The Scientific Temper (2023) Vol. 14 (1): 178-183
Doi: 10.58414/SCIENTIFICTEMPER.2023.14.1.21
E-ISSN: 2231-6396, ISSN: 0976-8653
https://scientifictemper.com/
RESEARCH ARTICLE

Development of digital twin for PMDC motor control loop

Pravin P. P, J. Arunshankar*

Abstract

Recent developments in the digital domain can reduce the resources needed for efficient plant control. The growth of cloud computing and big data analysis has paved the way for using digital domain as processing units. Metaverse is the animated version of the physical realm in digital domain. The control mechanism and dynamics of the real-world data cannot be manually fed to the digital domain. The real plant can change its dynamics during its operation which the previously modeled mathematical model cannot address. So, the model for the system needs to be developed by its own and it need to be adaptable. This can be done by implementing the digital twin. In this work, a nonlinear Auto Regressive Exogenous model captures the dynamics of a PMDC motor. A model predictive approach is used to control the PMDC motor, which uses updated model to predict the response and generate the desired control input.

Keywords: Digital twin, Nonlinear Auto Regressive Exogenous model, Predictive model controller, Particle swarm optimization

Introduction

The PMDC motor is used in the field of delicate applications. A digital twin is the digital representation of the system. But the flow of information between the digital model and real system decides whether the digital representation is a digital twin or not. If the information flows from model to plant and then plant to model it is called a digital twin. The process cycle of the PMDC motor can be monitored and useful data can be collected for future robust design of the machine. The variation of the parameters during the operation of the system makes it difficult to control the plant optimally. Digital twin helps us control the system optimally even in parameter variation[4].

The digital capability has grown enormous. The digital domain greatly reduces the need for real-world resources. With the development of cloud computing technology and metaverse, the need for digital representation of the system has increased. The digital domain acts as a bridge between disciplines. The non-technical entities of the system can be used to tune technical aspects of the system. Modeling of the system to the desired level of accuracy makes the digital twin efficient and useful. The nonlinear nature of the model also needs to be modeled and the model is to be updated for uncertainties that take place when the system is being controlled. To improve the control performance of the closed-loop system, there should be a flow of information from the digital twin to the controller. The received information must be effectively used to change the controller parameters to get a good response (Kritzinger et al., 2018).

The digital twin is implemented by the following. The NARX model helps to model the nonlinearities in the system. The expert systems' growth helps make the model more adaptive to the uncertainties. The NARX model can be implemented using different techniques. By using the neural network, can implement the NARX model. The optimization algorithm achieves adaptability. The proposed work's primary contribution is the creation of the digital twin for the PMDC motor.

Werner Kritzinger et al. used a categorical analysis of the DT in manufacturing and a categorization of the current publications according to their amount of DT integration to give a review on digital twins[1]. In their study on digital twins, Giovanni Lugaresi and Andrea Matta described a method for automatically recognizing manufacturing systems, developing appropriate digital twins, and extracting crucial information about a production system from data logs (Lugaresi and Matta, 2021). The creation of a digital twin for machining operations based on planning

Department of Instrumentation and Control Systems Engineering, PSG College of Technology Coimbatore, Tamil Nadu India
***Corresponding Author:** Pravin, P. P., Arunshankar, J., Department of Instrumentation and Control Systems Engineering, PSG College of Technology Coimbatore, Tamil Nadu India, E-Mail: 20mo07@psgtech.ac.in

How to cite this article: Author. (2023). Article Title. The Scientific Temper, **14**(1):178-183
Doi: 10.58414/SCIENTIFICTEMPER.2023.14.1.21
Source of support: Nil
Conflict of interest: None.

Received: 15/01/2023 **Accepted:** 18/02/2023 **Published** : 25/03/2023

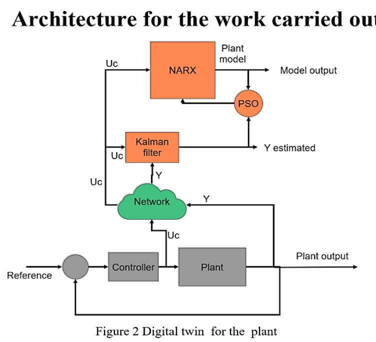

Figure 1: Digital twin for PMDC

and process data was reviewed by Hanel et al. and made it feasible to spot issues early on in the machining process (Hanel, 2020). Brylina *et al.*, presented a review in the importance of modeling in digital twins with an example of a PID controller model and cycle time of the PID controller (Brylina, N. N. Kuzmina, and K. V. Osintsev, 2020).

Duan and Tian talked about the digital twin's design, the model's maturity, and how the model learns once it gets going will not learn immediately about the complete aspect of the system (Duan and Tian, 2020). Naung et al offered an overview of the neural network's structure, functional diagram, and mathematical DC motor model (Naung, Anatolii and Lin, 2019) and contrasted the outcomes of PI controllers and neural networks.

The following is how the paper is structured; Section II describes the suggested technique. The experimental analysis is covered in Section III, and the recommended work is wrapped up in Section IV.

Proposed Digital Twin

The proposed digital twin is given in Figure 1. The NARX model is initially trained using PRBS signal. It predicts the output. The weights of the NARX model is actively changed using the PSO algorithm to model the changes in the system. The model generates the output based on current dynamics. The Kalman filter estimates the state of the PMDC motor from the noisy measurement. The prediction from the NARX model is applied to the MPC controller, which controls the speed of the PMDC motor.

Table 1: PMDC motor parameters

Parameter	*Value*
Coil resistance	0.157 Ω
Coil inductance	0.00031 H
Back- emf constant	0.049 volt/(rad/sec)
Torque constant	0.05Nm/Amps
Rotational inertia	0.00046 kg-m*m
Friction coefficient	0.00027 Nm/(rad/sec)

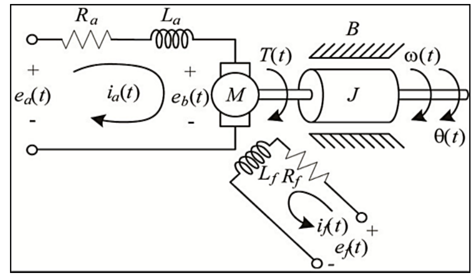

Figure 2: Electrical and mechanical equivalent circuit of PMDC motor

PMDC Motor

A PMDC motor can be represented in terms of its electrical dynamics and mechanical dynamics of the system. The PMDC motor's electrical and mechanical equivalent circuit is given in Figure 2.

The mathematical model of electrical subsystem obtained by applying Kirchoff's voltage law is,

$$v = iR + L\frac{di}{dt} + e_b \tag{1}$$

Where,is the excitation voltage applied to the PMDC motor, i is the current flowing through the motor, R is resistance of the armature winding, L is the inductance of armature winding and, e_b is the back electromotive force,

The mathematical model of mechanical subsystem obtained by applying force balance equation is,

$$T = J_m\frac{d^2\theta}{dt^2} + b_m\frac{d\theta}{dt} + T_L \tag{2}$$

Where, T is the generated torque, J_m is the motor inertia, b_m is the frictional constant, T_L is the load torque, θ is rotor angle.

Due to the presence of a feedback circuit inside the system, PMDC is a self-regulatory system. The system always settles at a new equilibrium point till the operating range is in its rated capacity. The manipulated variable is input voltage, torque being the disturbance and, the output is speed in rad/sec. The parameters of the PMDC motor considered in this work is given in Table 1.

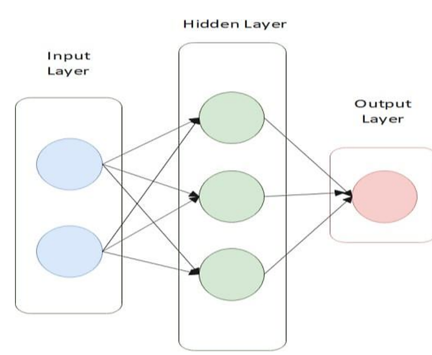

Figure 3: Basic structure of NARX using ANN

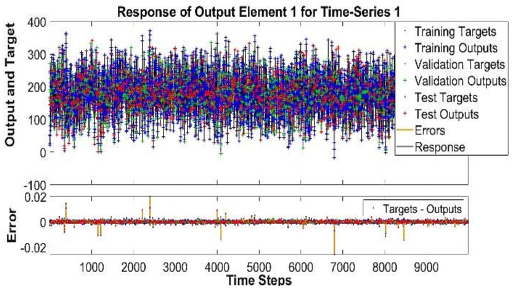

Figure 4: Response of NARX model for PRBS input

The rating of the PMDC is 24 V, 3 A, 4600 RPM. On substituting the parameters given in Table 1, in (1) and (2) gives,

$$\mathrm{I(s)} = \frac{1}{0.0003137\mathrm{s} + 0.157378}(\mathrm{V(s)} - 0.049) \tag{3}$$

$$\omega(\mathrm{s}) = \frac{1}{0.000466\mathrm{s} + 0.0002546}(0.0501\mathrm{I}) \tag{4}$$

The PMDC motor's characteristics in S-domain is considered as,

$$\frac{\omega(\mathrm{s})}{\mathrm{v(s)}} = \frac{0.0501}{1.4618 * 10^{-7}\mathrm{s}^2 + 7.3418 * 10^{-5}\mathrm{s} + 0.0025} \tag{5}$$

Based on the pole location it is seen that the system is stable.The response of the system is overdamped. The overdamped response is characterized by,

- Damping ratio ε is 1.918336146.
- Natural frequency ω_n is 130.9061006 rad/Sec.
- Settling time is 0.105 sec for 2% error and 0.121 sec for 1% error.
- Steady-state gain of the system is 20.

PID controller is chosen initially to control the speed of the PMDC and it is tuned using reaction curve method. The PID controller's parameters are: derivative gain (kd), proportional gain (Kp) 34032, and integral gain (Ki) 0.0025.

Nonlinear Auto Regressive Exogenous (Narx) Model

NARX model has the information about the nonlinearity in the system. For digital twin, the model needs to learn from the system being controlled (Table 2). The system in the real time exhibits many types of nonlinearities. Since the general model of the systems is often a localised model, it cannot establish the relationship between the system's input and output when the operating point changes. The nonlinearities that can occur at the time of operation of the systems are saturation of the actuators, minor degradation of the system components. This type of nonlinearities needs to be addressed. Thus, it is better to go with the model that is capable of adapting to the nonlinearities (Naung, Anatolii and Lin, 2019). The equation of the NARX model is,

y(t) = f(y(t - 1) + y(t - 2) + ... y(t - d) + ... u(t - d)   (6)

Where f (.) is the nonlinear function that represents the system's nonlinearities, y (.) is the expected output of the system, u (.) is the input that was applied to the system, and d is the amount of time that has passed before prior values are taken into account.

The dynamics of the PMDC motor can be represented by the selection of NARX model parameters. Artificial neural networks (ANN) can be employed to implement the NARX. The quantity of layers, neurons, and activation functions for each layer define an ANN. The model can also be made regressive by including delay components. The Figure. 3 describe the basic structure of NARX using ANN.

In order to build NARX, an ANN with three layers an input layer with six neurons, a hidden layer with ten neurons, and an output layer with one neuron was used. Transig is the selected activation function for the hidden layer. The output layer uses linear activation function. The data are scaled and normalized before given to the network for training. The inputs are voltage and load torque at current instance and immediate previous instance. Also, current and previous instance output are feedback as input.

Table 2: Performance measure of NARX

Label	*Target Values*	*MSE*
Training	2801	0.0129
Validation	600	0.0006
Testing	600	0.0004

Response of the NARX Model

The PMDC motor is excited by the PRBS signal. The input and output data set are collected and neural network is trained using the NN app. The model's MSE is in the range of 10^{-5}. Figure. 4 displays the NARX model's behaviour during training, testing, and validation.

Kalman Filter

It is the optimal linear filter. The RLS estimation is purely data oriented it does not have the knowledge about the

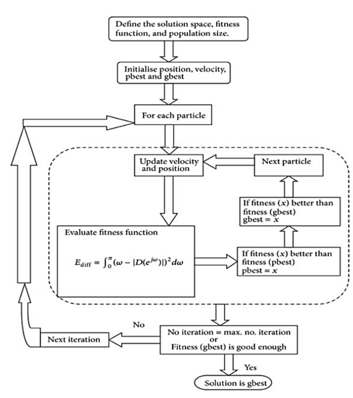

Figure 5: Flowchart of PSO algorithm

dynamics system. Kalman filter fuses the RLS and state space equation to form the robust estimator. The noise affected to the system should be of Gaussian in nature. The estimation is split into two pieces by the Kalman filter: the time update equation and the measurement update equation. The state space equation of the system serves as the basis for the time update equation. Priori and posteriori are terms used to describe the time update equation and measurement update equation, respectively (Arunshankar, 2002). The time update equation is given by,

$$\tilde{x}_{i+1|i} = A * \hat{x}_{i|i} + B * U_i$$

The measurement update equation is given by

$$\hat{x}_{i+1|i+1} = \tilde{x}_{i+1|i} + K\left(y_{i+1} - C * \tilde{x}_{i+1|i}\right)$$

Where, x is state variable of the system, y is measurement from the system, U is input given to the system, A, B, C, D are the state space matrices of the system, K is Kalman gain, i is the time instants.

The term $\left(y_{i+1} - C * \tilde{x}_{i+1|i}\right)$ is called residual or innovation term. This helps the Kalman filter to update its prediction or fine tune its prediction. Q represents the process noise covariance matrix, it holds the variance of each state noise. P is the error covariance which is set to high value if the knowledge about the initial state is poor. The values of Q, R and P used in this work are,

Q=[1 0 0 0 1 0 0 0 1] (9)

R=0.001 (10)

P=[100 0 0 0 100 0 0 0 100] (11)

Particle Swarm Optimization

It is a metaheuristic algorithm. The animals that live in group have some level of intelligence which is very much suitable for searching the solution in multidimension space (Qi, Shi and Zhang, 2019). In PSO algorithm, large number of particles called swarms are created and initialized with random position in the search space. The algorithm's fundamental goal is to enable the particle to move around, communicate with society each particle's local best, and work toward being the best possible globally. In this work the PSO is used minimize the MSE of NARX.

The network has 70 weights that need to be modified to make the model adaptable. So, it is a 70-dimensional search space. Every instant PSO do the search for the 70 weights which makes the cost function minimization. Figure 5 illustrates the flowchart of PSO algorithm.

The weights which transfers the input to input of hidden layer is,

$W^1 = [\, W_{11}\, W_{12}\, W_{13}\, W_{14}\, W_{15}\, W_{16}\, W_{17}\, W_{18}\, W_{19}\, W_{110}\, W_{21}\, W_{22}\, W_{23}\, W_{24}\, W_{25}\, W_{26}\, W_{27}\, W_{28}\, W_{29}\, W_{210}\, W_{31}\, W_{32}\, W_{33}\, W_{34}\, W_{35}\, W_{36}\, W_{37}\, W_{38}\, W_{39}\, W_{310}\, W_{41}\, W_{42}\, W_{43}\, W_{44}\, W_{45}\, W_{46}\, W_{47}\, W_{48}\, W_{49}\, W_{410}\, W_{51}\, W_{52}\, W_{53}\, W_{54}\, W_{55}\, W_{56}\, W_{57}\, W_{58}\, W_{59}\, W_{510}\, W_{61}\, W_{62}\, W_{63}\, W_{64}\, W_{65}\, W_{66}\, W_{67}\, W_{68}\, W_{69}\, W_{610}\,]$

INPUT LAYER HIDDEN LAYER OUTPUT LAYER (12)

$$Y^1 = [Y_{11}{}^1\, Y_{21}{}^1\, Y_{31}{}^1\, Y_{41}{}^1\, Y_{51}{}^1\, Y_{61}{}^1\, Y_{71}{}^1\, Y_{81}{}^1\, Y_{91}{}^1\, Y_{101}{}^1\,] = (W^{1^{-1}} * [X_1\, X_2\, X_3\, X_4\, X_5\, X_6\,]) \qquad (13)$$

Where, W^1 represents the weights of layer 1, X is input to the network, Y^1 is aggregated output of each neuron in the hidden layer.

The aggregated output of each neuron passing through the activation function.

$$Y^2 = W^{2^{-1}} \times z^1 \qquad (14)$$

$$Y = purelin(Y^2) \qquad (15)$$

Where, z^1 is the output from the activation function of hidden layer neurons,are the weights between hidden layer and neuron,is aggregated output of the input from the hidden layer and output, and Y is output of neural network.

The MSE equation that PSO optimizes is,

$$Error = +(Y(i) - y(i)^{estimate})^2 \qquad (16)$$

Where, i represents current instant, $y(i)^{estimate}$ is the estimate from Kalman filter.

Based on the global best, personal best and its current position the particles move. The new velocity and the particle's new position is given by (17) and (18).

$$v_i(t+1) = W * v_i(t) + C_1(P_i(t) - x_i(t)) + C_2(g(t) - X_i(t)) \qquad (17)$$

$$X_i(t+1) = X_i(t) + v_i(t+1) \qquad (18)$$

Let the position of the particle i be , the velocity of the particledenoted by,be the personal best of the particle I andbe the global best.

Model Predictive Controller

Model predictive controller is chosen as the controller for this work. Because of the constraint handling capacity and good adaptive nature, MPC is chosen. MPC employs a model to anticipate potential responses when a control step is required. The control is then found by maximising the cost function in the prediction window. The NARX model adapts the change in the state of the system. The MPC uses the NARX model to estimate the state (Hertneck et al., 2018) (Figure 6).

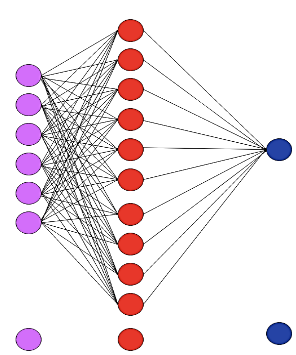

Figure 6: NARX snetwork structure for PMDC

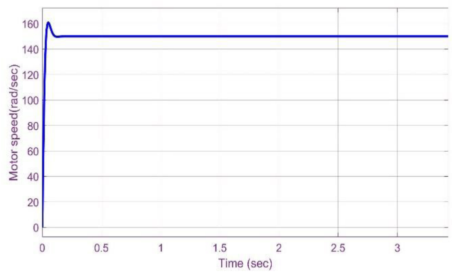

Figure 7: Servo response of PMDC

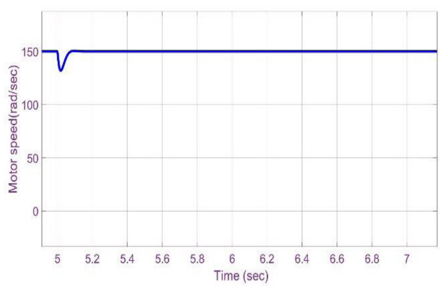

Figure 8: Regulatory response of PMDC

Optimization of the cost function is given by

$$J = (R_s - Y)^T (R_s - Y) + \Delta U^T \underline{R} \Delta U \tag{19}$$

Where, R_s is vector containing information about reference input at each instant, size is same as that of prediction window, Y is the plant output, ΔU is the step size of the input applied to the plant.

The first term in the cost function is the error due to deviation of the system output from the set point and, the second term incorporates is the step size of the input, with $\underline{R}$ the tuning parameters for controlling the step size is $\underline{R}$.

The control trajectory is given by,

$$\Delta u(k_i), \Delta u(k_i + 1), \ldots, \Delta u(k_i + N_c - 1) \tag{20}$$

Here, k_i represents the sampling instant, N_c control horizon.

To optimal control signal is given by,

$$\Delta U = \left(\emptyset^T \emptyset + \underline{R}\right)^T \emptyset^T \left(R_s - F * x(k_i)\right) \tag{21}$$

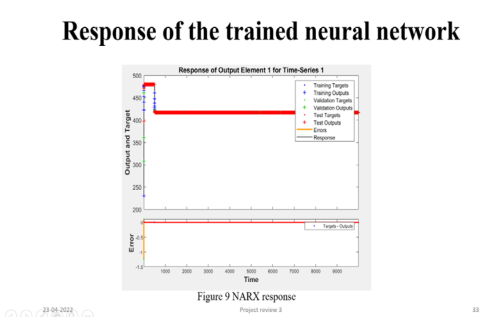

Figure 9: Output from NARX model

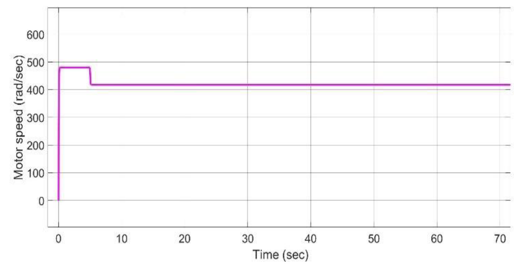

Figure 10: True response of PMDC

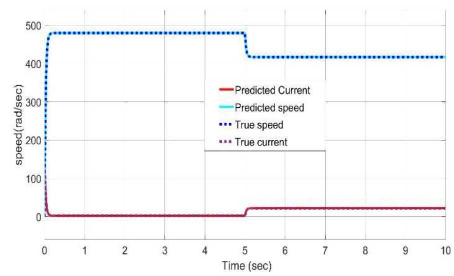

Figure 11: Estimation of current and speed by Kalman filter

Result And Discussion

Servo Response of PMDC with PID

The static and dynamic analysis of set point tracking is presented in this section. The step change in set point applied is 150 rad/sec. From Figure 7, Peak overshoot is 6.67%, rise time is 0.012 sec and settling time is 0.043 sec.

Regulatory Response of PMDC with PID

The static and dynamic analysis of disturbance rejection is done on this section. A disturbance of 4Nm is applied at time $t \geq 5$ sec. From Figure 8, undershoot is 6.67% and recovery time is 0.0293 sec.

Response of the NARX Model of PMDC

The system is simulated with an input of 24 volt and after 5 sec a torque of 1 Nm is applied. By comparing the response shown in Figure 9 and 10, it is evident that NARX model has approximated the PMDC motor.

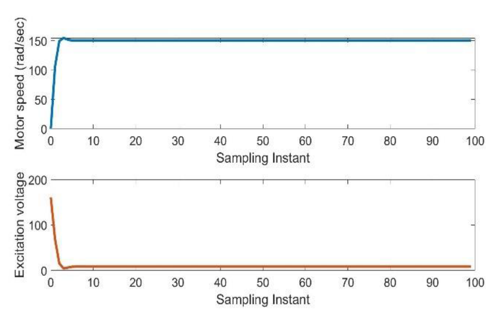

Figure 12: Response of system with MPC controller for case 1

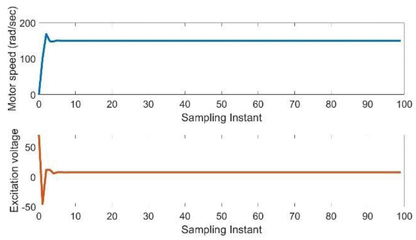

Figure 13: Response of system with MPC controller for case 2

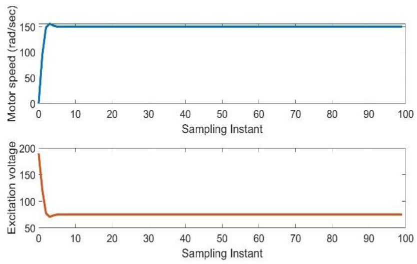

Figure 14: Response of system with MPC controller for case 3

Estimation by Kalman Filter

Figure 11 shows the estimated output obtained from the Kalman filter, with the input of 24 volts being applied to the system at t = 5 sec and, a disturbance torque of 1-Nm.

Convergence of PSO Algorithm for Different Cases

The PSO algorithm is tested for three cases,

- Case 1: The resistance value of armature winding is increased by 10 times.
- Case 2: The inductance value of armature winding is increased by 10 times.
- Case 3: Both resistance and the inductance value of armature winding are increased by 10 times.
- Case 1: MSE =7.7994e-15.
- Case 2: MSE =2.3617e-13.
- Case 3: MSE = 9.7e-10.

Response of System with Mpc Controller

The MPC controller uses the information from the NARX model to predict the response of the updated system. For each case the NARX model is updated by the PSO algorithm. The MPC has performed well with the help of an updated model. The performance of MPC in all three cases were shown as Figures 12-14, respectively.

The set point is fixed at 150 rad/sec. In all three cases, the MPC is capable of controlling the system at the desired setpoint.

Conclusion

The concept of digital twin is implemented for the PMDC motor. The Nonlinear Auto Regressive Exogenous model for the PMDC motor is developed. The model is updated by using PSO algorithm. This way the learning part is completed, the flow of information from the plant to the model. The second part is the flow of information from model to plant. Based on the updated model's prediction, the MPC controller controls the plants. The variation of the system is done for three cases. In all three cases the convergence of the PSO algorithm for the cost function as MSE is checked. The resistance variation and inductance variation of the plant is simulated. It is seen that the step size of the controller output is fixed. In case 3 the voltage given to the PMDC motor goes out of the specified rating.

References

Arunshankar, J. (2002). Estimation of launch and impact points of a flight trajectory using kalman filter/smoother.

Brylina, O. G., Kuzmina, N. N., & Osintsev, K. V. (2020, November). Modeling as the foundation of digital twins. In 2020 Global Smart Industry Conference (GloSIC) (pp. 276-280). IEEE.

Duan, H., & Tian, F. (2020). The development of standardized models of digital twin. IFAC-PapersOnLine, 53(5), 726-731.

Hänel, A., Schnellhardt, T., Wenkler, E., Nestler, A., Brosius, A., Corinth, C., ... & Ihlenfeldt, S. (2020). The development of a digital twin for machining processes for the application in aerospace industry. Procedia CIRP, 93, 1399-1404.

Hertneck, M., Köhler, J., Trimpe, S., & Allgöwer, F. (2018). Learning an approximate model predictive controller with guarantees. IEEE Control Systems Letters, 2(3), 543-548.

Kritzinger, W., Karner, M., Traar, G., Henjes, J., & Sihn, W. (2018). Digital Twin in manufacturing: A categorical literature review and classification. Ifac-PapersOnline, 51(11), 1016-1022..

Lugaresi, G., & Matta, A. (2021). Automated manufacturing system discovery and digital twin generation. Journal of Manufacturing Systems, 59, 51-66.

Naung, Y., Anatolii, S., & Lin, Y. H. (2019, January). Speed control of DC motor by using neural network parameter tuner for PI-controller. In 2019 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus) (pp. 2152-2156). IEEE..

Qi, Z., Shi, Q., & Zhang, H. (2019). Tuning of digital PID controllers using particle swarm optimization algorithm for a CAN-based DC motor subject to stochastic delays. IEEE Transactions on Industrial Electronics, 67(7), 5637-5646.

Semeraro, C., Lezoche, M., Panetto, H. and Dassisti M. (2021). Digital twin paradigm: A systematic literature review. Comput. Ind. 130.