THE CONTROL OF A TOY TRAIN SYSTEM USING THE PID CONTROLLER IN MATLAB / SIMULINK

Lecturer PhD, Gheorghe GÎLCĂ, “Constantin Brâncuși” University from Tîrgu-Jiu, ROMANIA, gheorghe.gilca@yahoo.com
Associate Professor PhD, Ilie BORCOȘI, “Constantin Brâncuși” University from Tîrgu-Jiu, ROMANIA, ilieborcosi@yahoo.com

Abstract: *Model-based systems engineering is widely used in the automotive and avionics domain but less in the railway domain. This paper shows how Matlab / simulink can simulate a toy train system and then control it with the PID control algorithm. To this end, an executable model has been implemented which allows for train movement simulation, this means that the engine first went forward, then backward.*

Keywords: train system, model-based system, velocity, executable model, signal

1. INTRODUCTION

In paper [1] the quarter railway vehicle model is designed as a 5-degree system of freedom and is examined according to variable load combinations and rail irregularities. When controlling the vertical vibrations of the components of the railway vehicle, the performance of a conventional PID and the parameters of an adaptive PID are compared.

In rail vehicle model, the interaction between wheel and rail is not linear [2], but the interaction model is considered as a simplified linear solution using Hertz spring [3, 4]. On the other hand, Dukkipati and Dong study the wheel-rail contact, which is modeled as a multispring contact [5]. Ahmadian and Mohan provide a methodology for improving hunting behaviour in rail vehicles by semiactive control of suspension elements [6]. A sliding mode control (SMC) method to improve the ride comfort of a railway vehicle with a flexible body is represented by Yagiz and Gursel [7].

2. THE IMPLEMENTATION OF A PID CONTROLLER IN SIMULINK

In order to build a model that simulates a toy train system we have started from the work [8] where we found the model created in Simulink, only here the signal generator is replaced by the force F at the input, this representing the force generated between the train engine and 0the railway. Moreover, at the output of the simulated system, instead of the 3 oscilloscopes, we introduced 3 output pins of the model, represented by the following signals: x1_dot (the speed at system exit), x1 (the displacement of the locomotive), x2 (the movement of the wagon). The model created in simulink / matlab is shown in Figure 1. Next, we create a subsystem with this model: we select all blocks in the created model (Ctrl + A), then right-click on the model window and select Create subsystem from the selection menu. Figure 2 shows the created subsystem.

Now we can add a controller to our system. We will use a PID controller that can be implemented using a PID controller block in the Continuous library placing this block in series with the train subsystem. After that, we model the PID controller as directly generating the "F" force. This neglects the dynamics with which the train engine generates the torque applied to the wheels, and then neglects the dynamics of the way the force is generated at the wheel / track interface. We take this simplified approach at this time because we just want to introduce the basic functionality of the Simulink tool for designing and analyzing the controller.

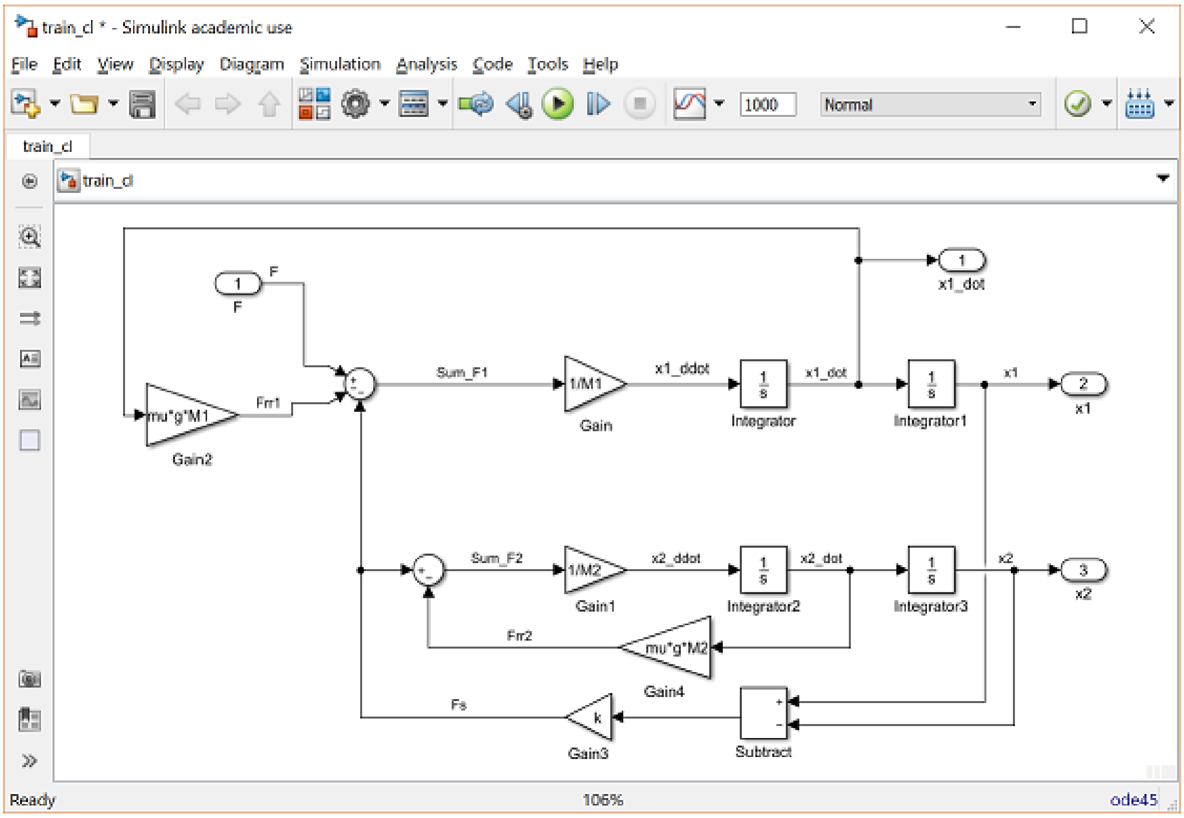


Figure 1. The structure for simulating the toy train system in unity feedback with a PID controller

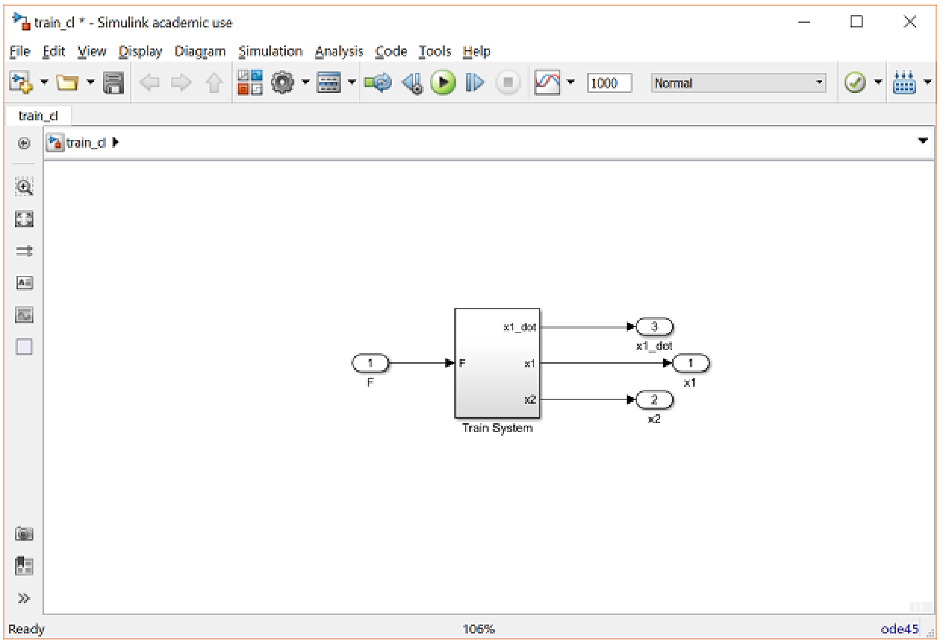


Figure 2. Creating the subsystem for the simulated model

If we double-click the PID Controller block, we will initially set the Integral gain field (I) equal to 0 and we will leave the Proportional (P) and Derivative (D) gains as default values of 1 and respectively, 0.

Next, we add a Block sum from the Math Operations library. Because we want to constantly control the speed of the toy train engine, we will return the engine speed signal. This is done by drawing a line of the x1_dot signal and connecting it to the negative block of the Sum block.

This is done by drawing a line of the x1_dot signal and connecting it to the negative block of the Sum block.

The output of the Sum block must be connected to the input of the PID Controller block, thus adjusting the system. Figure 3 shows the diagram with the added PID controller and system response.

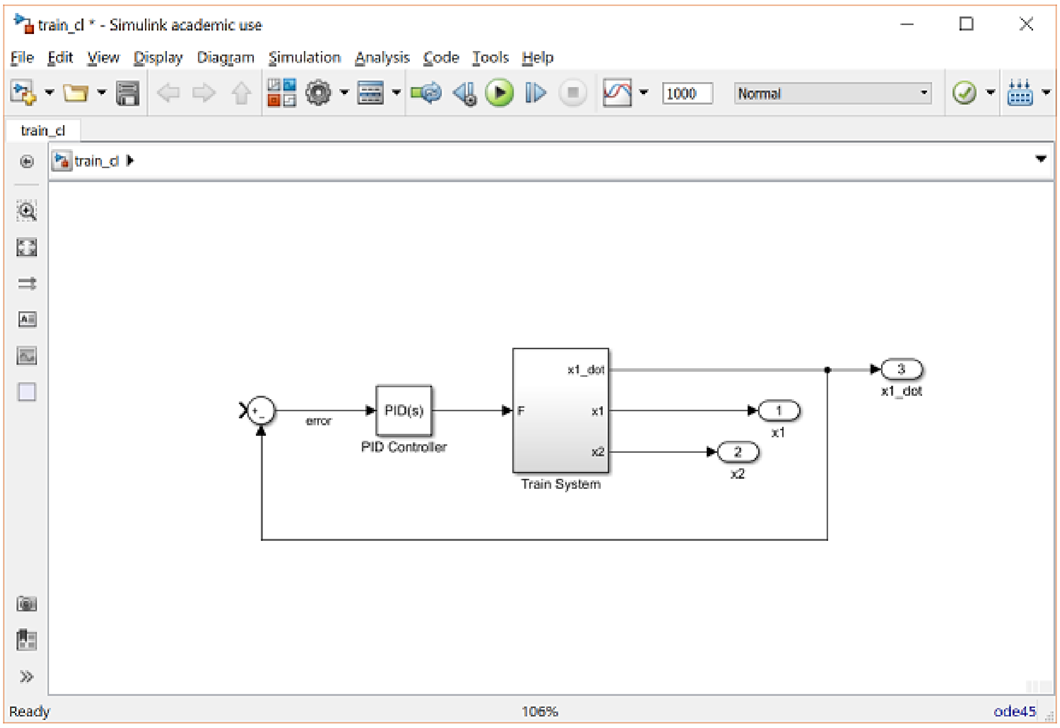
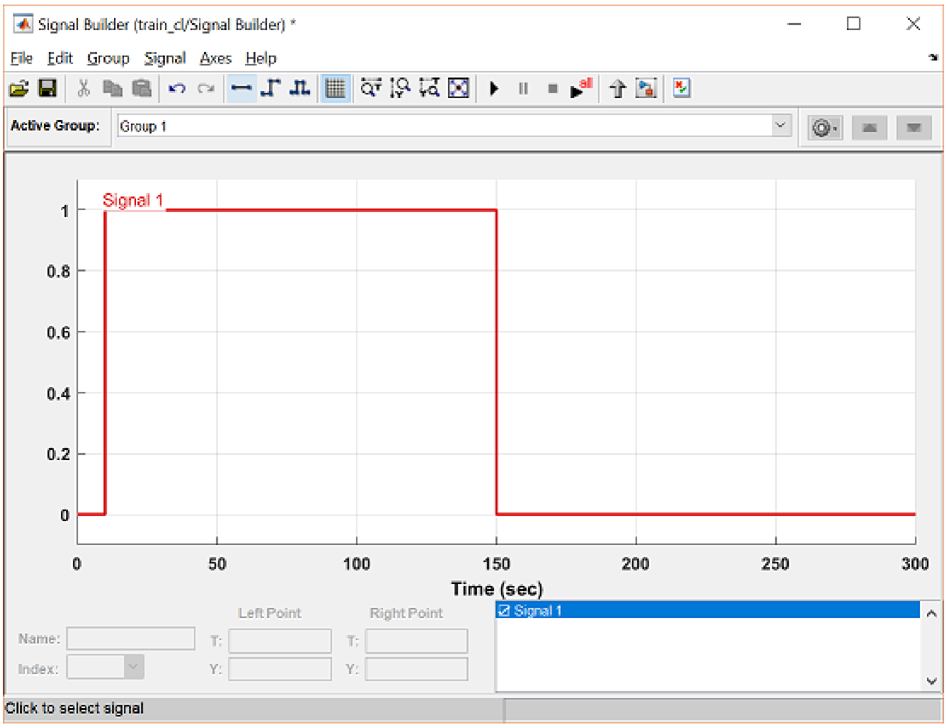


Figure 3. The model with PID controller and reverse feedback to control the speed of the toy train

Next, we add a Signal Builder input block from the Sources library to represent the speed of the train. Because we want to design a controller in order to bring the train up to speed and slow down, we'll test the system with a speed command that goes up to 1 m / s followed by a step back up to 0 m / s (the system, in this case, is a toy train). To generate this type of command signal, we double-click the Signal Builder block. Then we choose Change the time interval from the Axis menu at the top of the block dialog box. We set the Maximum time field to 300 seconds. After that, we set the step up to appear at 10 seconds and the step down to appear at 150 seconds.

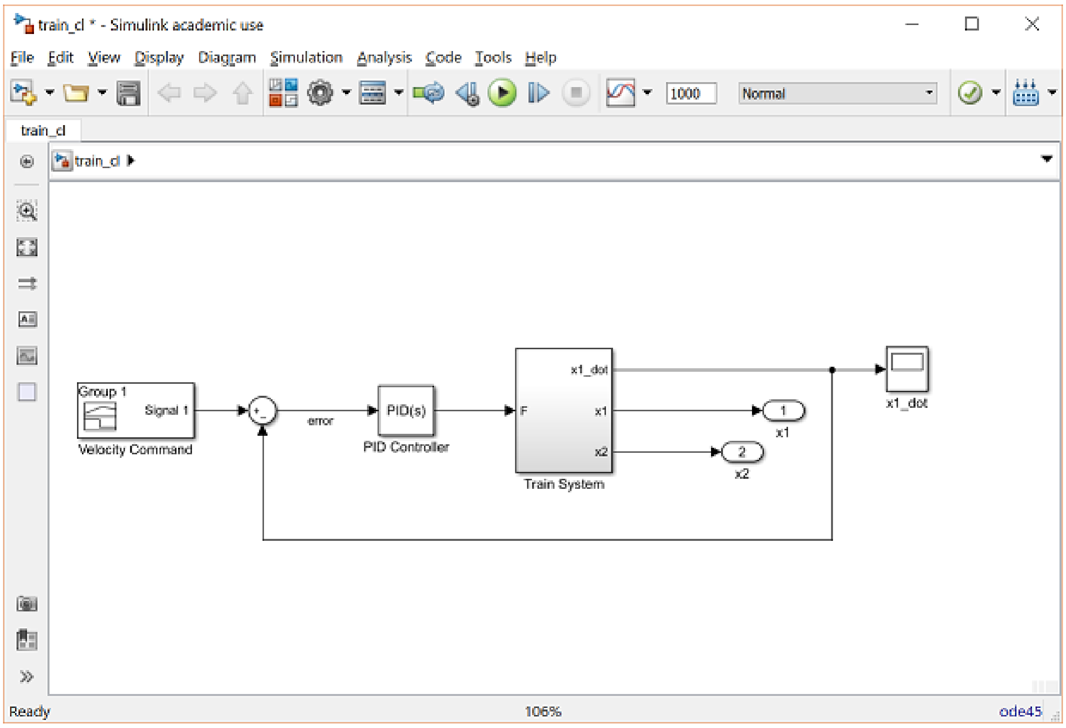
Figure 4 shows the time diagram for the toy train control signal.



Active Group: Group 1

Figure 4. Time diagram for toy train control signal

We also add a Scope block from the Sinks library and use it to replace the block 3 with an oscilloscope to view the output signal. Figure 5 shows the complete block diagram of the simulated model for a toy train.



Simulation stop time: 1000

Simulation mode: Normal

Figure 5. Full block diagram of the toy train simulated model.

3. THE EXECUTION OF THE MODEL IN A CLOSED LOOP

Before running the model, we must assign numeric values to each variable used in the model. For the toy train system we use the following values: M1 = 1 kg, M2 = 0.5 kg, k = 1 N / sec, F = 1 N, μ = 0.02 sec / m, g = 9.8 m / s2.

We create a new m file and enter the following commands:

```
M1 = 1;

M2 = 0.5;

k  = 1;

F  = 1;

mu = 0.02;

g  = 9.8;
```

We execute the m file in the MATLAB command window in order to define these values. Simulink will recognize these MATLAB variables for further use in the model. Then we have to set the time for which our simulation will be run to match the order time in the Signal Builder block. This is done by selecting the model configuration parameters from the Simulation menu at the top of the model window and changing the Stop Time field to 300.

Now, we run the simulation and open the x1_dot field to examine the output of the system (that is, the speed). The result, as shown in Figure 6, demonstrates that the closed-loop system is stable for this controller.

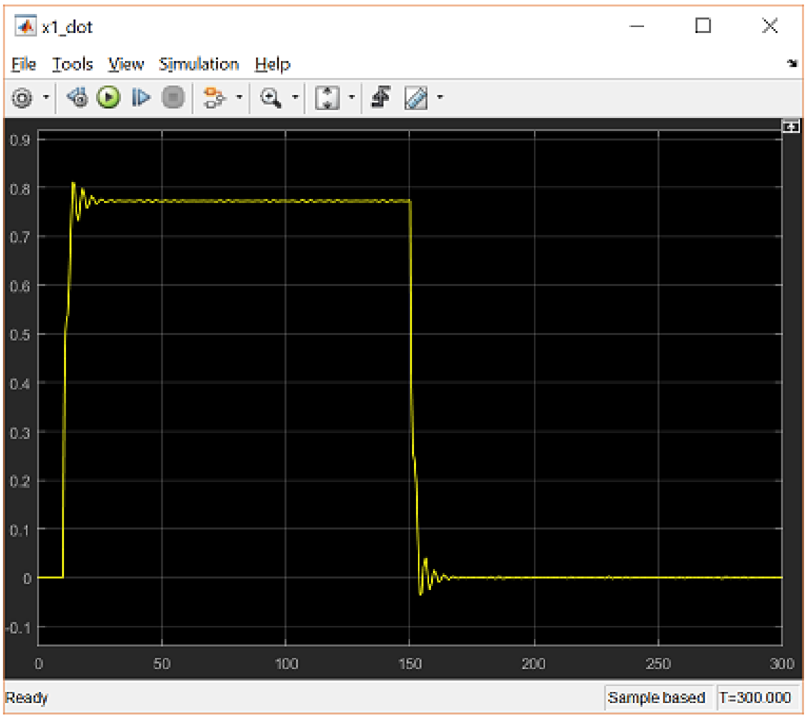


Figure 6. The velocity output of the train system

4. CONCLUSIONS

Simulink can simulate the complete control system, including the control algorithm in addition to the physical plant. Matlab/Simulink is especially useful for generating the approximate solutions of mathematical models that may be prohibitively difficult to solve "by hand." For example, consider that you have a nonlinear plant. A common approach is to generate a linear approximation of the plant and then use the linearized model to design a controller using analytical techniques. Simulink can then be employed to simulate the performance of your controller when applied to the full nonlinear model. Simulink can be employed for generating the linearized model and MATLAB can be employed for designing the controller. Various control design facilities of MATLAB can also be accessed directly from within Simulink.

REFERENCES

[1] M. Muzaffer, G. Rahmi, *Rail Vehicle Vibrations Control Using Parameters Adaptive PID Controller*, Mathematical Problems in Engineering, Hindawi Publishing Corporation, 2014, Article ID 728946, 10 pages, http://dx.doi.org/10.1155/2014/728946.

[2] Y.Q. Sun and M. Dhanasekar, *Adynamic model for the vertical interaction of the rail track and wagon system*, International Journal of Solids and Structures, vol. 39, no. 5, pp. 1337–1359, 2002.

[3] H. H. Jenkins, J. E. Stephenson, G. A. Clayton, G.W.Morland, and D. Lyon, *The effect of track and vehicle parameters on wheel/rail vertical dynamic forces*, Railway Engineering Journal, vol. 3, no. 1, pp. 2–16, 1974.

[4] C. Esveld, *Modern Railway Track*, MRT Productions, Zaltbommel, The Netherlands, 2001.

[5] R. V. Dukkipati and R. Dong, *The dynamic effects of conventional freight car running over a dipped-joint*, Vehicle System Dynamics, vol. 31, no. 2, pp. 95–111, 1999.

[6] M. Ahmadian and A. Mohan, *Semiactive control of hunting stability in rail vehicles*, in *Proceedings of the ASME* International Mechanical Engineering Congress and Exposition (IMECE ’05), pp. 63–68, Orlando, Fla, USA, November 2005.

[7] N. Yagiz and A. Gursel, *Active suspension control of a railway vehicle with a flexible body*, International Journal of Vehicle Autonomous Systems, vol. 3, no. 1, pp. 80–95, 2005.

[8] Gîlcă G., *The modelling of a train system using matlab/Simulink*, Fiability & Durability No 2/ 2018, Editura “Academica Brâncuşi” , Târgu Jiu, ISSN 1844 – 640X, pp 132-137.