,82,83,84,85,86,87,88,89];

dutyCycle=[10,10,70,70,70,70,70,70,70,70,70,70,70,70,70,70,70,70,70,70,70,70,70,70,70,70,70,70,70,70,70,70,70,70,70,70,70,70,70,70,70,70,70,70,70,70,70,70,70,70,70,70,70,70,70,70,70,70,70,70,70,70,70,,70,70,70,70,70,70,70,70,70,70,70,70,70,70,70,70,70,70,70,70,70,70,70,70,70,70];

variable\_rpm=[571.459167,576.725525,588.079712,567.189453,557.681335,570.780701,1318.07092,1986.33936,2474.3186,2745.15381,2826.96753,2855.41528,2908.76099,2979.03198,3004.44141,2983.88818,2975.6311,2998.55078,2986.59961,3025.92017,3037.58594,2922.97363,3098.24854,3004.19067,3015.01025,3025.29517,2979.81567,2986.05469,3009.72021,2973.25391,2882.31396,2914.08325,2992.80713,2957.19409,3045.25317,2914.40869,2947.58813,2943.76416,2900.36304,2931.11182,2922.14844,2956.64575,2895.35132,2931.95654,2901.74683,2852.34497,2898.37036,2909.00977,2985.18359,2928.80518,2937.84497,3005.69238,2938.75635,2934.40674,2996.27271,2913.71045,2849.02002,2912.17993,2935.00439,2860.97119,2938.03955,3010.03711,2923.30249,2835.57861,2956.76709,2898.87402,2955.33081,2952.2522,2940.54712,2935.40161,2938.51416,.97485,2919.84644,2949.18579,2968.52051,2937.20142,2948.77539,2978.36694,3059.33789,2960.17822,2958.44727,2938.22168,2955.0835,2944.45142,3059.31006,2939.18823,2933.29199,2966.81885,2952.55518, 2924.23584];

%Figure

plot(time,dutyCycle);

plot(time,variable\_rpm);

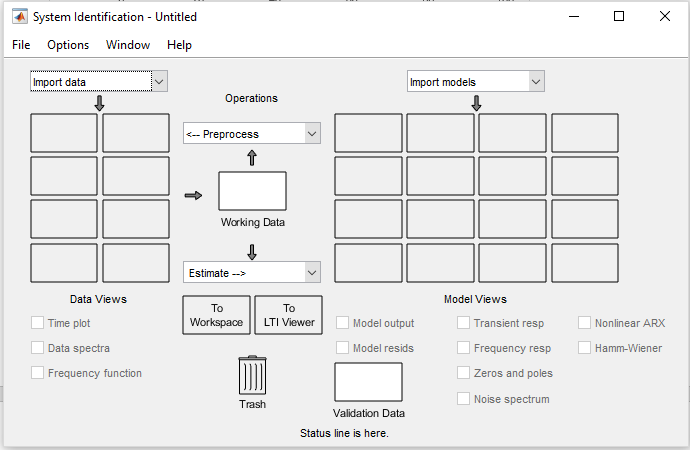
**Figure 4.** Code in MATLAB for insert Input and Output.

**SYSTEM IDENTIFICATION**

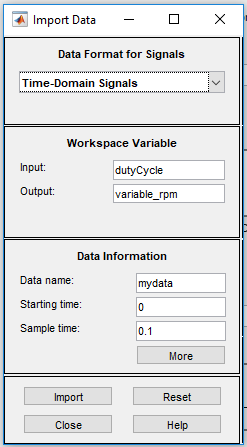
The identification of the system includes its modeling, in this case the “System Identification” tool was used in MATLAB which allows the system to be modeled by simply entering experimental input and output data. The Figure 5 shows the main window of the before mentioned identification module, in which the main panel at the top, a data import menu, a preprocess menu, an estimation menu and an import menu are shown. Models.

In the data import menu, the option “Time Domain Data” is selected, for the simple reason that the experimental data obtained was recorded in the time domain.

A new window in which the name of the input and output variable that are registered in the MATLAB workspace is inserted, followed by the initial time from where you want to contemplate the modeling and also the time of sampling that for this case would be 0.1 seconds. The import option is selected to confirm the use of the selected data, to see Figure 6.

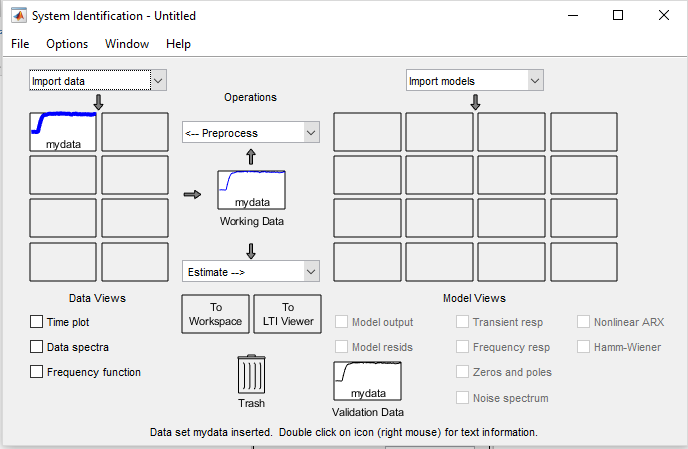


**Figure 5.** Window of System Identification tool from MATLAB.



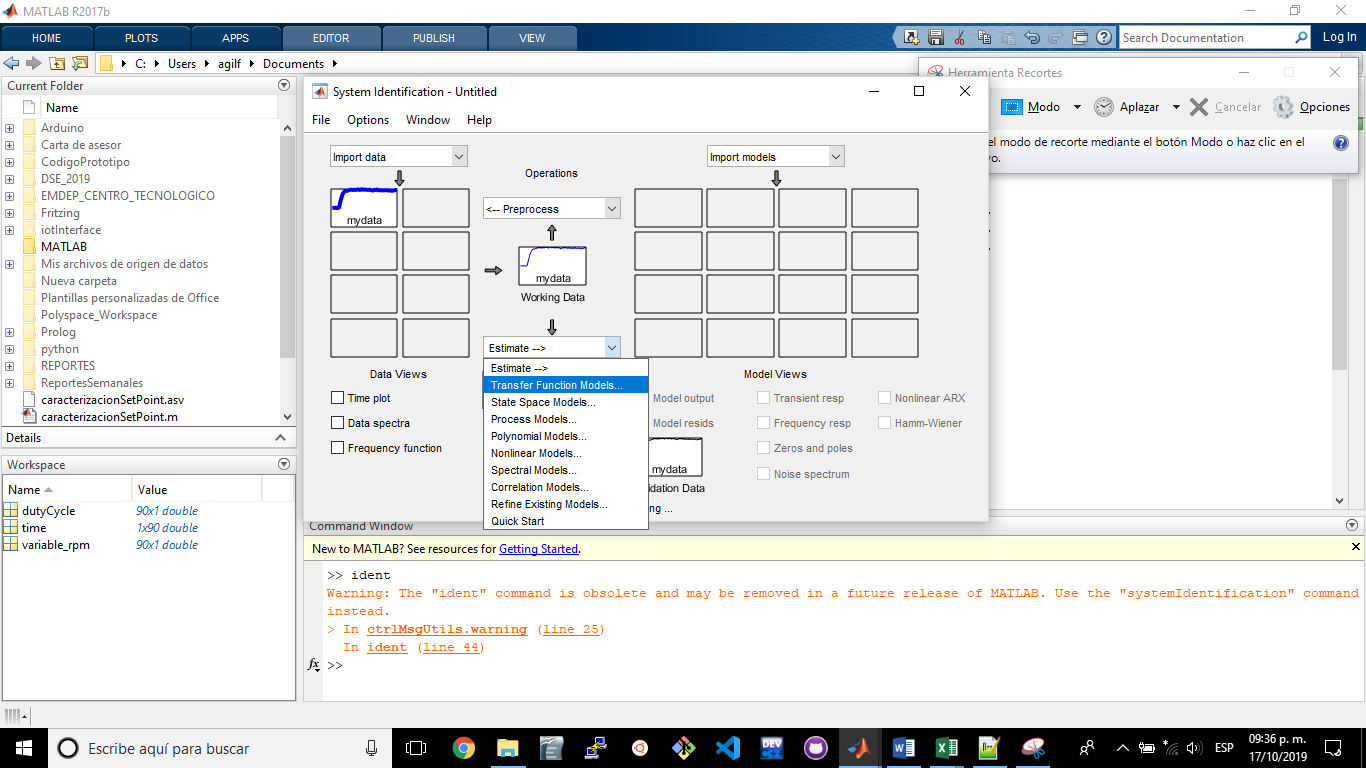
**Figure 6.**  Window to Import Data.

The Figure 7 allows you to visualize the moment after the data has been loaded to perform the estimate.

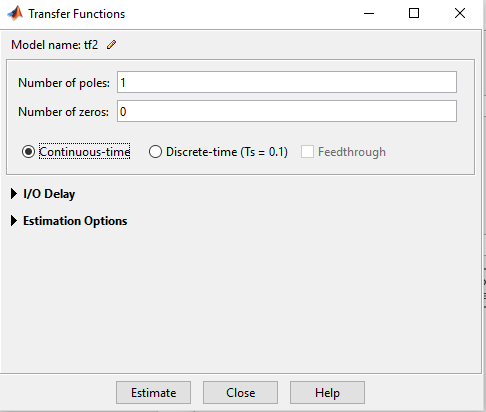


**Figure 7.** Update data.

The next step is to select in the estimation menu the option of transfer function models, which displays another window for the selection of parameters, in this case the selection of zero poles to define the polynomial order of the transfer function to be obtained, to see Figure 9. It is worth mentioning that due to knowing the system, it can be proposed that it has a single pole and no zero, because this information was obtained through previous research.

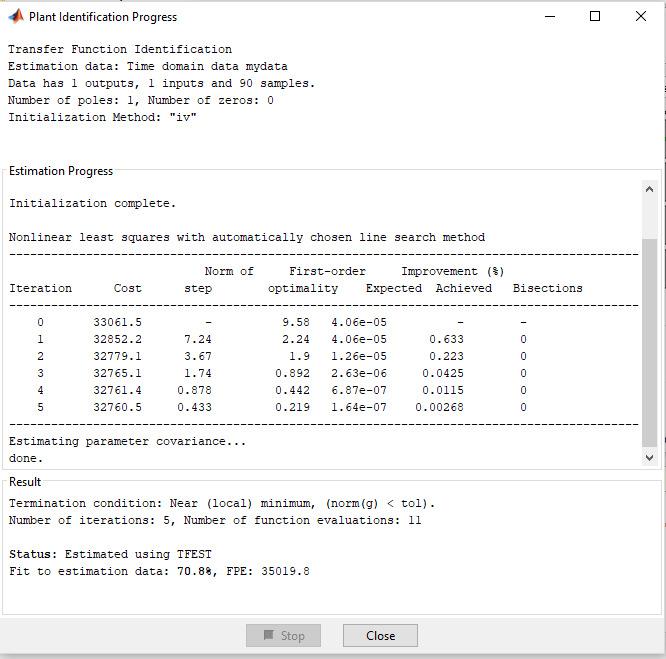


**Figure 8.** Selection of transfer function models option.



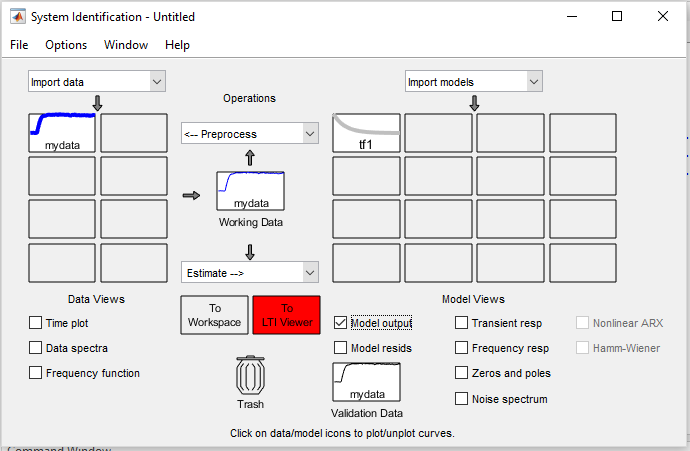
**Figure 9.** Selection of zeros (numerator) and poles (denominator) of the transfer function.

The Figure 10 shows the result of the estimation of the model in which a value of 70.8% similarity with the experimental data is observed, after performing five iterations.



**Figure 10.** Estimate result obtained.

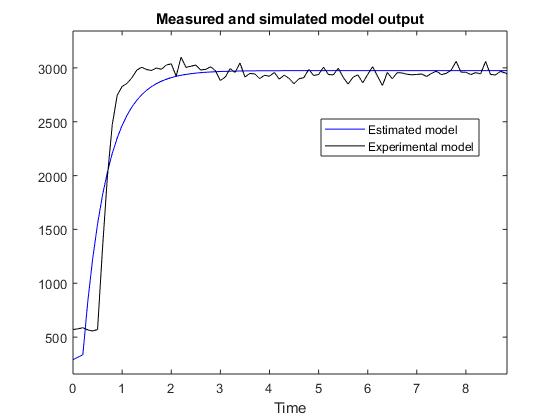
To graphically show the comparison of the experimental data against the model obtained by the System Identification, the check box named “Model output” is selected below the “Import models” menu in Figure 11. In addition to To extract the estimated mathematical model, you must drag the “tf1” model to the box called “To Workspace”.



**Figure 11.** Screen sample after the estimate is made.

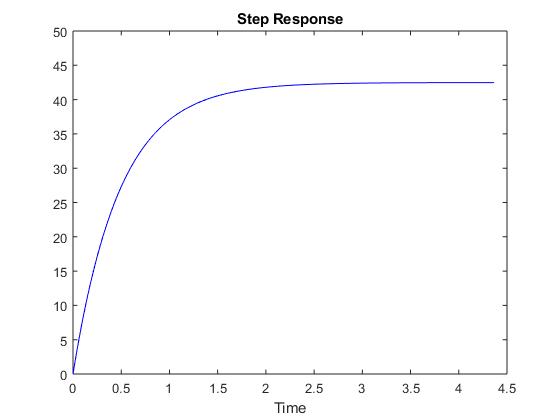
The transfer fuction obtained in continuous time is:

The comparison of the output of the measured (experimental) model against the estimated model showing the value of the rpm as a function of time is contemplated in Figure 12.

****

**Figure 12.** Measured and simulated model output.

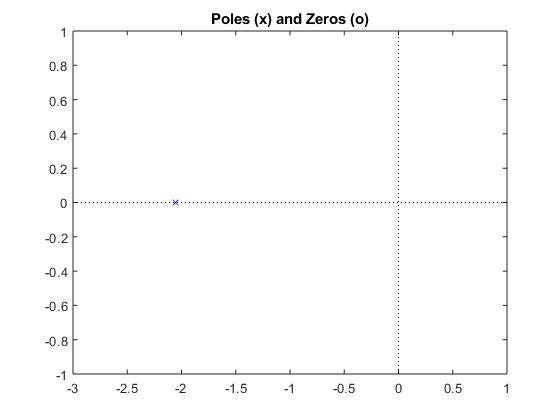
The response to the entry of a step was also obtained, where the stable state value corresponding to approximately 42.5 can be evaluated, as shown in Figure 13.



**Figure 13.** Step Response.

To analyze the stability of the plant, the location of the poles and zeros in the complex plane is reviewed, contemplating that the poles are denoted with “X” and the zeros with “O”, and with the premise that the poles would be in the left part of the plane if the system is stable and is located on the real axis.

The Figure 14 records the location of the only pole of the system on the real axis with a value of -0.2058.

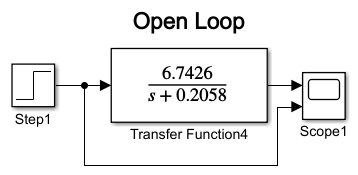


**Figure 14.** Poles and zeros of the estimated system.

**Open loop system**

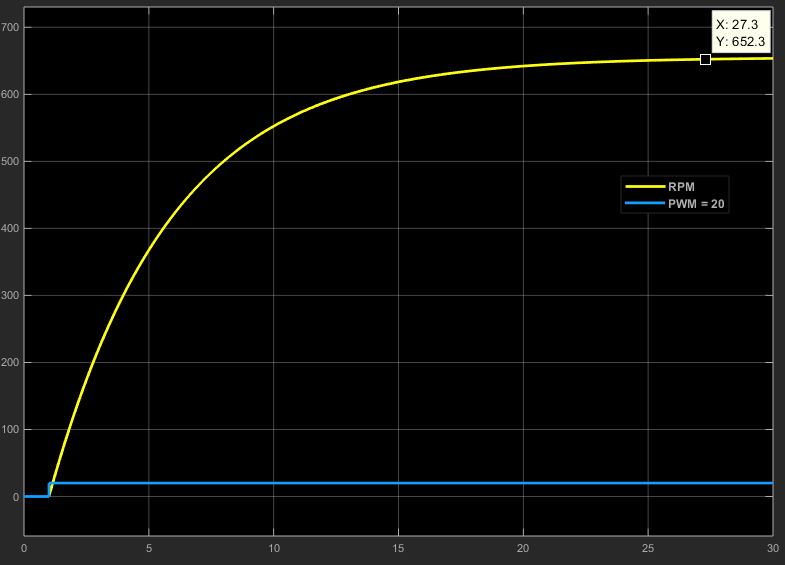
An open loop control system is characterized by the fact that it does not receive any information or feedback on the status of the variable, usually these are used when the variable is predictable and has a wide margin of error, since time can be calculated or the times the cycle must be repeated to complete the process.

In the Figure 15 shows the simulation scheme of the experimentally obtained plant model, connected to an input signal and a signal display to show the behavior of the input and output signal, respectively.



**Figure 15.** Open loop system.

Also, it shows in Figure 16 the graph obtained from the simulation of the open loop system with a step-type input signal with a PWM value of 20% and obtaining an RPM output of 652 approximately.



**Figure 16.** Pwm with value of 20% for a velocity from 652.3 rpm.

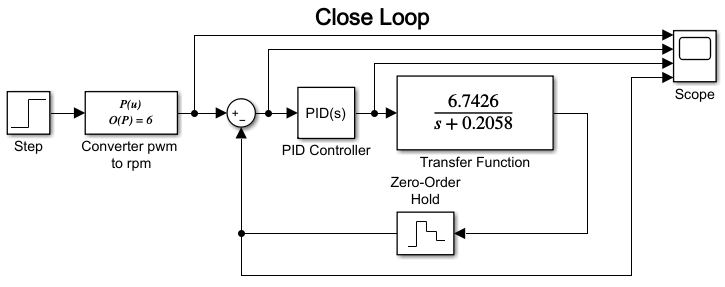
**Close loop system**

This system is more complete since it receives information about the states that the variable is taking. This feedback is achieved by placing sensors that send information on key points of the process so that it can act autonomously.

The closed loop system obtained is contemplated in Figure 17, in which there is the “Step” block that represents the input in PWM with a sampling time of 0.1 seconds, the “Converter pwm to rpm” block where through PWM change characterization from 0 to 100% was obtained RPM for the entire operating range of the system.

In the “PID Controller” block, the parameters of the PID controller found in simulation for the experimentally obtained plant are found. The “Zero-Order Hold” block is a sampler or discretizer of the continuous output signal of the plant, which in this case corresponds to the RPM sensed.

The block that joins the input signal in RPM with the output signal sensed and discretized in RPM of the plant is responsible for finding the difference or error signal that the system has and that is compensated by introducing the controller block called “PID Controller. "

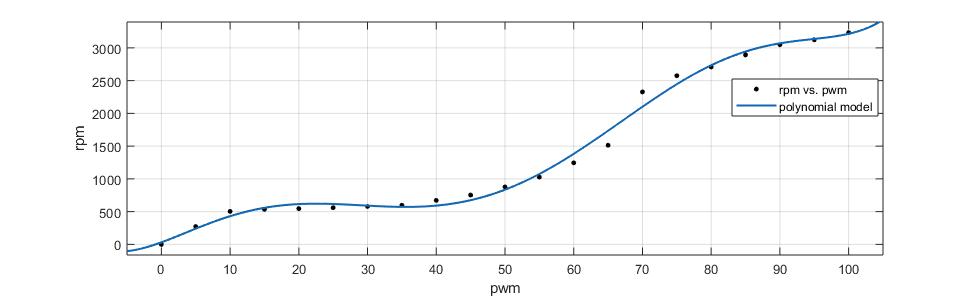


**Figure 17. Close loop system.**

To find the adjustment parameters of the polynomial used in the “Converter pwm to rpm” block, it was necessary to make a curve adjustment with the experimental data recorded in Figure 2 and Figure 3, previously.

**Curve Adjustment Polynomial model of degree 6:**

The adjustment of the polynomial model was measured at 21 points. For a PWM value of 5% the average speed was 272 RPM while for 100% of the PWM the average speed value was 3234 RPM; thus throwing **the operating range** **of the system** between **272 to 3234 RPM**, approximately.



**Figure 18.** Curve adjustment polynomial model obtained.

Function obtained: , with coefficients (with 95% confidence bounds):

Where is defined in PWM and correspond to RPM.

All this parameters were searched in MATLAB using the module curve fitting for a estimation good and with results follow:

***Goodness of fit:***

***SSE: 1.74e+05***

***R-square: 0.9928***

***Adjusted R-square: 0.9897***

***RMSE: 111.5***

The code used in MATLAB to load the data and find the polynomial setting is found in Figure 19.

close all

clear

clc

pwm=[0,5,10,15,20,25,30,35,40,45,50,55,60,65,70,75,80,85,90,95,100];

rpm=[0,272,504,536,547,561,580,598,673,754,877,1027,1245,1514,2329,2576,2709,2893,3050,3125,3234];

p=polyfit(pwm,rpm,4);

%graphic

hold on

plot(pwm,rpm,'ro','markersize',4,'markerfacecolor','r')

z=@(pwm) polyval(p,pwm);

fplot(z,[pwm(1),pwm(end)])

ylabel('rpm')

xlabel('pwm')

grid on

title("Polynomial regression")

hold off

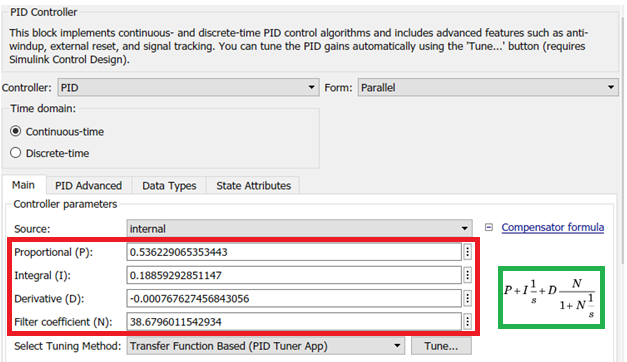
**Figure 19.** Code in MATLAB to load the data used in the polynomial adjustment.

**Manually tune a PID loop**

The procedure to tune the PID loop is described below:

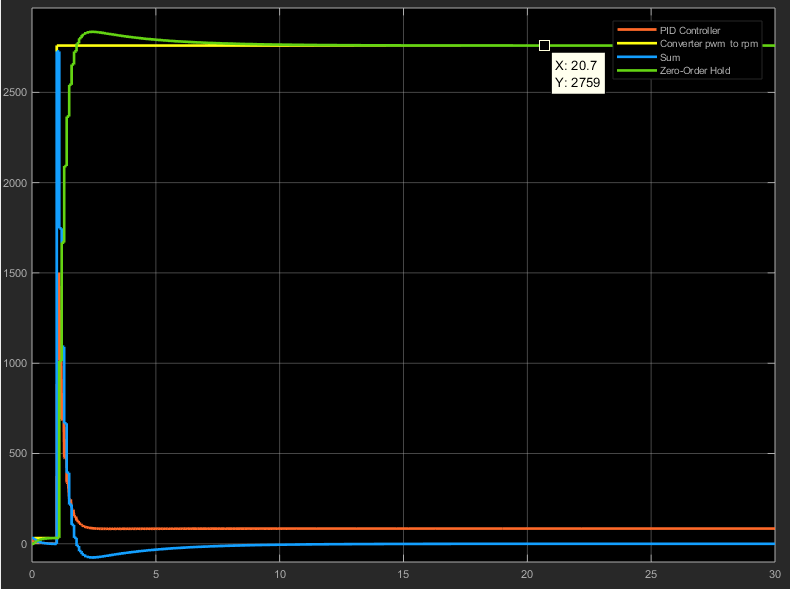
All constants are set to zero. The proportional constant is increased until an answer is obtained that is closest to the desired response. At this point it may appear on impulse or oscillations. Probably the steady state error is different from zero. To correct the steady state error, the integral constant must be adjusted. It is possible that the impulse and oscillations increase. To reduce the over impulse and oscillations the derivative constant is adjusted.

The Figure 20 shows an image capture of the parameters of the tuned controller following the previous steps. The resulting controller is a PID with the parameters and their formula enclosed in the red and green box, respectively as indicated in Figure 20.



**Figure 20. PID Controller obtained.**

It is then observed that in Figure 21 several measured signals are recorded in different parts of the closed loop system shown in the scheme of Figure 17. The red line corresponds to the compensated input signal for the plant, the yellow line is the reference signal, the blue line is the error signal, while the green line is the feedback signal of the system following the reference and reaching the value of 2759 RPM, for a PWM of 80%.



**Figure 21.** Pwm of 80% for a velocity from 2759 rpm.

**Data for characterization of potenciometer**

To find the phenomenon of possible hysteresis, measurements were taken of both the input and output variables of the plant, making variations in the potentiometer and displaying the speed sensor reading both on the display and on the graphic display interface of the e2 studio. First, the ascending variation test was done and then the descending variation test, which consisted of varying the PWM with the potentiometer with a PWM delta of 5% starting from zero to 100% and recording both the value of the input variable as the output value recorded on the display.

It is clearly seen in Figure 22 a certain notable variation in speed between 60 and 70% of PWM is observed, which indicates that the engine of the system when its PWM is decreasing is more revolutionized compared to when its PWM is increasing.

**Figure 22.** Hysteresis pwm vs rpm.

**Results obtained with the controller implemented in the renesas S7G27H3CFC card.**

The Figure 23 shows the behavior of the controller implemented in the speed control system shown in Figure 1. Figure 23 records the behavior of three variables that are: “referenceRPM” corresponding to the system reference, “ controllerRPM ”refers to the signal fed back by the sensor and“ errorRPM ”corresponds to the difference between the previous variables.

**Figure 23.** Results obtained Close loop system using PID controller.