Enhanced pid vs model predictive control applied to bldc motor

M.S Gaya¹,a, Auwal Muhammaq²,b, Rabiu Aliyu Abdulkadir¹, S.N.S Salim³, I.S Madugu¹, Aminu Tijjani¹, Lukman Aminu Yusuf⁴, Ibrahim Dauda Umar², M.T.M Khairi⁵,c
¹Dept of Electrical Engineering, Kano University of Science & Technology, Wudil, Nigeria
²Dept of Physics, Kano University of Science & Technology, Wudil
³Dept of Industrial Automation & Robotic, Universiti Teknikal Malaysia, Melaka, Malaysia
⁴Dept of Electrical Engineering, Bayero University Kano, Nigeria
⁵Dept of Control & Mechatronics, Universiti Teknologi Malaysia, Skudai, Nigeria

a)muhdgayasani@gmail.com, b)auwalsafa@gmail.com, c)rabiukk@gmail.com

Abstract. BrushLess Direct Current (BLDC) motor is a multivariable and highly complex nonlinear system. Variation of internal parameter values with environment or reference signal increases the difficulty in controlling the BLDC effectively. Advanced control strategies (like model predictive control) often have to be integrated to satisfy the control desires. Enhancing or proper tuning of a conventional algorithm results in achieving the desired performance. This paper presents a performance comparison of Enhanced PID and Model Predictive Control (MPC) applied to brushless direct current motor. The simulation results demonstrated that the PSO-PID is slightly better than the PID and MPC in tracking the trajectory of the reference signal. The proposed scheme could be useful algorithms for the system.

1. Introduction
Recently, BLDC has received much attention due to high efficiency, long operating range and noiseless operation as compared to brushed and induction motor. Complex nonlinear systems such as BLDC whose internal parameter values vary with environment or reference signal are mostly difficult to control. Control techniques are implemented to provide the desired stability and improve performance, which may reduce energy consumption and cost. Conventional controllers like PID are effective for linear systems, but their performances deteriorate when handling systems such as BLDC.

This enticed the implementation of the different kind of control techniques to the system such as hybrid neuro fuzzy [1], adaptive fuzzy logic [2], robust H∞ fuzzy [3], robust nonlinear control [4], neural network control [5], so that the desired performance and stability can be achieved. Nevertheless, the main issues with these control techniques are the requirement of prior knowledge, high expertise and tuning complexities. Enhanced PID control is quite simple, capable of dealing with nonlinearities and environmental variation.
Several methods are used to enhance performance of PID but this research work utilized particle swarm optimization (PSO) method. PSO is straightforward and has fewer parameters to tune. Model predictive control (MPC) uses a dynamic model of a system to predict the future direction of the desired response of the system. The primary concept of MPC is the frequent selection of a control effort to solve on line an optimal control problem, and this results in minimizing the performance cost function. MPC methodology has been in used in many areas of applications.

This paper focused on comparing the performances of enhanced PID and MPC in terms set-point tracking. The performances of the controllers were evaluated based on the commonly used control measures such as rise time, settling time and percent overshoot.

2. Methodology
2.1 Plant Dynamic Model
Mostly the representation of a system either mechanistic or empirical is a key in building reliable controller. Model describes the dynamics of a system which aid significantly in developing an effective control algorithm for the system.

From the block diagram shown in Fig. 1 using Laplace transformation the motor torque can be expressed as

\[ T_m(s) = K_c i_a(s) \]  \hspace{1cm} (1)

The back e.m.f is given by

\[ E_b(s) = K_b \omega(s) \]  \hspace{1cm} (2)

The mechanical load is expressed as

\[ \omega(s) = \frac{1}{J_s + B} \left( T_m(s) - T_L(s) \right) \]  \hspace{1cm} (3)

The angular position is mathematically represented as

\[ \theta(s) = \frac{\omega(s)}{s} \]  \hspace{1cm} (4)

Setting \( T_L = 0 \), the transfer function is given by

\[ \frac{\theta(s)}{E_a(s)} = \frac{K_c}{s \left[ (sL + R_e)(J_s + B) + K_c K_b \right]} \]  \hspace{1cm} (5)
Where $E_a$ is the armature e.m.f, $L$ is the inductance, $R_a$ is the armature resistance, $J$ is rotor inertia, $B$ is damping coefficient and $K_τ$ is the torque constant. It is assumed that $L/R_a$ and $L$ is set to zero in most cases.

Once the mathematical model of the plant is realized, the control algorithm can be design for the system.

2.2 Model Predictive Control

Model predictive control is a class of control algorithm that has been applied experimentally and through simulation to control several systems ranging from simple to complex, linear to nonlinear and multivariable plant. The algorithm uses the process model to predict the future output response of a system. At each control interval, the algorithm attempts to optimize future plant behaviour by computing a sequence of future manipulated variable adjustments. The first input in the optimal sequence is then sent to the plant, and the entire calculation is repeated at subsequent control intervals [6].

The predictive capability coupled with the classical feedback operation allows the algorithm to make adjustments that are smoother and closer to the optimal control action values. The controller comprises of an optimization problem at each time interval, $k$ and the main goals are to calculate a new control signal $u_k$ to be injected into the plant and concurrently consider the process constraints. The Fig. 2 shows the architecture of the model predictive control.

\[
\mathcal{J} = \sum_{k=d}^{N_p} \left( \hat{y}_k - r \right)^T Q \left( \hat{y}_k - r \right) + \sum_{k=0}^{N_p} \left( \Delta u_k^T R \Delta u_k \right) \tag{6}
\]

where $N_p$ depicts prediction horizon, $r$ is the set-point, $\hat{y}$ is the predicted output, $\Delta u$ is the predicted change in the control signal, $Q$ and $R$ are output error weight matrix and control weight matrix respectively.

![MPC architecture](image)
In developing MPC algorithm, the proper choice of design parameters such as control horizon \((N_C)\), prediction horizon \((N_P)\), model horizon \((N_M)\), sampling interval \((\Delta t)\), output error weight matrix \((Q)\) and control weight matrix \((R)\) is of paramount importance as this would lead to an effective and robust controller [7].

Control horizon \((N_C)\): This refers to the amount of manipulated variable moves to be optimized at control interval \(k\). Control horizon is selected such that \(N_C \leq N_P\). Small \(N_C\) results in an internally stable controller where high (or increasing) value results in computational burden and makes the MPC more aggressive.

Prediction horizon \((N_P)\): This refers to the amount of future control intervals the MPC controller must assess by prediction when optimizing its manipulated variables at control interval \(k\). Decreasing the value of \(N_P\) makes the controller more aggressive whereas results in less aggressive. The practice is to increase \(N_P\) until further increases has a slight effect on performance.

Sampling Period: Selecting the value of sampling period is a compromise between performance and computational effort. Small value of sampling period results increase in computational burden.

Q and R matrices: Since the goal is to minimize the cost function these matrices are selected as \(Q \geq 0\) (positive semi-definite symmetry matrix) and \(R > 0\) (positive definite symmetry matrix).

2.3 MPC Design
Since there is no systematic technique to choose the parameters, the aforementioned criteria was utilized as the guide, after some trials and error. The controller parameters are selected as: control horizon \((N_C) = 2\), prediction horizon \((N_P) = 30\) and sampling interval \((S_T) = 0.0002\).

2.4 Enhanced PID
Several techniques to enhance PID exist, but this paper utilizes particle swarm optimization method. Particle swarm optimization is a population-based search procedure introduced by [8] in 1995. It is straightforward to implement because of few parameters to update. Appropriate selection of these parameters values are essential to achieve the desired results and speedy convergence [9]. For detailed explanation on the parameters selection could be found in [9], [10]. The flow chart in Fig. 3 illustrates the procedure for searching of the optimal gains values of PID. The PSO algorithm used are number of particles \(N = 20\), Dimension \(D = 3\), number of iteration \(it = 50\), cognitive constant \(c = 1.98\), social constant \(s = 0.89\), inertia weight \(iw = 0.85\).

The optimal gains of the PID controller were obtained through using the PSO algorithm shown in Fig. 3 and running the simulation on MATLAB/SIMULINK environment.
The realized PSO-PID parameters gains are $k_{p_{ps0}} = 0.12$, $k_{i_{ps0}} = 0.01$ and $k_{d_{ps0}} = 0.02$. Since the gains were PSO-PID obtained the performance of the controller could be assessed through simulation.

3. Simulation Results and Discussion

The simulation parameters used throughout this paper are shown in the Table 1. The open-loop response from the system indicated that error exists as illustrated in Fig. 4 when a step signal of 1 radian was injected into the system.

**Table 1. Simulation Parameters**

| Parameters | $B$ (Nms) | $J$ (Nms$^2$) | $K_f$ | $L_a$ (H) | $R_a$ (Ω) | $i = 50$ |
|------------|-----------|---------------|-------|-----------|-----------|-----------|
| Value      | $0.708 \times 10^{-4}$ | $11 \times 10^{-6}$ | 36.8  | $2.758 \times 10^{-6}$ | 4.31      | $i = 50$  |
Therefore, an enhanced PID and MPC were designed as described above to minimize the error. The gains of the classical PID parameters are \( k_p = 0.104 \), \( k_i = 0.010 \) and \( k_d = 0.011 \). The performances of the controllers are shown in Fig.5, the commonly used control measures as mentioned earlier were utilized to evaluate the performances and the results were presented in Table 2. In addition, a variable reference signal was also used to assess the tracking capability of the controllers as depicted in Fig 6. The controllers have tracked well the reference signals, but the PSO-PID has faster response compared to the PID and MPC. Nevertheless, proper tuning of MPC may yield promising results.

| Control Measure       | Controllers | PID    | PSO-PID | MPC       |
|-----------------------|-------------|--------|---------|-----------|
| Rise time \([t_r](s)\) | 0.1242      | 0.1074 | 0.8128  |
| Settling time \([t_s](s)\) | 0.3972      | 0.5299 | 1.6590  |
| Maximum overshoot \([M_p](s)\) | 8.1870      | 1.5297 | 2.3940  |
4. Conclusion

The paper has presented the enhanced PID and MPC for BLDC motor position control. The results obtained demonstrated that the PSO-PID performed slightly better than the classical PID and MPC. Even with the variable reference signal, the PSO-PID has better response than the MPC and PID. Although disturbance rejection capability of the controllers was not evaluated, with proper tuning of MPC, a promising result could be achieved. The proposed controllers could serve as an alternative control architecture for the brushless direct current motor.

References

[1] M. Gokbulut, B. Dandil and C. Bal: TAINN Springer Vol. 83-87 (2006), p. 125-132
[2] E. N. Binti Sabudin: Development of Position Tracking of BLDC Motor using Adaptive Fuzzy Logic Controller (Universiti Tun Hussein Onn, Malaysia, Masters Thesis, 2012).
[3] W. Assawinchaichote and N. Chayaopas: Applied Mathematics and Information Sciences Vol. 10 (2016), p. 987-995
[4] N. Hemati, J. S. Thorp and M. C. Leu: IEEE Transactions on Industrial Electronics Vol. 37 (1990), p. 460-468
[5] M. Gokbulut and A. Tekin: Int. J. Engng Ed. Vol. 22 (2006), p. 197-204
[6] H. P. Halvorsen: Model Predictive Control in Labview (Faculty of Technology, Porsgrunn, Norway, 2011)
[7] A. Bemporad, M. Morari and N. L. Ricker: Model Predictive Control Toolbox for use with MATLAB (Math Works, Inc., U. S. A., 2014)
[8] J. Kennedy and R. C. Eberhart: Proceeding of IEEE international conference on neural networks Vol. IV, pp. 1942-1948. IEEE service center, Piscataway, NJ, 1995
[9] M. S. Gaya, A. A Bisu, S. N. S. Salim, I. S. Madugu, L. A. Yusuf, M. L Kaurangini, M. T. M. Khairi: International Review of Automatic Control Vol. 9 (2016), p. 167-204.
[10] S. Sumathi and P. Surekha: Computational Intelligence Paradigms- Theory and Application using MATLAB (CRC Press USA 2010)