Designing MPC algorithms for velocity control of brushed DC motor and verification with SIL tests

Murat Sahin

Published online: 02 Feb 2023.

Submit your article to this journal ⏎

Article views: 1698

View related articles ⏎

View Crossmark data ⏎

Taylor & Francis
Taylor & Francis Group

REGULAR PAPER

OPEN ACCESS | Check for updates

Designing MPC algorithms for velocity control of brushed DC motor and verification with SIL tests

Murat Sahin

Control Systems Department, Roketsan Inc., Ankara, Turkey

ABSTRACT
The model predictive control (MPC) tries to find the best control output by optimizing among the predictions. In this context, determining the objective function of optimization is a critical process. This study discusses the velocity control of the mini brushed direct current (BDC) motor used in the actuator with MPC. Four different functions have been defined to design the most appropriate cost function for MPC by considering the parameters that may be effective in the motor dynamics such as velocity, current, power, and switching states. Software in the Loop (SIL), one of the modeling-based testing methods, was used within the scope of testing the algorithms. After the SIL tests, it has been observed that the mini BDC motor can be successfully controlled with simple adjustments to the cost function of the MPC. As a result of the analysis obtained the best results with the objective function consisting of velocity error, estimated current, and the difference between the two estimated velocity values. With this controller, the BDC motor can be controlled without overshoot and with a steady-state error of less than 2% under load.

1. Introduction

Brushed DC (BDC) motors have been widely used in various fields, such as pumps, air fans, automotive window mechanisms, and windshield wipers. With the development of technology, it has found intensive use in high-technology areas such as robotics, drones, and missile actuators [1,2]. An essential step in controlling the BDC motor is commutation. The commutation process in BDC motors is carried out by the contact of a brush with a sliding mechanical surface. This mechanical method is reliable and able to withstand high voltage and current. In addition to these advantages, sparks or arcs may occur in the mechanical commutator and thus create a risky situation [3].

Because of these disadvantages, controller design is another critical step for controlling the BDC motor. In the 2000s, MPC became preferred controlling of electrical machines, including high-speed BDC motors. Especially in the last ten years, the usage level has increased significantly [4]. When controlling the BDC motor with a classical controller like PID, the pulse width modulation (PWM) method is used to reduce torque ripples and achieve a more efficient drive [5]. At each step in controlling with MPC, the controller determines the voltage or switching selection that gives to get the best value for the cost function. This will result in smoother switching for inverters and, therefore, fewer torque ripples [6,7].

When designing MPC, one critical step is defining an appropriate cost function. Generally, there are

two parts to the cost function; the difference between the reference and the output and the controller signal difference between the two steps. While the error is minimized with the first parameter, smoother control performance is tried to be obtained with the second one [8]. The speed error and the values of the current and future control signals were used in the cost function to perform speed control of the DC motor [9]. And also, the weighting factor was added to the inputs in the studies on the transfer function of the motor, and the controller performance was examined according to the change of these coefficients [10]. Weight factors need to be adjusted by the system. One of the methods of adjusting these factors is the deterministic method. In deterministic methods, the value of the factors is increased or decreased with a constant value from a specific starting point [11]. Another advantage of MPC is that it can be easily implemented into microprocessors because of its simple code structure. Using a floating number during the optimization process in MPC provides excellent convenience to the user. However, this may necessitate the use of costly equipment. To solve this problem, studies have been carried out on MPC, which will work at fixed point in DC motor control and can be implemented into a 32-bit microcontroller [12,13].

In this study, MPC was designed within the scope of velocity control of a mini BDC motor used in actuators. In the design, especially the cost function was emphasized. By examining different studies in the literature,

four different cost functions were prepared and analyzed within the scope of BDC motor control. Analysis studies were carried out in the SIL environment, which is a part of the model-based design method. SIL tests enable software components running on the existing system to be tested in a simulation environment [14,15]. It provides high flexibility as it does not require hardware, and test parameters can be changed easily. This way, different scenarios and control algorithms can be tested very quickly in control applications [16]. One of the other motivations of the study is that the designed MPC can run on a low-level microcontroller. It is an essential parameter realizing cost-effective designs. For this reason, sampling times in which the algorithm can work on microcontrollers were preferred in analysis studies. Firstly, studies about BDC motor control and inverters were examined in detail. In Section 3, four different cost functions were designed for MPC. In addition, an h-bridge inverter circuit was established to make the model more realistic. The SIL environment was prepared for testing algorithms. The MATLAB Coder application has transformed each MPC algorithm into C codes that can run on real hardware. The results and discussion of SIL tests are presented in last part.

2. BDC motor equations and H-Bridge circuit

BDC motor equations are based on armature current and angular velocity variables [17].

$$\frac{di_a}{dt} = \frac{1}{L_a}(V_a - R_a i_a - V_{emf}) \tag{1}$$

R_a indicates armature resistance, L_a is armature inductance, i_a and V_a are armature current and voltage, and V_{emf} is the back emf voltage. The back emf voltage depends on the motor velocity (ω_m) and the back emf constant (k_b) [17]:

$$V_{emf} = k_b \omega_m \tag{2}$$

The electric torque (T_e) generated by the motor depends on the armature current and torque constant (k_t) and is found in the following equation [17].

$$T_e = k_t i_a \tag{3}$$

The velocity of the motor (ω_m) depends on the resistance of the electrical torque against external and internal factors. In the velocity equation given below, T_l represents external load, B friction coefficient, and J_m motor inertia [17].

$$\frac{d\omega_m}{dt} = \frac{1}{J_m}(T_e - T_l - B\omega_m) \tag{4}$$

The parameters of the motor are given in the Appendix. BDC motors are generally driven by H-Bridges. H-Bridge is an electronic circuit that allows voltage to
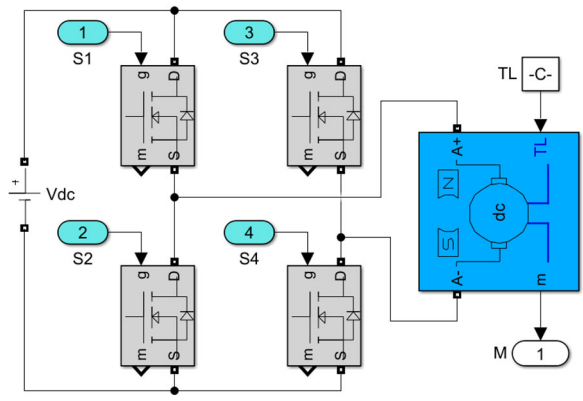

Figure 1. H-Bridge circuit and BDC motor.

be applied to a BDC motor in both directions. A primary H-Bridge circuit has four switching components (Transistor, Mosfet, etc.). Usually, there are 16 different switching possibilities if there are four switches. But due to the nature of the H-Bridge circuit, short circuits occur in 7 of these possibilities. For example, If S1 and S2 are open at the same time, it is seen that there will be a short circuit, and this situation will damage the circuit. This condition also occurs when S3 and S4 are open. In 5 switching combinations, the current will not flow from the circuit and turn OFF. For example, the state that all switches are OFF or only one of the switches is ON. Setting the S1 and S4 switches ON will cause the motor to turn in the forward direction while setting S2 and S3 ON will cause the motor to turn in the reverse direction. If only S1 and S3 are ON or only S2 and S4 are ON, the motor terminals will be short-circuited, causing a brake [18]. The H-Bridge circuit and BDC motor driving method are shown in Figure 1.

3. MPC design

MPC is a robust optimization and control technique that makes predictions about the system. The optimal control signal is generated by predicting future responses of the system, minimizing the cost function. The optimization process is set up to minimize the error between the predicted and the reference value, and MPC controls the variables that will make this minimization [19]. The cost function evaluation for each sampling period is determined according to the forecast horizon (N) selected depending on the system type and performance requirement. A cost function that includes future situations, references, and future actions can be defined as follows [20]:

$$J = f(x(k), u(k), \dots, u(k + N - 1)) \tag{5}$$

A more detailed version of the equation given in (5) is shown below. In the equation, x_{ref} represents the reference value, $x(i + 1)$ represents the predicted system response, and $u(i)$ and $u(i - 1)$ represent the control variables. λ is the weighting factor greater than 0, which

can be used for adjustment. For $u(i)$, variables such as voltage, current, and switching states can be used [21].

$$J = \sum_{i=k}^{k+N-1} \left((x(i+1) - x_{ref})^2 + \lambda(u(i) - u(i-1))^2\right) \tag{6}$$

MPC for BDC motor control with H-Bridge can be designed using the following three steps [22]:

1) Equations describing the relationship between all possible switching states and generated voltages are prepared [22]:

S1 and S3 switches must work opposite to each other. (S2 and S4 are the same.) Therefore, to simplify the process, only two switches were used instead of four. S1 and S2 are controlled by the Sa, and S3 and S4 are controlled by the Sb. The voltage values obtained by three switchings on the H-Bridge are shown below.

Sa	Sb	V$_a$
1	0	V$_{dc}$
0	1	−V$_{dc}$
0	0	0

2) Discrete-time equations are prepared to predict the future behaviour of the variables to be controlled [22]:

Various discretization methods can be used to obtain a discrete-time model for calculating predictions. One of the simplest methods is the Forward Euler, which is based on derivatives. This method brings the prediction expression by leaving the term $x(k+1)$ alone. T_s is the sampling time [23].

$$\frac{dx}{dt} \approx \frac{x(k+1) - x(k)}{T_s} \tag{7}$$

The motor velocity depends on the current, so it is necessary to predict the current first. The following equation is obtained when the Euler method given in (7) is applied to equation (1).

$$i_a(k+1) = \left(1 - \frac{R_a}{L_a}T_s\right)i_a(k) + \frac{V_a}{L_a}T_s - \frac{V_{emf}}{L_a}T_s \tag{8}$$

Similarly, the following equation is obtained when Euler's method is applied to the velocity equation given in (4).

$$\omega_m(k+1) = \left(1 - \frac{B}{J_m}T_s\right)\omega_m(k)$$
$$+ \frac{k_t}{J_m}T_s i_a(k+1) - \frac{T_l}{J_m}T_s \tag{9}$$

3) A cost function representing the system's behaviour is determined using the equations obtained in step 2 and adding other variables or constraints if necessary [22].

The first member of the cost function is the difference between the reference value and the system response. This value will be the velocity response with the velocity reference in this application. A cost function consisting of only this member can also be used, but as seen from previous studies, more than such a function is needed to control all the systems' dynamics. In this section, four different cost functions are prepared to analyze the effects of different parameters within the cost function. The coefficients of the parameters of each MPC consisting of these cost functions were optimized separately and each was tested in the SIL environment.

It is necessary to limit the current to get a smoother response and prevent overshoot. The predicted current value and the velocity error were used in the first function. This function select the switching that gives the lowest velocity error and the lowest current value at each step.

$$J_{cur} = \lambda_1 * (\omega_m(k+1) - \omega_{mref})^2 + \lambda_2 * i_a(k+1)^2 \tag{10}$$

In motor control applications, inverter switching plays an important role. For energy efficiency, the number of switching may need to be reduced or controlled. In this context, the difference between the previous switch state and the new switch state can be added to the cost function [24]. In equation (11), the predicted velocity with $\omega_m(k+1)$, and the reference velocity by ω_{mref}. Possible switch states are indicated by $S_a(k+1)$ and $S_b(k+1)$, and previous switch states by $S_a(k)$ and $S_b(k)$. The predicted current value and the other objective functions were also used here to avoid the high current.

$$J_{sw} = \lambda_1 * (\omega_m(k+1) - \omega_{mref})^2 + \lambda_2 * i_a(k+1)^2$$
$$+ \lambda_3 * (|S_a(k+1) - S_a(k)| + |S_b(k+1)$$
$$- S_b(k)|)^2 \tag{11}$$

In real control applications, obtaining a smooth system response without overshooting is critical for system safety. A more controlled reference tracking is tried to avoid applying significant instant signals to the system. In this context, the difference in velocity predicted in two consecutive steps can be added to the cost function. In this case, the minimization process will tend to alternatives with less difference in velocity predictions.

$$J_{vel} = \lambda_1 * (\omega_m(k+1) - \omega_{mref})^2 + \lambda_2 * i_a(k+1)^2$$
$$+ \lambda_4 * (\omega_m(k+1) - \omega_m(k))^2 \tag{12}$$

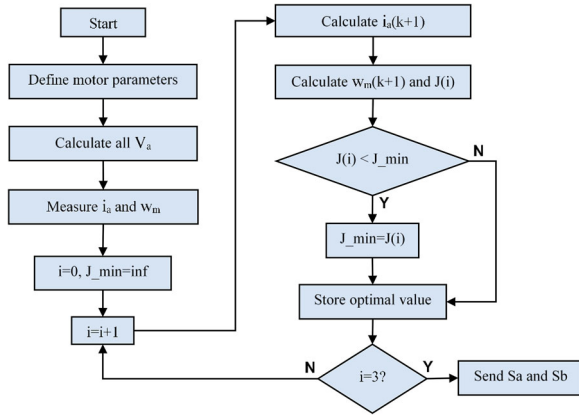Another critical factor is the voltage applied in the switching state and the current produced by the motor.

Figure 2. Flowchart of MPC algorithm.

Figure 3. Structure of controller.

Figure 4. The SIL block diagram.

The combination of these two parameters will give input power equation of the motor. Therefore, this factor can be added to the J_{pow} cost function as in (13) [25]. With V_a, the possible voltage value is shown.

$$J_{pow} = \lambda_1 * (\omega_m(k+1) - \omega_{mref})^2 + \lambda_2 * i_a(k+1)^2$$
$$+ \lambda_5 * (V_a * i_a(k+1))^2 \qquad (13)$$

One of the powerful features of MPC is that the cost function can define the constraints required for the system. The combination that gives the best result among the switching alternatives may also cause a high current from the BDC motor. Therefore, the current constraint given in (14) has been added to all cost functions [24]. When the current values are higher than a limit value, a significant value is assigned to the appropriate switching option to be excluded from the possibilities. (With i_{amax}, the maximum current value that can be used for the system is shown. In the part indicated by ∞ the equation, 1e10 is used in the application.)

$$\hat{f}(i_a(k+1)) = \begin{cases} \infty \ if \ |i_a(k+1)| > i_{amax} \\ 0 \ if \ |i_a(k+1)| \le i_{amax} \end{cases} \qquad (14)$$

The MPC algorithm first starts with defining the BDC motor parameters. In the next step, switching combinations and corresponding voltage values are determined. The values of $\omega_m(k+1)$ and $i_a(k+1)$ are then calculated for all possible switching cases. While calculating these values, current and velocity values measured through the system are used. To find the situation that gives the minimum cost at the end of the loop, a high value is initially assigned to J_min. This loop calculates cost functions ($J(i)$) for three switching situations. Therefore, a loop with three iterations (with $i = 3?$ expression in Flowchart was created.), and the cost functions defined above are calculated for each $\omega_m(k+1)$ and $i_a(k+1)$. At the end of the loop, the switching states giving the most minor cost are selected and transmitted from the algorithm to the inverter circuit. The flowchart of the algorithm is shown in Figure 2.
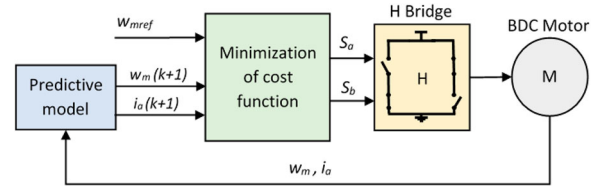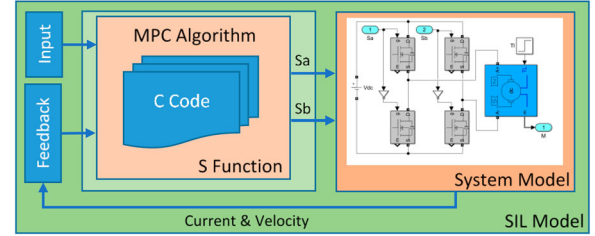
The general structure of the control system is given in Figure 3. The feedback from the DC motor is transferred to the prediction algorithm, and the minimization process is performed.

4. Software in the loop

In the SIL method, the control algorithms can work in a simulation environment on a computer, during simulation time, or in real time, depending on the needs. The SIL method enables the performance of the control algorithm to be seen through the system model [26]. This versatile method provides fast and simple verification of the various algorithms that will work within a system. In addition, it provides detailed comparison of different strategies developed to solve a specific task [14].

This study created a SIL simulation environment including, the MPC algorithm, H-Bridge circuit, and BDC motor models was created. Since MPC algorithms are the focus of the study, each MPC algorithm has been transformed into C codes that can work on real hardware with the MATLAB Coder application. Mosfet blocks were used for the H-bridge circuit, and DC machine block was used for the BDC motor. A solver working at fixed time is used to make the simulation close to the actual application. Considering the low-cost microprocessors, the sampling time was chosen as 10^{-5}. Detailed information about the simulation environment and SIL is given in the Appendix. The SIL block diagram is shown in Figure 4.

The code conversion process is shown in Figure 5. While performing the works in the fixed step, all actions of the model must be completed within the current sampling time. False control outputs will occur if the control algorithm flow cannot be completed within this period. The working times of the SIL tests in each step were controlled, and it was seen that they could run
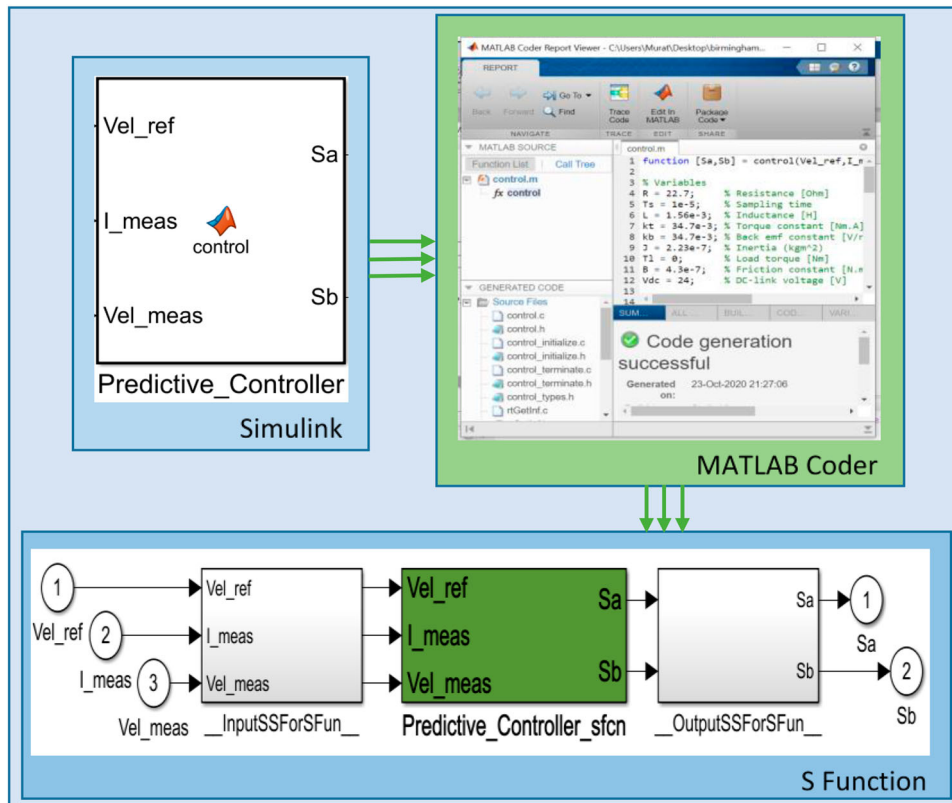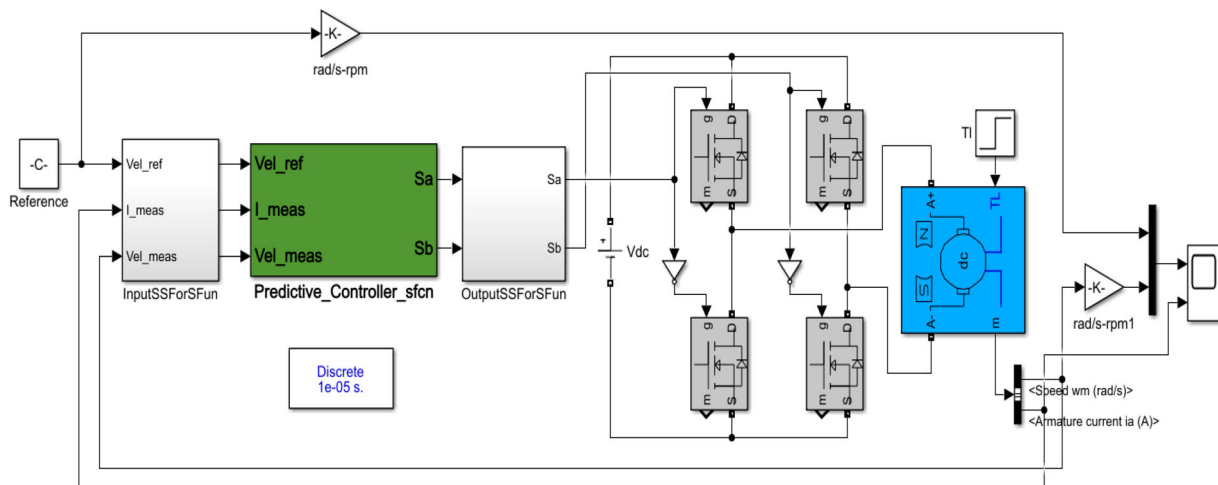
Figure 5. The code conversion process.

Figure 6. The SIL model.

within 10^{-5} sampling time. The data of these verifications are given in the Appendix.

4.1. SIL tests and results

The Simulink model prepared for SIL tests is given in Figure 6. The Predictive Controller block changes for each different cost function.

A 1000 rpm step reference was applied to test the algorithm. The duration of the test is 10ms, and a load of 10.6 mNm (this is the nominal torque of the motor.) is applied in the opposite direction at 5 ms. The results of the tests performed with MPC with the J_{cur} cost function are shown in Figure 7. The focus of this function is the motor current. MPC produces control signals that draw less current and consume less power as the λ_2 value increases. In large λ_2 values, smooth answers are obtained without overshooting. However, steady-state errors will increase. If the current results are observed, the motor reaches the 1A starting current at the beginning of the movement. Regarding the reference velocity, the current value is around 0. When the load is applied at 5 ms, it is seen that the current increases to around 0.3A. These values coincide with the nominal current value given in the motor parameters. The best values for this function were 1.5 for λ_1 and 10 for λ_2. There was a
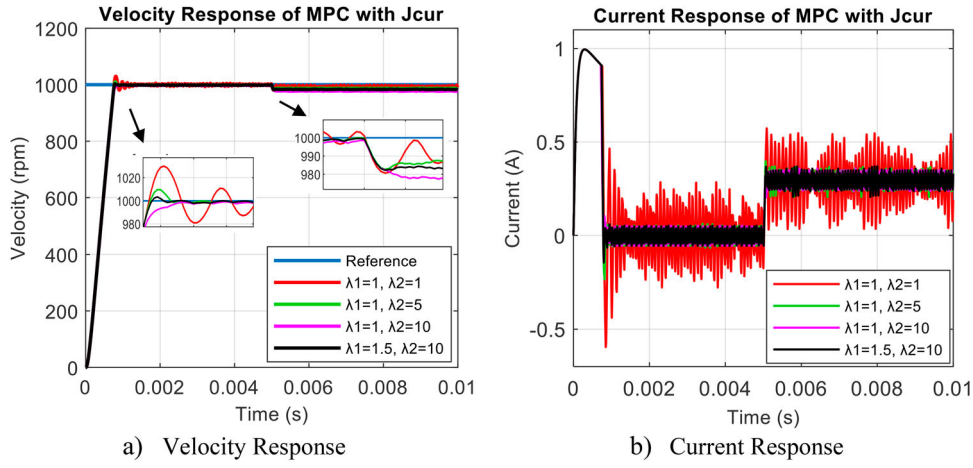
Figure 7. Velocity and current response of MPC with J_{cur} cost function.
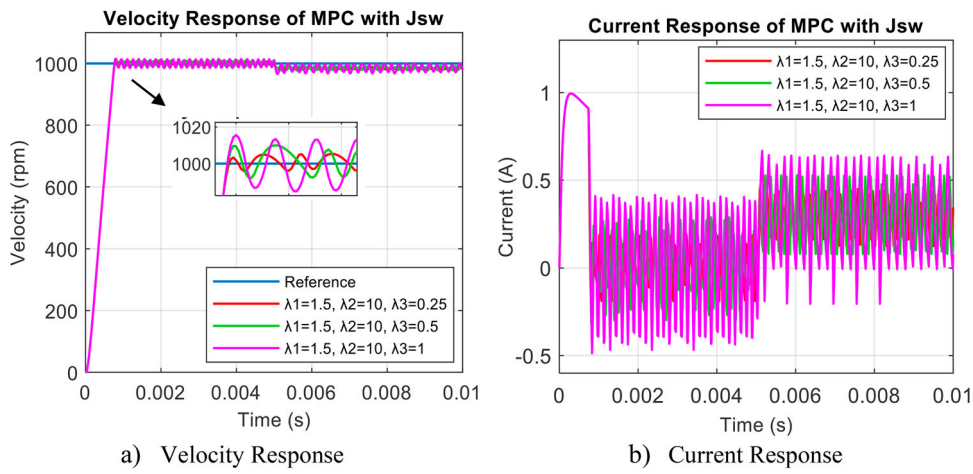

Figure 8. Velocity and current response of MPC with J_{sw} cost function.

slight overshoot in the system response, but under load, it gave a better response than where λ_1 had a value of 1.

The results of the tests performed with MPC with the J_{sw} cost function are shown in Figure 8. Oscillation was observed in velocity responses. As the λ_3 value increased, the oscillation amplitude increased. With the reset of the λ_3 value, the amplitude of the oscillation has reached the smallest value. As can be seen from Figure 8, oscillations increase as the switching effect increases in the objective function. The switching effect that was successful in the three-phase inverter [24] has not been successful here. Reducing the number of switching will also reduce the switching losses. However, in cases with errors in the system response, staying in the same switching may cause the error to be higher.

The results of the tests performed with MPC with J_{vel} cost function are shown in Figure 9. This function tries to minimize the differences between successively predicted velocity values. This is to generate smoother control signals. In this way, the overshoot seen in the tests with the J_{cur} is not seen here. As the λ_4 value decreases, the MPC produces closer control outputs to the reference. As a result of the tests, 0.25 was found suitable for λ_4.

The results of the tests performed with MPC with the J_{pow} cost function are shown in Figure 10. The tests were started by selecting the λ_5 value of 1. The focus of this function is power. Therefore, the MPC produces control signals that consume less power as the λ_5 value increases. In this case, more smooth answers can be obtained. However, steady-state errors occur. When the disturbing load is applied, the controller starts not to respond to the reference value.

With these results, it has seen that MPC with J_{cur} or J_{vel} functions given by equations (15) and (16) can be used successfully in BDC motor control.

$$J_{cur} = 1.5 * (\omega_m(k+1) - \omega_{mref})^2$$
$$+ 10 * i_a(k+1)^2 \tag{15}$$

$$J_{vel} = 1.5 * (\omega_m(k+1) - \omega_{mref})^2 + 10 * i_a(k+1)^2$$
$$+ 0.25 * (\omega_m(k+1) - \omega_m(k))^2 \tag{16}$$

As a final test, the performance of MPC against a sinusoidal reference signal was examined. For reference, 1000 rpm, 1 Hz sinusoidal wave, and as disturbing load, nominal torque between 0.25 and 0.75 s is applied. As in the step command, the controllers gave successful
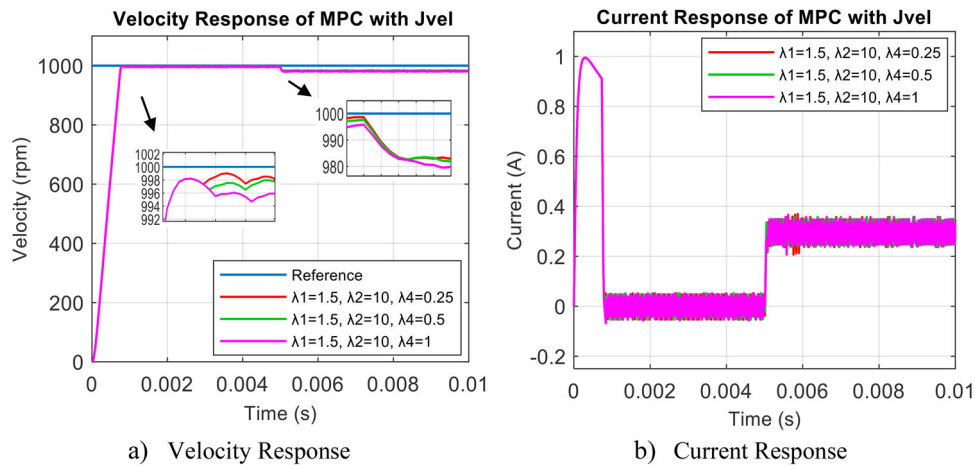
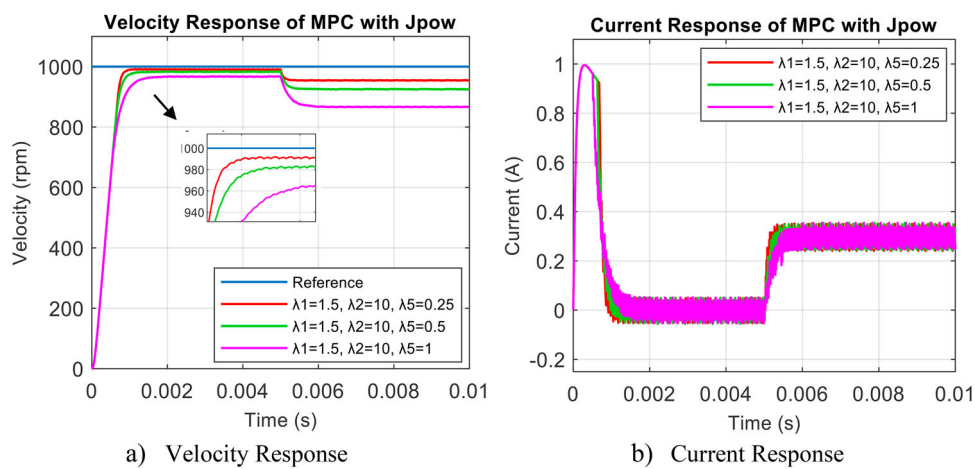**Figure 9.** Velocity and current response of MPC with Jvel cost function.

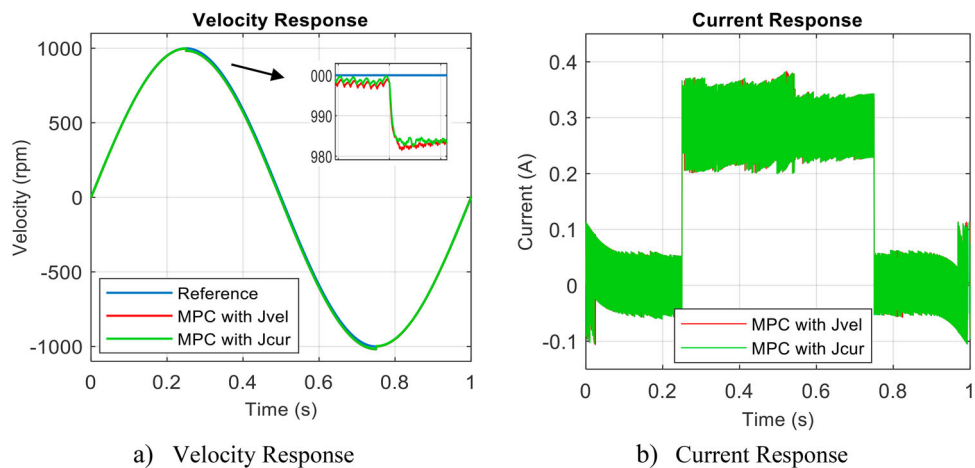**Figure 10.** Velocity and current response of MPC with J_{pow} cost function.

Figure 11. Velocity and current response for sinusoidal reference signal.

answers in the sinusoidal reference. While they followed the reference with 2-5 rpm differences in the no-load condition, they could continue with the differences of 20 rpm when the load was applied (Figure 11).

Figure 12 shows the velocity error values obtained in the tests. In the test where step reference is applied, it is seen that the controller responds quickly and

minimizes the error. It responds with a 2% error after the disrupting load is applied. In cases with no load, the error values are below 1%. Similarly, when a sinusoidal reference is applied, it quickly minimizes the error and continues to respond with an error of 2% when the load is applied. λ coefficients were determined very simply in this study. These values can be adjusted more precisely, and the error values can be further reduced

a) Step Reference

b) Sinus Reference
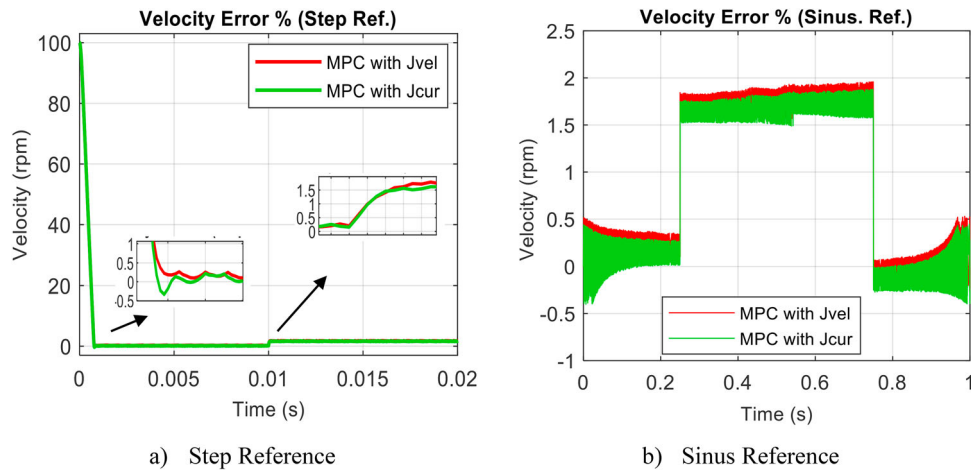
Figure 12. Velocity error % (Step Ref. and Sinus. Ref.)
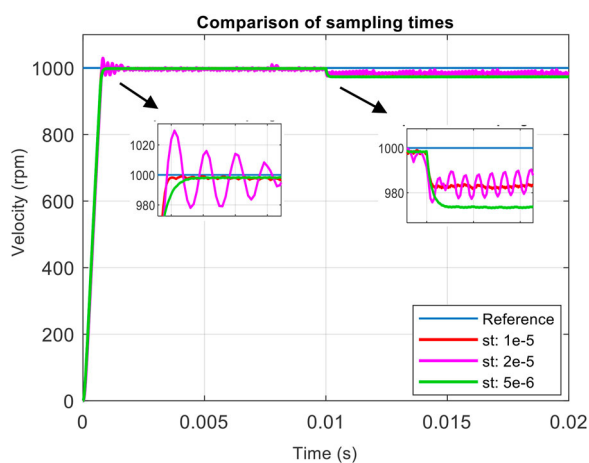
Figure 13. Comparison of sampling times.

according to the system to be applied and the controller requirements.

Tests at different values were performed to see the effect of sampling time on the system response. The results are shown in Figure 13. As the sampling time gets smaller, the oscillations in the response decrease. As the control signals are updated in smaller time steps, the velocity error is eliminated faster. But, the steady-state error increased in responses under load. Lambda coefficients are prepared for 1e-5 sampling time. For new values, readjustment is required. The selection of sampling time depends on the hardware to be used in the system. Within the scope of the study, the value of 1e-5 was determined by considering low-level processors.

5. Conclusion

In this paper, four MPCs with different cost functions were designed. Controllers have been studied with SIL tests and load applied to the BDC motor. With the first controller using the velocity error and the predicted current, the control was carried out with less than 2% steady state error and without overshoot. Oscillations

were observed in the velocity response in the tests performed with the second controller, which is formed by adding switching changes to the first controller. Switch exchange was more dominant within the cost function than velocity and current values. Therefore, the optimization left the velocity error in the background. In the tests performed with the third controller, which is formed by adding the predicted speed to the first controller, the control was performed without less than 2% permanent error and without overshoot. Unlike the first controller, the new parameter added flexibility to the controller. With this parameter, more precise adjustments can be made for applications where speed performance is adequate. In the fourth controller, predicted power is added to the first. Since the power parameter is more dominant in the tests, more steady-state errors occurred under load than the others.

SIL tests are the step before the hardware tests in model-based design, and the systems developed here can be quickly transferred to the hardware with minor updates. Therefore, the controllers designed in this study can work on real hardware with the necessary updates.

Acknowledgements

Disclosure statement

No potential conflict of interest was reported by the author.

ORCID

Murat Sahin http://orcid.org/0000-0002-3659-3528

References

[1] Batool A, Ain NU, Amin AA, et al. A comparative study of DC servo motor parameter estimation using various techniques. Automatika. 2022;63(2):303–312.

[2] Tripathi RP, Singh AK, Gangwar P, et al. Sensor-less speed control of DC motor using EKF estimator and TSK fuzzy logic controller. Automatika. 2022;63(2):338–348.

[3] Xiang X, Chai J, Sun X. A novel DC motor based on mechanical–electrical hybrid commutation. IEEE J Emerg Sel Top Power Electron. 2018;6(3):1605–1615.

[4] Schwenzer M, Ay M, Bergs T, et al. Review on model predictive control: An engineering perspective. Int J Adv Manuf Technol. 2021;117(5):1327–1349.

[5] Sainz, B, Leal, D, & Martinez, E. Novel Method to Ensure the Reliability of the DC Motors Controlled by an IGBT/PWM Converter. In 2021 IEEE IAS Pulp and Paper Industry Conference (PPIC); 2021, pp. 1–9.

[6] Trivedi MS, Keshr RK. Evaluation of predictive current control techniques for PM BLDC motor in stationary plane. IEEE Access. 2020;8:46217–46228.

[7] Taher, SM, Niasar, AH, & Taher, SA. (2021). A new MPC-based approach for torque ripple reduction in BLDC motor drive. 12th Power Electronics, Drive Systems, and Technologies Conference (PEDSTC), p. 1–6.

[8] Dani, S, Sonawane, D, Ingole, D, & Patil, S. Performance evaluation of PID, LQR and MPC for DC motor speed control. 2nd International Conference for Convergence in Technology (I2CT); 2017.

[9] Sahoo, S, Subudhi, B, & Panda, G. Optimal Speed Control of DC Motor using Linear Quadratic Regulator and Model Predictive Control. International Conference on Energy, Power and Environment: Towards Sustainable Growth (ICEPE); 2015.

[10] Alkurawy, LEJ, & Khamas, N. Model predictive control for DC motors. 1st International Scientific Conference of Engineering Sciences – 3rd Scientific Conference of Engineering Science (ISCES); 2018, p. 56–61.

[11] Carvalho, DF, Ganzaroli, CA, Dias, RN, Couto LA, & Calixto, WP. Optimization process applied to tuning of dynamic matrix control: study case with DC motor. IEEE 16th International Conference on Environment and Electrical Engineering (EEEIC); 2016.

[12] Boshkovski, G, Stojanovski, G, & Stankovski, M. Development of embedded model predictive controller. 13th IEEE International Conference on Control & Automation (ICCA); 2017, p. 76–81.

[13] Mishra I, Tripathi RN, Singh VK, et al. Step-by-Step Development and Implementation of FS-MPC for a FPGA-Based PMSM Drive System. MDPI Electronics. 2021;10(4):395.

[14] Motahhir S, Ghzizal A, Sebti S, et al. MIL and SIL and PIL tests for MPPT algorithm. Cogent Eng. 2017;4(1):1–18.

[15] Alonso-Eugenio V, Guerra V, Zazo S, et al. Software-in-loop simulation environment for electromagnetic underwater wireless sensor networks over STANAG 5066 protocol. Electronics. 2020;9(10):1611.

[16] Beyer C, Emmerich J, Werner U. Software in the loop – a window lifter model to guide students through the software development process. IEEE Global Engineering Education Conference (EDUCON); 2017, p. 1488–1493.

[17] Ekinci S, Hekimoğlu B, Izci D. Opposition based Henry gas solubility optimization as a novel algorithm for PID control of DC motor. Eng Sci Technol Int J. 2021;24(2):331–342.

[18] Özer T, Kıvrak S, Oğuz Y. H brıdge DC motor driver design and implementatıon with using dsPIC30f4011. Int J Innov Res Technol Sci Eng. 2017;6(10):75–83.

[19] Guler N, Biricik S, Bayhan S, et al. Model predictive control of DC-DC SEPIC converters with auto-tuning weighting factor. IEEE Trans Ind Electron. 2020:9433–9443.

[20] Irmak E, Güler N. A model predictive control-based hybrid MPPT method for boost converters. Int J Electron. 2020;107(1):1–16.

[21] Dragicevic T. Model predictive control of power converters for robust and fast operation of AC microgrids. IEEE Trans Power Electron. 2018;33(7):6304–6317.

[22] Shadmand MB, Balog RS, Abu-Rub H. Model predictive control of PV sources in a smart DC distribution system: maximum power point tracking and droop control. IEEE Trans Energy Convers. 2014;29:4.

[23] Zhang, Y, Gong, H, Li, J, Qiu, B, & Yang, M. Discrete time-varying four fundamental operations implemented by Euler forward difference. 36th Chinese Control Conference; 2017, p. 9807–9812.

[24] Rodriguez, J., & Cortes, P. (2012). Predictive control of power converters and electrical drives. John Wiley & Sons, Ltd., Publication, pp. 137-152.

[25] Vargas R, Rodriguez J, Rojas CA, et al. Predictive control of an induction machine fed by a matrix converter with increased efficiency and reduced common-mode voltage. IEEE Trans Energy Convers. 2014;29(2):473–485.

[26] Rahman AA, Mohamad NR. Software in The loop technique: an approach To support reconfiguration of manufacturing system. ARPN J Eng Appl Sci. 2016;11(16):9789–9795.

Appendix

Motor parameters:

R = 22.7 Resistance [Ohm]
Ts = 1e-5 Sampling time
L = 1.56e-3 Inductance [H]
kt = 34.7e-3 Torque constant [Nm.A]
kb = 34.7e-3 Back emf constant [V/rad/s]
J = 2.23e-7 Inertia (kgm^2)
B = 4.3e-7 Friction constant [N.m.s]
Vdc = 24 DC-link voltage [V]

Model configuration

Solver type: Fixed step
Solver: ode 3 (Bogacki-Shampine)
Fixed-step size: 1e-5 [s]
Working time of each step of SIL

Working time of each step of SIL