Comparison between Compensated and Uncompensated PD with Cascade Controller Design of PMDC Motor: Real Experiments

Sopheak Yean a,1,*, Sarot Srang a,2

a Dynamics and Control Laboratory, MIT Research Unit, Institute of Technology of Cambodia, Phnom Penh, Cambodia
1 sophek.yean@gsc.itc.edu.kh; 2 srangsarot@itc.edu.kh
* Corresponding Author

1. Introduction

DC motors have wide applications in industrial control system (mechatronics system, robotics and low to medium power machine tools) because they are easy to model and control. The PMDC machines have a simpler construction since the field winding of a DC motor is replaced by permanent magnet (PM). Consequently, they have become one of the most commonly used machines in many applications such as electrical vehicles, robotic manipulators, trams, or motion-controlled devices, etc [1]-[11]. Each low-cost PMDC motor produced may have different dynamical and electrical properties such as back electromotive force and torque constant, rotor resistance, friction coefficients.

As stated in [12]-[14], over the year, several methods are used to estimate these unknown parameters of DC motors such as hybrid equivalent circuit model, step response, dynamic load variation, closed-loop error approach, least-square parametric estimation, least-squares approximation technique, and a closed-loop disturbance observer. Some of the mentioned methods may need applying a varying voltage to the motor and recording the motor data shown on the measuring equipment before calculating the unknown parameters by comparing motor data with the model equation of the motor. One obstacle of using this method is the need to have third-party measuring equipment for recording data which demands high cost.
There is a large number of parameter identification methods, namely: the algebraic identification methods, the recursive least square method, the inverse theory, the least square method, the moments method [15]-[21]. When applying to DC motor parameter identification, these methods work so well for steady state response, but only some parameters can be identified. The first difficulty with these methods is to conduct many experiments at different steady state responses. Another difficulty is that in order to get all parameters for complete modeling, there needs to be distinct experiment setup to conduct for obtaining transient responses. The last disadvantage of these methods is that they require 3 measurements such as voltage, current, and angular velocity. For our method, only angular velocity measurement is needed. That thanks to the simplification technique. Furthermore, we can estimate all the lumped parameters with one-go experiment.

For control method, there are as follows: Proportional-Integral (PI), Proportional-Integral-Derivative (PID) or bipositional, Continuous PID Control Design Method, A state-space Control Design Method, Discrete PID Control Design Method [22]-[25]. The most popular among them is the PID controller since it is quite simple and easy to be implemented in a real hardware system [26]. However, in [27] the PID controller also has some weaknesses. Any slightest change in setpoint will affect the whole system’s performance, and it is very hard to tune the parameters for PID controller. With our simplified model, simpler controller like PD is very suitable.

As implemented in [28]-[38], the hybrid unscented Kalman filter is used to estimate the effect of Stribeck + viscous friction of a DC motor. The estimated parameters compensate to the controller of the motor. The findings reveal that the system with friction compensation shows much better performance. It shows that by using one or another variance of the Kalman filter, it is possible to effectively estimate the unknown parameters of the DC motor. In this paper, all of the parameters are lumped, and we choose Unscented Kalman Filter for parameter estimation because the state-parameter model of the dynamic is nonlinear.

We have seen numerous previous researches comparing only two types of controllers but they never subdivide them into subtypes—compensated and uncompensated respectively. Therefore, our research problem is to investigate the different outcomes between PD controller (compensated and uncompensated) and Cascade controller (compensated and uncompensated). The results will help us conclude which control architecture is superior to the others.

This paper is structured as follows: Section 2 presents the dynamics modeling, controller design and describes the obtained UKF algorithm for parameters estimation. Section 3 presents the experiment results, discussion and conclusion are stated in section 4.

2. Method

2.1. Model DC Motor

The following DC motor model and controller design are derived based on the work as seen in [15], [16], [39]-[44].

In Fig. 1 both the back EMF (electromotive-force voltage) and the electromagnetic torque depend direct in proportionality with speed and current:

\[ T_a = K_t l_a \]  \hspace{1cm} (1)
\[ V_b = K_b \omega \]  \hspace{1cm} (2)

where \( K_t \) is torque constant and \( K_b \) is back EMF constant. Electromagnetic theory and classical mechanics form the basis for the development of electromechanical system models. The differential equation of the armature circuit is

\[ V_a = L_a \frac{d l_a}{dt} + R_a l_a + V_b \]  \hspace{1cm} (3)
In many practices of PMDC motor, the inductance $L_a$ in the armature circuit is very small which cause the dynamic response of the electrical part to be much faster than that of mechanical part (usually 100 to 1000 times faster). Therefore, only the dynamic of the mechanical part is of importance. Once inductance is ignored ($L_a \approx 0$), then (3) becomes

$$V_a = R_a i_a + V_b$$

(4)

Application of Newton’s second law of motion yields the torque balance equation

$$J \ddot{\omega}(t) + T_f = T_a$$

(5)

where $T_f$ friction torque effected by Coulomb friction $T_d$ and viscous friction $D \omega(t)$. Then the friction torque is modeled as

$$T_f = T_d \text{sign}[\omega(t)] + D \omega(t).$$

(6)

Substituting (6) into (5) yields

$$J \ddot{\omega}(t) + T_d \text{sign}[\omega(t)] + D \omega(t) = T_a.$$  

(7)

Using (1), (2), (4), and (7) the dynamic of the DC motor can be written as

$$\dot{\omega}(t) = - \left( \frac{R_a D + K_t K_b}{R_a J} \right) \omega(t) + \frac{K_t}{R_a J} V_a - \frac{T_d \text{sign}[\omega(t)]}{J}.$$  

(8)

where $V_a$ is voltage applied to the armature, $R_a$ is the internal resistance of the armature, $J$ is the moment of inertia of the motor

Let

$$a = \left( \frac{R_a D + K_t K_b}{R_a J} \right)$$

and

$$b = \frac{K_t}{R_a J}, c = \frac{T_d}{J}.$$  

Then (8) can be reduced to

$$\dot{\omega}(t) = -a \omega(t) + b V_a - c \text{sign}[\omega(t)].$$  

(9)

We see that the equation (9) is a nonlinear state equation for velocity model of a DC motor. The parameters $a$, $b$, and $c$ of the models can be numerically estimated by using UKF. From the equation (9), we can sketch block diagram for velocity model as Fig. 2.

2.2. PD Controller Design by Root locus

Consider a position controller architecture with the PD controller as shown Fig. 2. The governing equation of the control system Fig. 2 can be written as

$$\left( \dot{\theta}_d - \dot{\theta} \right) + (b K_d + a)(\dot{\theta}_d - \dot{\theta}) + b K_d z (\theta_d - \theta) = 0.$$  

(10)
Let the error between desired and actual position \( e_\theta = \theta_d - \theta \), then \( \dot{e}_\theta = \dot{\theta}_d - \dot{\theta} \) and \( \ddot{e}_\theta = \ddot{\theta}_d - \ddot{\theta} \). The equation (10) can be rewritten as
\[
\ddot{e}_\theta + (bK_d + a)e_\theta + bK_dze_\theta = 0 \quad (11)
\]
Equation (11) is a second order differential equation which has characteristic solution as follow.
\[
\lambda^2 + (bK_d + a)\lambda + bK_dz = 0 \quad (12)
\]
Equation (12) has roots which can be real or complex numbers defined as below.
\[
\lambda_1 = \frac{1}{2}((-a - bK_d) + \sqrt{(a - bK_d)^2 - 4bK_dz})
\]
\[
\lambda_2 = \frac{1}{2}((-a - bK_d) - \sqrt{(a - bK_d)^2 - 4bK_dz})
\]
Let \( K_p = K_dz \). The roots of (12) can be analyzed in terms of parameter as below.
When \( K_d = 0 \), then \( \lambda_1 = 0 \), and \( \lambda_2 = -a \).
When \( K_d \to \infty \), then from limited techniques, \( \lambda_1 = -z \) and \( \lambda_2 = -\infty \).

When \( \Delta = \sqrt{(a - bK_d)^2 - 4bK_dz} = 0 \), then (12) has double roots \( \lambda_1 = \lambda_2 = \frac{1}{2}(-a - bK_d) \) and values of \( K_d \) and \( K_p \) are obtained as follow.
\[
K_d = \frac{-a + 2z + 2\sqrt{z(z - a)}}{b} \quad \text{and} \quad K_p = K_dz \quad (13)
\]

![Fig. 2. Position control architecture with PD controller](image)

There are two double roots (Delta = 0) as illustrated by root locus in Fig. 3, we choose the left double root because we prefer the critically-damped response that is the fastest one without overshoot [45].

For the friction compensation block, the velocity \( \dot{\theta} \) is taken from the desired output of the motor. From root locus techniques, we can sketch a set of solutions of characteristic equations by parameter \( K_d \) as in Fig. 3. Here, we see that all solution of characteristic equation is on the left side of s-plane. Hence the designed system is stable.

By selecting the value of \( z \) in terms of \( a \), (e.g., \( z = 1.2|a| \to 1.5|a| \)), proportional gain \( K_p \) and derivative gain \( K_d \) for the designed PD controller can be found using equation (13).
2.3. Cascade Controller Design

Consider a position controller architecture with the Cascade controller as shown in Fig. 4. The governing equation of the control system Fig. 4 can be written as

\[
\ddot{\theta} + (bK_d + a)\dot{\theta} + bK_dzb\dot{\theta} = 0
\]  

(14)

and its characteristic form is

\[
\lambda^2 + (a + bK_z)\lambda + bK_t = 0
\]  

(15)

Comparing (15) to the standard form of second order differential equation

\[
\lambda^2 + 2\zeta\omega_n\lambda + \omega_n^2 = 0
\]  

(16)

\(k_1\) and \(k_2\) can be found as

\[
k_1 = \frac{\omega_n^2}{b}
\]  

(17)

\[
k_2 = \frac{2\zeta\omega_n - a}{b}
\]  

(18)

![Root Locus](image)

**Fig. 3.** Root locus of position control system with PD controller

![Cascade Controller System](image)

**Fig. 4.** Position control architecture with Cascade controller
2.4. Unscented Kalman Filter Algorithm

Unscented Kalman filter (UKF) is a variant of Kalman filter that is used to estimate state of a nonlinear dynamical system. The dynamical system is modeled as stochastic differential equation with assumption that both state dynamic and measurement are disturbed by Gaussian noises. Unlike other variants of Kalman filter that compute the state propagation and output estimation based on linear mapping function directly or obtained from linearization, UKF uses unscented transform to propagate generated sigma points via nonlinear function directly.

2.4.1. Unscented Transformation

UKF consists of two steps, time update and measurement update. The proceeding step to these two is for the selection of sigma points. The points are taken from the source assumed to be Gaussian and map them on the target Gaussian through some nonlinear function, and the new mean and variance of the transform are calculated. The process is called Unscented Transformation (UT). The UT is presented as the following steps 1) computing set of sigma points, 2) assigning weights to each sigma point, 3) transforming the points through nonlinear function, 4) computing weighted and transformed points, 5) Computing mean and variance of the new Gaussian [46], [47].

- The selection of sigma points and weight

Sigma points generation

\[ X_{[0]} = \hat{x} \]

\[ X_{[i]} = \hat{x} + \left( \sqrt{(L + \lambda)P_x} \right)_i \quad i = 1, \ldots, L \]  \hspace{1cm} (20)

\[ X_{[i]} = \hat{x} - \left( \sqrt{(L + \lambda)P_x} \right)_{i-L} \quad i = L + 1, \ldots, 2L \]  \hspace{1cm} (21)

Mean contraction weights

\[ W_0^{[m]} = \frac{\lambda}{(L + \lambda) + (1 - \alpha^2 + \beta)} \]  \hspace{1cm} (22)

\[ W_i^{[m]} = \frac{1}{2(L + \lambda)} \quad i = 1, \ldots, 2L \]  \hspace{1cm} (23)

Covariance contraction weight

\[ W_0^{[c]} = \frac{\lambda}{(L + \lambda)} \]  \hspace{1cm} (24)

\[ W_i^{[c]} = \frac{1}{2(L + \lambda)} \quad i = 1, \ldots, 2L \]  \hspace{1cm} (25)

where \( X \) is sigma points matrix with dimension \( L \). \( X \) has mean \( \hat{x} \) and covariance matrix \( P_x \). \( X \) is to be propagated through \( Y = g(X) \). \( \lambda = \alpha^2(L + k) - L \) is a scaling parameter. \( \alpha \) determines the spread of the sigma points around \( \hat{x} \), usually set to a small value. \( k \) is a secondary scaling parameter, and \( \beta \) is used to incorporate prior knowledge of the distribution of \( X \). \( \left( \sqrt{(L + \lambda)P_x} \right)_i \) is \( i \)th row of the matrix square root. It is to be propagated through the nonlinear function.

- Transform sigma points through nonlinear function

\[ Y_i = g(X_{[i]}) \quad i = 0, \ldots, 2L \]  \hspace{1cm} (26)
Mean and covariance approximation

\[
\hat{y} \approx \sum_{i=0}^{2L} W_i^{[m]} y_i
\]

\[
P_y \approx \sum_{i=0}^{2L} W_i^{[c]} \{y_i - \hat{y}\} \{y_i - \hat{y}\}^T
\]

where \(\hat{y}\) and \(P_y\) are mean and covariance of \(Y\) respectively. They are approximated using a weighted sample mean and covariance of the posterior sigma points.

### 2.4.2. UKF State Estimation

The UKF algorithm is provided below. The full algorithm explanation can be found in [31], [48]-[51]. In this paper, we only give a brief explanation of the algorithm. The time update and measurement update are like those of the standard Kalman filter except for the sigma points and weight calculation.

We have the state space model

\[
x_{k+1} = f_d(x_k, u_k) + \sqrt{Q_d} v_k
\]

\[
y_k = h_d(x_k, u_k) + \sqrt{R} w_k
\]

Creating the sigma points equations

\[
X_{k|k-1} = [\hat{x}_{k|k-1} \ldots \hat{x}_{k|k-1}] + \sqrt{L + \lambda} \begin{bmatrix} 0 & \sqrt{P_{k|k-1}} & -\sqrt{P_{k|k-1}} \end{bmatrix}
\]

The sigma points are transformed through the nonlinear function

\[
Y_{k|k-1} = h_d(X_{k|k-1}, u_k)
\]

Calculating the mean in measurement space from the propagated points

\[
\hat{y}_{k|k-1} = Y_{k|k-1} w_m
\]

where \(w_m = [W_0^{(m)} \ldots W_{2L}^{(m)}]^T\)

Calculating the covariances in measurement space

\[
P_{xy,k|k-1} = X_{k|k-1} \bar{W} Y_{k|k-1}^T
\]

\[
P_{yy,k|k-1} = Y_{k|k-1} \bar{W} Y_{k|k-1}^T + R
\]

where \(\bar{W} = (I - [w_m \ldots w_m]) \times \text{diag}\left(\begin{bmatrix} W_0^{(c)} \ldots W_{2L}^{(c)} \end{bmatrix}\right) \times (I - [w_m \ldots w_m])^T\)

Computing the Kalman gain

\[
K_k = P_{xy,k|k-1} P_{yy,k|k-1}^{-1}
\]

Correcting the mean and covariance

\[
\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k (y_k - \hat{y}_{k|k-1})
\]
\[ P_{k|k} = P_{k|k-1} + K_k P_{y|k|k-1} K_k^T \] (38)

- Time update equations

Creating the set of sigma points

\[ X_{k|k}^{(r)} = \left[ \hat{x}_{k|k} \ldots \hat{x}_{k|k} \right] + \sqrt{\Lambda} \left[ 0 \quad \sqrt{P_{k|k}} \quad -\sqrt{P_{k|k}} \right] \] (39)

where \( X_{k|k}^{(r)} \) is the set of resampled sigma points.

Propagating the sigma points

\[ X_{k+1|k} = f_d \left( X_{k|k}^{(r)} , u_k \right) \] (40)

Calculating the predicted mean and covariance

\[ \hat{x}_{k+1|k} = X_{k+1|k} w_m \] (41)

\[ P_{k+1|k} = X_{k+1|k} \bar{W} X_{k+1|k}^T + Q_d \] (42)

3. Experiment Results

In Fig. 5 shows the Apparatus, which was built with a low-cost PMDC motor, H-bridge driver, and Arduino Due microcontroller, is used for the experiment with hardware testing in MATLAB Simulink. The PMDC motor has gear ratio 10.2, optical encoder 100 ppr and maximum 24 voltage. The other parameters of the PMDC motor are unknown. The desired dynamic angular velocity is chosen as \( V_a = 10 \sin(2\pi \times 0.1t) \) (rad) for observation during the experiment. The desired velocity is mathematically calculated from the desired position. Compensation for dynamic reference.

![Fig. 5. Apparatus for experiments](image-url)
algorithms is implemented in microcontroller to execute the PWM signal for DC motor drive. A 24VDC gear-motor is a powerful motor to drive the position control system. It comes with the photoelectric encoder output, planetary gear reducer and 10.2 gear ratio, which provide 100 rpm with 24VDC rated voltage. H-Bridge motor driver which is allows controlling the direction and position of DC motor.

3.1. UKF Implementation

We use the UKF algorithm to estimate parameters $a$, $b$, and $c$ of the velocity model. For defining parameters as state variables, Let $x_1 = \dot{Q}_{\text{est}}$, $x_2 = a_{\text{est}}$, $x_3 = b_{\text{est}}$, and $x_4 = c_{\text{est}}$ and $x = [x_1, x_2, x_3, x_4]^T$, $T_s = 0.01s$, $Q_c = 1e-6 \times \text{diag}([1; 0.1; 0.1; 0.1])$ and $R = 1e-6 \times \text{diag}([2; 2])$. We performed the estimator tuning by selecting an appropriate ratio between $Q_c$ and $R$. Choosing a smaller $Q_c$ means that we have greater confidence in our model while choosing smaller $R$ show greater confidence in the measurement. The velocity model (9) is rewritten as a stochastic nonlinear system as processing (State equation) and measuring (Output equation) model below.

$$\dot{x} = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} -x_1x_2 + x_3u - x_4\text{sign}(x_1) \\ 0 \\ 0 \\ 0 \end{bmatrix} + \sqrt{Q_c}v(t) \quad (43)$$

$$y_k = [1 \ 0 \ 0 \ 0] \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \sqrt{R}w_k \quad (44)$$

Discretize (42) for implementing UKF algorithm, we obtain

$$x_{k+1} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}_k + T_s \begin{bmatrix} -x_1x_2 + x_3u - x_4\text{sign}(x_1) \\ 0 \\ 0 \\ 0 \end{bmatrix} + \sqrt{T_sQ_c}v_k \quad (45)$$

$$y_k = [1 \ 0 \ 0 \ 0] \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}_k + \sqrt{R}w_k \quad (46)$$

3.2. Experiment Lumped Parameter Estimation

In Fig. 6 illustrates the model of parameter estimation for the experiment. We used input signal $V_a = 10\sin(2\pi \times 0.1t)$ (rad) with velocity model in (9). The signals of the state equation and output equation are eliminated and substituted by angular velocity signal and it is fed to UKF block. The angular velocity is approximated from angular position at shaft of gearbox of DC motor as shown in Fig. 6.

Experimental results as shown in Fig. 8, Fig. 9, and Fig. 10 reveal that estimated parameter $a_{\text{est}}$, $b_{\text{est}}$ and $c_{\text{est}}$ of the DC motor converge to 12.23 (1/s), 51.31 (rad/s² / v) and 27.99 (Nm/kg.m²) at the time of 30 second respectively with the appropriately tuned parameters of the UKF algorithm. These values are used to compensate for the feed-forward controller. We have shown that UKF could estimate the value of these parameters satisfactorily.
Fig. 6. Experiment model for estimated parameter $a$, $b$, $c$

Fig. 7. Experiment result for velocity

Fig. 8. Experiment results for estimated parameter $a$
Sopheak Yean (Comparison between Compensated and Uncompensated PD with Cascade Controller Design of PMDC Motor: Real Experiments)

3.3. Position Tracking using PD Controller

In Fig. 11 is used for experiment model in Simulink model for position control architecture with PD controller. In this real experiment, we chose lumped parameters to $a_{\text{est}} = 12.23 \text{ (1/s)}$, $b_{\text{est}} = 50.31 \text{ (rad/s}^2/\text{v)}$, $c_{\text{est}} = 27.99 \text{ (Nm/kg.m}^2\text{)}$ and input signal $V_a = uk = 10 \sin(2\pi * 0.1t) \text{ (rad)}$, $T_s = 0.01 \text{ s}$. For the feed-forward ensures that the input to the motor is always regulated based on the motor’s dynamic and electrical property.

![Experiment model for position control with PD controller](image)
In Fig. 12 shown the experiment results for position control using PD controller with friction compensation. We see that the actual position $\theta_{\text{actual}}$ is starting to converge to the desired position $\theta_d$ at a time of 2 seconds while the error values of the experiment result for error position control architecture with Compensated PD controller with the largest error within 0.028 (rad) from the actual value has shown in Fig. 14.

In Fig. 13 shown the experiment results for position control using PD controller without friction compensation. We see that the actual position $\theta_{\text{actual}}$ is starting to converge to the desired position $\theta_d$ at a time of 2 seconds while the error values of the experiment result for error position control architecture with uncompensated PD controller with the largest error within 0.24 (rad) from the actual value has shown in Fig. 15.
3.4. Position Tracking using Cascade Controller

In Fig. 16 is used for experiment model in Simulink model for position control architecture with Cascade controller. In this real experiment, we chose lumped parameters to $a_{est} = 12.23 \ (1/s)$, $b_{est} = 50.31 \ (\text{rad}/s^2/\nu)$, $c_{est} = 27.99 \ (\text{Nm/kg/m}^2)$, $\zeta = 1$, $\omega_n = 2 \times \pi \times 4$, $k_1 = \omega_n^2/b$, $k_2 = (2\zeta\omega_n - a)/b$ and input signal $V_a = 10\sin(2\pi \times 0.1t) \ (\text{rad})$, $T_s = 0.01 \ s$. For the feed-forward ensures that the input to the motor is always regulated based on the motor’s dynamic and electrical properties.

![Fig. 16. Experiment model for position control with Cascade controller](image)

In Fig. 17 shown the experiment results for position control using Cascade controller with friction compensation. We see that the actual position $\theta_{actual}$ is starting to converge to the desired position $\theta_d$ at a time of 2 seconds while the error values of the experiment result for error position control architecture with compensated Cascade controller with the largest error within 0.029 (rad) from the actual value has shown in Fig. 19.

In Fig. 18 shown the experiment results for position control using Cascade controller without friction compensation. We see that the actual position $\theta_{actual}$ is starting to converge to the desired position $\theta_d$ at a time of 2 seconds while the error values of the experiment result for error position control architecture with uncompensated Cascade controller with the largest error within 0.8 (rad) from the actual value has shown in Fig. 20.

In Fig. 21 shown the final experiment results of the combination of all the 4 controllers (2 compensated controllers and other 2 uncompensated controllers). We see that the dark green curve of the Uncompensated Cascade controller surges highly to 0.8 (rad) while the orange graph of the
Uncompensated PD controller rises mildly to 0.24 (rad). The two unparallelled graphs prove big errors, which is a drawback of the uncompensated controllers.

In contrast, if we take glance at the two compensated graphs (the blue and red ones), they are well-overlapped and run alongside each other. This shows that they almost have zero errors.

**Fig. 17.** Experiment results for desired and actual theta of Cascade controller with friction compensation

**Fig. 18.** Experiment results for desired and actual theta of Cascade controller without friction compensation

**Fig. 19.** Experiment results for Position error of compensated Cascade controller
Sopheak Yean

Comparison between Compensated and Uncompensated PD with Cascade Controller Design of PMDC Motor: Real Experiments

4. Conclusion

In this research, we derive models of a DC motor with three lumped parameters $a_{est}$, $b_{est}$, and $c_{est}$ for estimation by using UKF. To simplify the complexity of the parameter estimation, we approximate the models including the Coulomb frictional effect. Then, we use the three estimated parameters to design PD controller by root locus for position control and Cascade controller. In the parameter estimation, dynamic reference is used to capture all possible physical properties of the motor. The real experiment was conducted. Based on this experiment, the Uncompensated graphs show bigger errors than the compensated ones. In spite of trying our best to tune the Uncompensated ones, they are not still as good as the compensated ones. Therefore, the compensated controllers—either PD or Cascade—outperform the Uncompensated controllers. The maximum estimation bias of the compensated PD controller is only within 0.28% whereas that of the compensated Cascade
controller is only within 0.29%. Meanwhile, the maximum estimation bias of the Uncompensated PD controller is only within 2.4% whereas that of the Uncompensated Cascade controller is only within 8%. Not knowing any parameters for compensation between both Cascade and PD controller, the expected result is not good.

One major limitation of the study is that the enterprise that produced the motor (used in this experiment) does not specify the parameters of the motor. If we know the parameters clearly, we can design controllers and apply them on mobile robots, robot arm, and UAV, etc.

In the future, we will apply that work with Online Parameter Estimation of Permanent Magnet DC Motor using an Unscented Kalman Filter.

Author Contribution: All authors contributed equally to the main contributor to this paper. All authors read and approved the final paper.

Funding: This research was funded by National Polytechnic Institute of Cambodia.

Conflicts of Interest: The authors declare no conflict of interest.

References


[43] T. N. Gücin, M. Biberoğlu, B. Fincan and M. O. Gülbahçe, “Tuning cascade PI(D) controllers in PMDC motor drives: A performance comparison for different types of tuning methods,” 2015 9th International


