IOSR Journal of Mechanical and Civil Engineering (IOSR-JMCE)
e-ISSN: 2278-1684,p-ISSN: 2320-334X, Volume 14, Issue 4 Ver. VI (Jul. - Aug. 2017), PP 12-22
www.iosrjournals.org

Controlling Speed of Hybrid Cars using Digital Internal Model Controller

*HeshamAL Salem[1], Ayman M. Mansour[2]
[1]*Mechanical Engineering Department,Tafila Technical University,Tafila, 66110,JORDAN*
[2]*Communication, Electronics and Computer Engineering Department,Tafila Technical University,Tafila, 66110, JORDAN*
Corresponding Author: Hesham AL Salem

Abstract: *Controlling the speed of motor in hybrid vehicles plays a key role in maintaining the stability and smoothness ofcar movement. In this paper a new methodology of digital internal model controller (DIMC) has been designed and tuned tomaintain a selective operating speed by optimizing the maximum overshoot and settling time. In this paper the performance of the proposed DIMC has been investigated and compared with other techniques such as PID controller, fuzzy PID, fuzzy PI, and observer based controller. The designed controller will enhance the performance of speed controlling of hybrid car which leads to an improvement in fuel efficiency.*

Date of Submission: 18-01-2017 Date of acceptance: 17-08-2017

I. Introduction

Global warming issue erg researchers to explore a more efficient and better utilization of energy resources. Public transportation plays a major role in increasing the CO_x and NO_x which contributes hugely to global warming issue. Hybrid vehicles can maximize the utilization of all possible power resources if they are well designed and controlled [1-7]. Vehicles can get its power from different methods based on the amount of both gas and electricity in order to achieve either better fuel economy or higher power output [8-12]. Electric motor drive assist vehicles to accelerate and have more power when needed in different running speeds and operating conditions [1,3,4,6,7,11].In this paper we will use MatLABSimulink to design a digital controller to control a DC motor in order to enhance the performance and stability of the speed of motor. Electric machines are essential systems in electric vehicles and are widely used in other applications. In particular, permanent magnet direct current (PMDC) motors have been extensively employed in industrial applications such as battery powered devices like wheelchairs and power tools, guided vehicles, welding equipment, X-ray and tomographic systems, CNC machines, etc. PMDC motors are physically smaller in overall size and lighter for a given power rating than induction motors. The unique features of PMDC motors, including their high torque production at lower speed, flexibility in design, make them preferred choices in automotive transmissions, gear systems, lower-power traction utility, and other fields.

The stability, robust, and short rise time are needed in motor systems [3,4]. Digital controllers are far more convenient to implement on microprocessors than are continuous-time controllers. Continuous-time controllers must be implemented either using analog circuitry (op amps). Discrete-time controllers, on the other hand, are easily implemented using simple computer software.

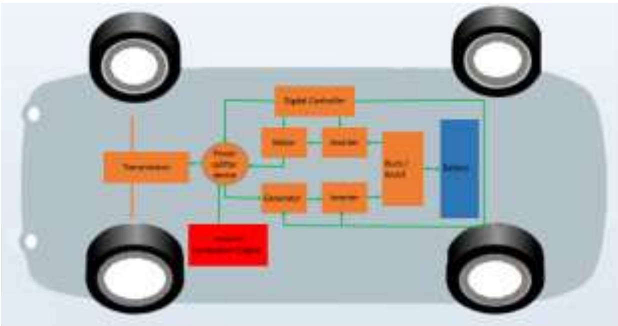

Fig1: Hybrid car framework

Two methods are available for design of digital controllers:

• Discretize the continuous plant, either the state-space model or the transfer function, to obtain a DT system. Use that DT system to design a DT controller.
• Design a CT controller, and then discretize that to obtain a DT controller.
The objective of this paper is to provide smooth movement and zero steady state speed error for a selected speed by minimizing the overshoot and optimizing the settling time through turning the controller
Here we will focus on the second method. Given a continuous-time controller, designed by any technique (PID). We will show how to convert it to a digital controller. The controller will be designed by a Root Locus method.
A digital DC motor model can obtain from conversion of analog DC motor model. A proportional–integral–derivative controller (PID controller) is a generic control loop feedback mechanism (controller) widely used in industrial control systems; The PID controller algorithm involves three separate constant parameters, and is accordingly sometimes called three-term control: the proportional, the integral and derivative values, denoted P, I, and D.

1.1 Physical System Background

Typical models for a DC motor contains one differential equation for the electric part, one differential equation for the mechanical part, and their interconnections. This is valid for the PMDC motors as well.
The motor torque, T, is related to the armature current, i, by a constant factor Kt. The back emf, e, is related to the rotational velocity by the following equations:

$$T=K_t i$$
$$E=K_e \Theta'$$

In SI units (which we will use), Kt (armature constant) is equal to Ke (motor constant).
From the figure above we can write the following equations based on Newton's law combined with Kirchhoff's law:

$$J\,\Theta''+b\,\Theta'=K\,i$$
$$L\frac{di}{dt}+Ri=V-K\Theta'$$

1.2 Transfer Function

Using Laplace Transforms, the above modeling equations can be expressed in terms of s.

$$S\,(Js+b)\,\Theta(s)=KI(s)$$
$$(Ls+R)\,I(s) = V-Ks\Theta(s)$$

By eliminating I(s) we can get the following open-loop transfer function, where the rotational speed is the output and the voltage is the input.

$$\frac{\Theta}{v}=\frac{K}{(Js+b)(Ls+R)+K^2}$$

1.3 State-Space Equations

In the state-space form, the equations above can be expressed by choosing the rotational speed and electric current as the state variable and the voltage as an input.
The output is chosen to be the rotational speed.

$$\frac{d}{dt}\begin{bmatrix}\dot{\theta}\\ i\end{bmatrix}=\begin{bmatrix}-\frac{b}{J} & \frac{K}{J}\\ -\frac{K}{L} & -\frac{R}{L}\end{bmatrix}\begin{bmatrix}\dot{\theta}\\ i\end{bmatrix}+\begin{bmatrix}0\\ \frac{1}{L}\end{bmatrix}V$$

$$\dot{\theta}=\begin{bmatrix}1 & 0\end{bmatrix}\begin{bmatrix}\dot{\theta}\\ i\end{bmatrix}$$

2.1 Digital Controller Design Methodology

In practice controllers are nowadays almost exclusively implemented digitally. This means that the controller operates in discrete time, although the controlled systems usually operate in continuous time.

Digital controllers are far more convenient to implement on microprocessors than are continuous-time controllers. Continuous-time controllers must be implemented either using analog circuitry. Discrete-time controllers, on the other hand, are easily implemented using difference equations, i.e. simple computer software.

Two methods are available for design of digital controllers:
• Discretize the CT plant, either the state-space model or the transfer function, to obtain a DT system. Use that DT system to design a DT controller.
• Design a CT controller, then discretize that to obtain a DT controller.
Here we will focus on the second method. Given a continuous-time controller, designed by any technique (root locus, PID, lead, lag, etc.), and show how to convert it to a digital controller.

2.2 Discretization of Continuous-Time Controllers Methodology
By any of a variety of techniques, one may design a continuous-time compensator K(s). This may be converted to digital form K(z) using several techniques, among the most direct of which is the bilinear transformation (BLT).
The relation between the Laplace transform variable s and the Z-transform variable z is z=esT, with T the sampling period. However, using this to transform K(s) to K(z) will give non-polynomial transfer functions. Note that

$$e^{sT} \approx \frac{1+\frac{sT}{2}}{1-\frac{sT}{2}}$$

Therefore define the BLT by

$$z = \frac{1+\frac{sT}{2}}{1-\frac{sT}{2}}$$

and its inverse

$$S = \frac{2}{T}\frac{Z-1}{Z+1}$$

To convert a continuous transfer function G(s) to a discrete transfer function usingsample period T.
Then, one simply replaces all occurrences of s by
The continuous-time PID controller can be written in the form

$$G_c(s) = Kc[1 + \frac{1}{\tau_I s} + \tau_D S]$$

where Ti is the integration time constant or 'reset time', Td is the derivative time constant,

To convert this to digital form using the BLT, write

$$G_c(z) = Kc[1 + \frac{1}{\tau_I\left(\frac{2}{T}\frac{z-1}{z+1}\right)} + \tau_D\left(\frac{2}{T}\frac{z-1}{z+1}\right)]$$

This may be simplified to obtain

$$G_c(z) = Kc[1 + \frac{T(Z+1)}{\tau_I(2(z-1))} + \tau_D\left(\frac{2}{T}\frac{z-1}{z+1}\right)]$$

$$Then$$

$$G_c(z) = Kc[1 + \frac{T(Z+1)}{\tau_{ID}(z-1)} + \frac{\tau_{DD}}{T}\left(\frac{z-1}{z+1}\right)]$$

where the digital integral and derivative time constants are

$$\tau_{ID} = 2\tau_I$$
$$\tau_{DD} = 2\tau_D$$

3.1 Planet modeling

We can express the DC speed motor system by the following diagrams

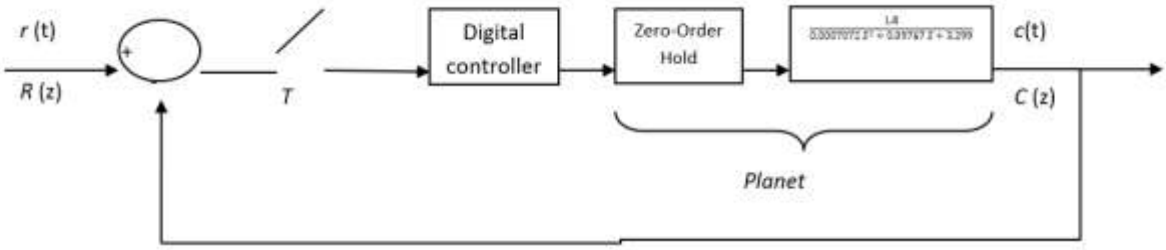

Fig 2: Digital Control System of the Motor

3.2 Design requirements

The most basic requirement of a motor is that it should rotate at the desired speed; the steady-state error of the motor speed should be less than 1%. The other performance requirement is that the motor must accelerate to its steady-state speed as soon as it turns on. In this case, we want it to have a settling time of 2 seconds. Since a speed faster than the reference may damage the equipment, we want to have an overshoot of less than 5%.
If we simulate the reference input (r) by an unit step input, then the motor speed output should have:

- Settling time less than 2 seconds
- Overshoot less than 5%
- Steady-state error less than 1%

3.3 Planet Transfer Function in Continues and Discrete (open loop)

$$G(s) = \frac{1.8}{0.0007072S^2 + 0.09767\,S + 3.299}$$

To find G(z) using ZOH cascaded with the planet

$$G(Z) = \zeta\left\{\frac{1 - e^{st}}{s} . \frac{1.8}{0.0007072S^2 + 0.09767\,S + 3.299}\right\}$$

$G(Z) =$

$$(1 - z^{-1})\zeta\left\{\frac{1.8}{s(0.0007072S^2 + 0.09767\,S + 3.299)}\right\}$$

The first part is $\frac{(z-1)}{z}$
From the table we can get the

$$\zeta\left\{\frac{1.8}{s(0.0007072S^2 + 0.09767\,S + 3.299)}\right\}$$

Then after simplification:

$$G(z) = \frac{0.4607\,z^4 + 1.161z^3 + 0.9688z^2 + 0.2974\,z + 0.02894}{z^4 + 0.52z^3 - 0.9372z^2 - 0.52z - 0.06285}$$

3.4 Continuous Time PID Controller for continues signal Design Procedure

More practical approach is to specify the closed loop transfer function so the realistic setting time is achieved. Design of continuous PID controller for continuous signal, If the system of second order as

$$G(s) = \frac{K}{(\tau_1 S + 1)(\tau_2 S + 1)}$$

$$G_C(s) = \frac{\tau 1 + \tau 2}{K\tau_c}\left[1 + \frac{1}{(\tau 1 + \tau 2)s} + \frac{\tau 1\ \tau 2}{\tau 1 + \tau 2}s\right]$$

Comparing with

$$G_c(s) = Kc[1 + \frac{1}{\tau_I s} + \tau_D S]$$

Where

Kc = $\frac{\tau 1+\tau 2}{K\tau_c}$

$\tau I = (\tau 1 + \tau 2)$

$\tau D = \frac{\tau 1\ \tau 2}{\tau 1 + \tau 2}$

Comparing the used motor transfer function to the previously mentioned TF.

$$\boldsymbol{G(s) = \frac{K}{(\tau_1 S + 1)(\tau_2 S + 1)}}$$

$$\boldsymbol{G(s) = \frac{K}{\tau_1\tau_2 S^2 + (\tau_1 + \tau_2)S + 1}}$$

$$\boldsymbol{G(s) = \frac{1.8}{0.0007072 S^2 + 0.09767 S + 3.299}}$$

Divide by 3.299 the numerator and denominator to make the coefficient of the lowest power is 1 in order to use the controller equation mentioned above.

$$\boldsymbol{G(s) = \frac{\frac{1.8}{3.299}}{\frac{0.007072}{3.2999}S^2 + \frac{0.09767}{3.299}S + 1}}$$

Comparing the two G(s) equations

K=1.8/3.299=0.54562

$$\boldsymbol{\tau_1\tau_2 = \frac{0.007072}{3.2999} = 2.14368 * 10\text{^} - 4}$$

$$\boldsymbol{\tau_1 + \tau_2 = \frac{0.09767}{3.299} = 2.96059 * 10\text{^} - 2}$$

Kc = $\frac{\tau 1+\tau 2}{K\tau_c}$=$\frac{0.029605}{0.54562\,\tau_c} = 1\ when \tau_c = \frac{0.54562}{0.029605} = 18.43$

Where τ_c is design parameter

$\tau I = (\tau 1 + \tau 2)$=0.029605

$\tau D = \frac{\tau 1\ \tau 2}{\tau 1+\tau 2}$=$\frac{0.0214368}{0.029605}$=7.2404*10^-3

$$G_c(s) = [1 + \frac{1}{0.029605s} + (7.2404 * 10\hat{} - 3)S]$$

3.4 Discretize the system using Bilinear Transformation

To convert this to digital form using the BLT, write

$$G_c(z) = Kc[1 + \frac{T(Z+1)}{\tau_{ID}(z-1)} + \frac{\tau_{DD}}{T}\left(\frac{z-1}{z+1}\right)]$$

where the digital integral and derivative time constants and the sampling period are

$\tau_{ID} = 2\tau_I$=2*0.029605=0.05921

$\tau_{DD} = 2\tau_D = 2$ *7.2404*10^-3=0.0144808

T=0.05

Kc=1

Then

$$G_c(z) = [1 + \frac{0.05(Z+1)}{0.05921(z-1)} + \frac{0.0144808}{0.05}\left(\frac{z-1}{z+1}\right)]$$

After Simplification the controller Transfer Function:

$$\boldsymbol{G_c(z) = \frac{2.134z^2 + 1.11z + 0.1341}{z^2 - 1}}$$

3.5 The Closed Loop Discrete Time System

$$\boldsymbol{C(z)} = \frac{\boldsymbol{G_c(z)G\ \ (z)}}{\boldsymbol{1 + G_c(z)G\ \ (z)}}\boldsymbol{R(z)}$$

We would like to see what the closed-loop response of the system looks like when no controller is added. First, we have to close the loop of the transfer function by using the feedback command. After closing the loop, let's see how the closed-loop stair step response performs by using the stepand stairs commands. The step command will provide the vector of discrete step signals and stairs command will connect these discrete signals.

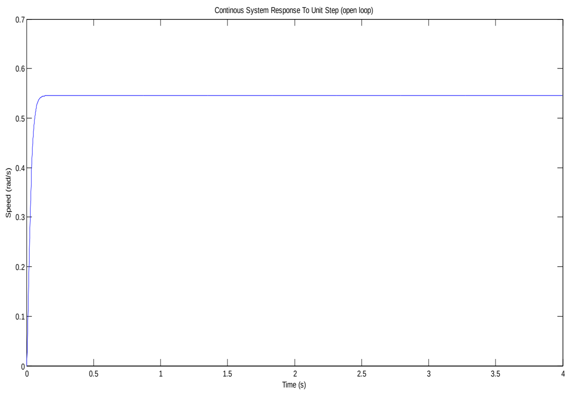

Fig 3: The Continues System Response to unit step (open system)

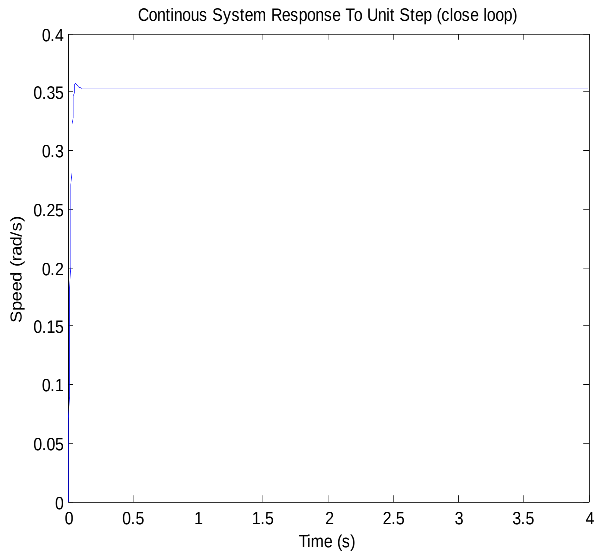

Fig 4: The Continues System Response to unit step (Close system)

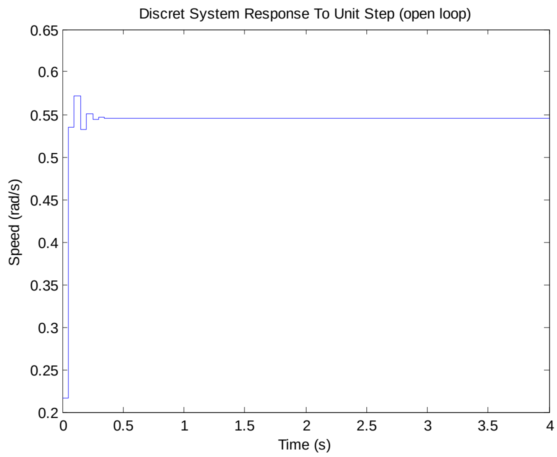

Fig 5: The Discrete System Response to unit step (Open loop system)

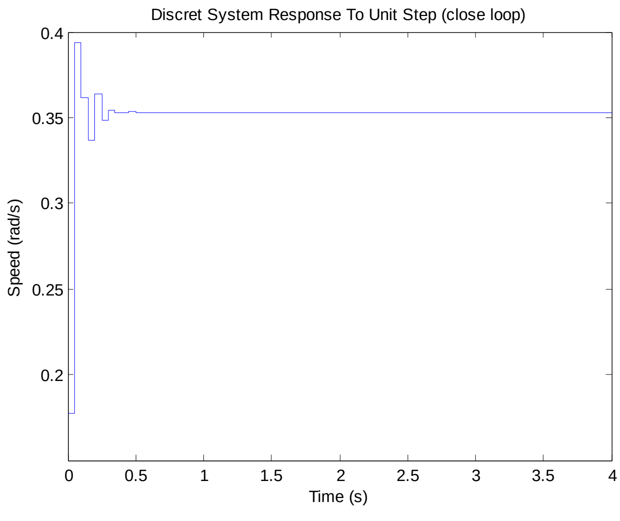

Fig 6: The Discrete System Response to unit step (Close loop system)

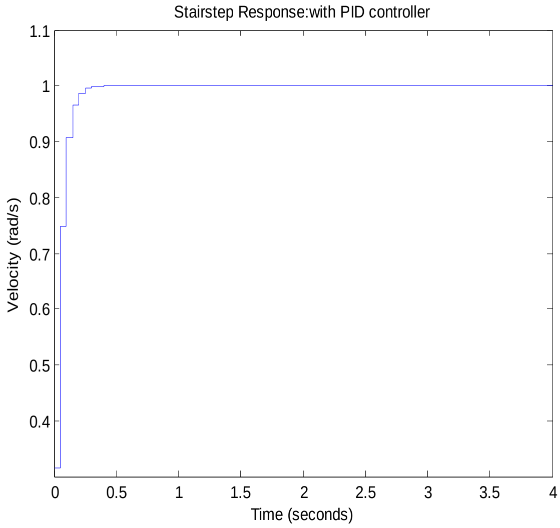

Fig 7: The Discrete System Response to unit step (Close system with PID controller)

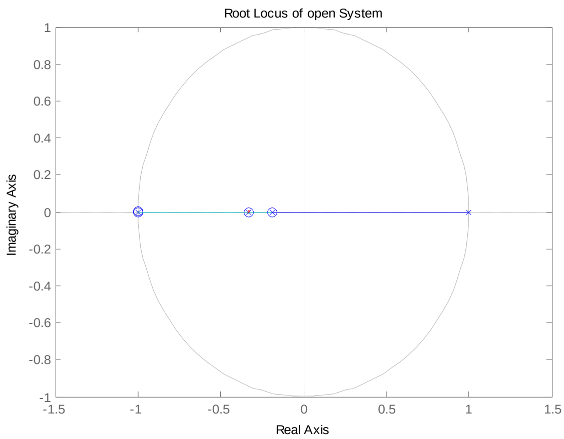

Fig 8: The Root Locus of the Open system

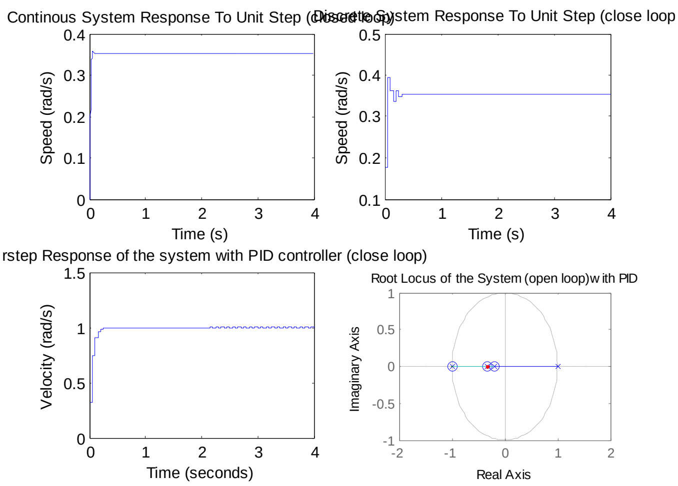

Fig 9: Summarized Results

From the plot original open-loop system performance we see that when 1 volt is applied to the system, the motor can only achieve a maximum speed of 0.1 rad/sec, ten times smaller than our desired speed. Also, it takes the motor 3 seconds to reach its steady-state speed; this does not satisfy our 2 seconds settling time criterion. The plot above shows that the settling time is less than 2 seconds and the percent overshoot is around 2%. Additionally, the steady-state error is zero. Therefore, this response satisfies all of the given design requirements.

4.3 Simulink PID Design and Simulation

4.3.1 Simulink Block Diagram

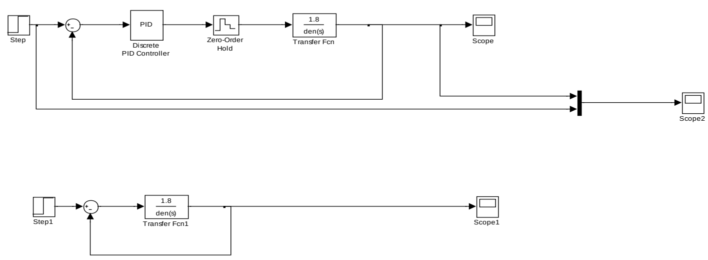

Fig 10: The Simulink Block Diagram of Discrete system with PID controller and the Continuous system

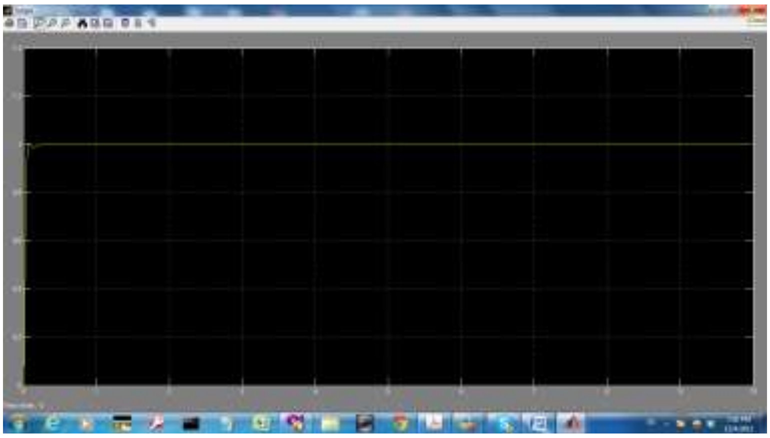

Fig 15: Step Input Response Result with PID controller.

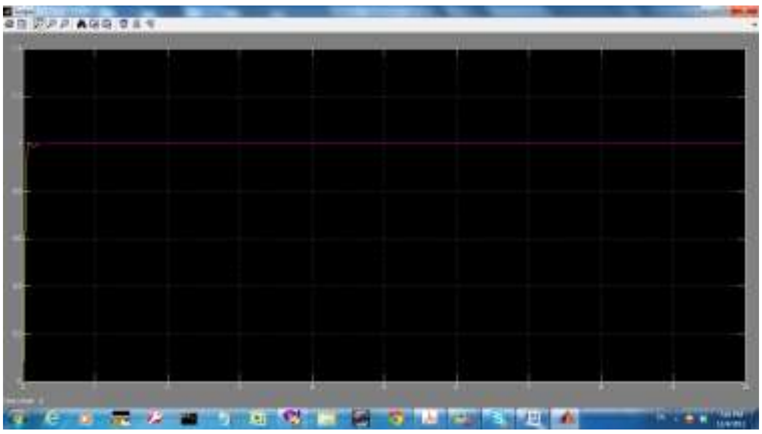

Fig 16: Step Input Response Result with PID controller besides the Step Input.

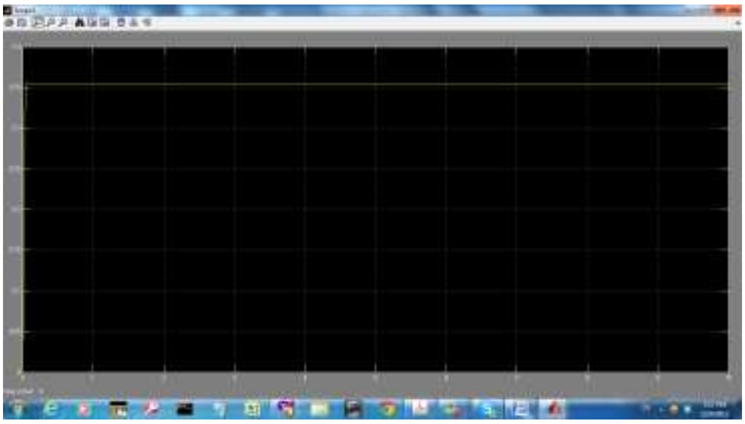

Fig 17: Step Input Response Result without PID controller.

4.3.3 Simulink in simplified form

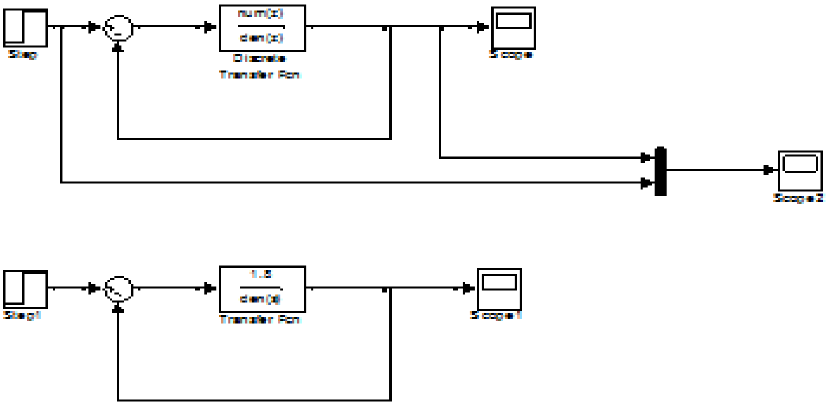

Fig 18: Simulink block Diagram

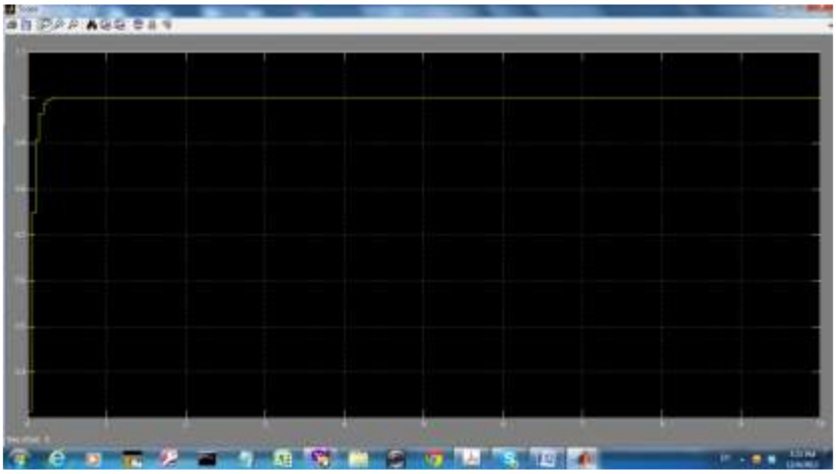

Fig 21: Step Response of the closed system with PID controller.

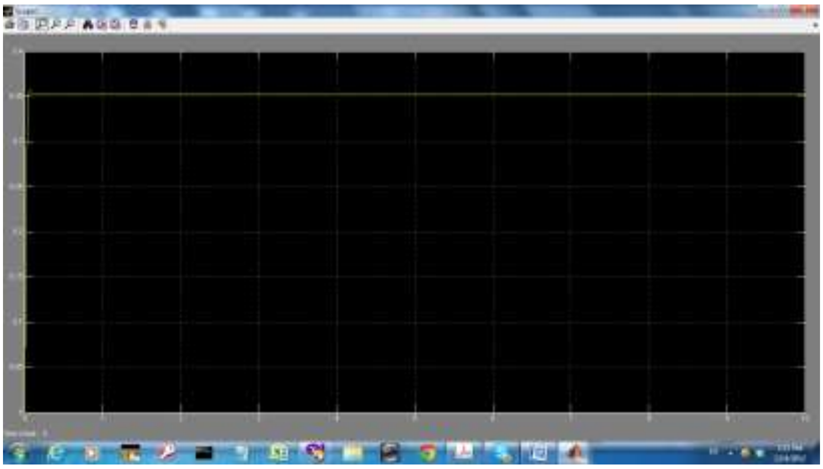

Fig 22: Step Response of the open loop system .

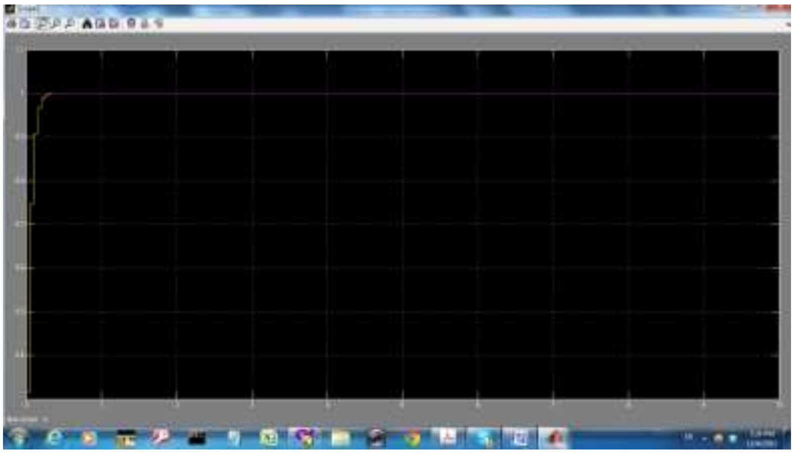

Fig 23: Step Response of the close loop system with step input.

The result of the simplified form is the same as the original system. From the plot original open-loop system performance we see that when 1 volt is applied to the system, the motor can only achieve a maximum speed of 0.1 rad/sec, ten times smaller than our desired speed. Also, it takes the motor 3 seconds to reach its steady-state speed; this does not satisfy our 2 seconds settling time criterion. The plot above shows that the settling time is less than 2 seconds and the percent overshoot is around 2%. Additionally, the steady-state error is zero. Therefore, this response satisfies all of the given design requirements.

II. Conclusion

In this paper we have usedMATLAB Simulink to design a digital PID controller to control a DC motor (continues time transfer function). Electric machines are essential systems in electric vehicles and are widely used in other applications. In particular, permanent magnet direct current (PMDC) motors have been extensively employed in industrial applications such as battery powered devices like wheelchairs and power tools, guided vehicles, welding equipment, X-ray and tomographic systems, CNC machines, etc. PMDC motors are physically smaller in overall size and lighter for a given power rating than induction motors. Because of this controlling such motors is very important. In this paper we used a digital PID controller to control the speed of the motor. Digital controllers are far more convenient to implement on microprocessors than are continuous-time controllers. Continuous-time controllers must be implemented either using analog circuitry (op amps). Discrete-time controllers, on the other hand, are easily implemented using simple computer software. Two methods are available for design of digital controllers. Discretize the continuous plant is the first used method, either the state-space model or the transfer function, to obtain a DT system. Use that DT system to design a DT controller. The second method is to Design a CT controller, and then discretize that to obtain a DT controller. Here we focus on the second method. Given a continuous-time controller, designed by any technique (PID). A proportional–integral–derivative controller (PID controller) is a generic control loop feedback mechanism (controller) widely used in industrial control systems. The PID controller algorithm involves three separate constant parameters, and is accordingly sometimes called three-term control: the proportional, the integral and derivative values, denoted P, I, and D.

The result of the simplified form is the same as the original system. From the plot original open-loop system performance we see that when 1 volt is applied to the system, the motor can only achieve a maximum speed of 0.1 rad/sec, ten times smaller than our desired speed. Also, it takes the motor 3 seconds to reach its steady-state speed; this does not satisfy our 2 seconds settling time criterion. The plot above shows that the settling time is less than 2 seconds and the percent overshoot is around 2%. Additionally, the steady-state error is zero. Therefore, this response satisfies all of the given design requirements.

References

1. Sequeira, J. and M.I. Ribeiro. Hybrid control of a car-like robot. in Proceedings of the Fourth International Workshop on Robot Motion and Control (IEEE Cat. No.04EX891). 2004.
2. Shahverdi, M., et al., Bandwidth-Based Control Strategy for a Series HEV With Light Energy Storage System. IEEE Transactions on Vehicular Technology, 2017. 66(2): p. 1040-1052.
3. Stiegeler, M., et al. Influence of battery size on predictive control for hybrid cars. in Melecon 2010 - 2010 15th IEEE Mediterranean Electrotechnical Conference. 2010.
4. Tchenderli-Braham, S.A. and F. Hamerlain. Trajectory tracking with a hybrid control applied to a bi-steerable car. in 2nd International Conference on Systems and Computer Science. 2013.
5. X. Q, L. and W.F. Jiang. The structures and the energy management strategies in FCHVs. in 2016 IEEE Workshop on Advanced Robotics and its Social Impacts (ARSO). 2016.
6. Xiaogang, W., et al. Design of vehicle controller for ISG hybrid electric car based on fuzzy control. in Proceedings of 2011 6th International Forum on Strategic Technology. 2011.
7. Ebbesen, S., et al., Time-optimal Control Strategies for a Hybrid Electric Race Car. IEEE Transactions on Control Systems Technology, 2017. PP(99): p. 1-15.
8. Hace, A. and M. Golob. Chattering-free sliding mode control algorithm for a haptic throttle lever. in IECON 2016 - 42nd Annual Conference of the IEEE Industrial Electronics Society. 2016.
9. Jeong, J., et al. Fuel economy analysis of a parallel hybrid bus using the optimal control theory. in 2011 IEEE Vehicle Power and Propulsion Conference. 2011.
10. Kasinathan, D., et al., An Optimal Torque Vectoring Control for Vehicle Applications via Real-Time Constraints. IEEE Transactions on Vehicular Technology, 2016. 65(6): p. 4368-4378.
11. Xu, C.D. and K.W.E. Cheng. All-electric intelligent anti-lock braking controller for electric vehicle under complex road condition. in 2016 International Symposium on Electrical Engineering (ISEE). 2016.
12. Yerrawar, R.N. and R.R. Arakerimath. Performance assessment and control policies for semiactive suspension using SIMSCAPE. in 2016 International Conference on Automatic Control and Dynamic Optimization Techniques (ICACDOT). 2016.

Controlling Speed of Hybrid Cars using Digital Internal Model Controller

Dr.Hesham I. Alsalem is currently an Assistant Professor and chair of the Mechanical Engineering Department at Tafila Technical University, Jordan. He earned his PhD in Mechanical Engineering from Wayne State University, Detroit, Michigan, USA, in 2016. He also received the BSc degree in Mechanical Engineering in 1996 from Jordan University of Science and Technology, Irbid, Jordan and his MSc degrees in Mechanical Engineering in 1999 from Jordan University of Science and Technology, Irbid, Jordan. His research experience and interests are in combustion emission control of SI engines, Hybrid cars,Intelligent mechanical systems, Nanotechnologyand Lithiumsulfur batteries. He is a Member of the American Society of Mechanical Engineers (ASME), Society of Automotive Engineers(SAE), and Jordan Engineers Association (JEA).

Dr.Ayman M Mansour received his Ph.D. degree in Electrical Engineering from Wayne State University in 2012. Dr. Mansour received his M.Sc degree in Electrical Engineering from University of Jordan, Jordan, in 2006 and his B.Sc degree in Electrical and Electronics Engineering from University of Sharjah, UAE, in 2004. He graduated top of his class in both Bachelor and Master. Currently, Dr. Mansour is an assistant professor in the Department of Communication and computer Engineering, Tafila Technical University, Jordan. He is also the director of energy research centre in Tafila Technical University. His areas of research include communication systems, multi-agent systems, fuzzy systems, data mining and intelligent systems. He conducted several researches in his area of interest. Dr. Mansour is a member of IEEE, Michigan Society of Professional Engineers, IEEE Honor Society (HKN), Society of Automotive Engineers (SAE), Tau Beta Pi Honor Society, Sigma Xi and Golden Key Honor Society.

Hesham AL Salem. “Controlling Speed of Hybrid Cars using Digital Internal Model Controller .” IOSR Journal of Mechanical and Civil Engineering (IOSR-JMCE) , vol. 14, no. 4, 2017, pp. 12–22.