CHAPTER 4

A DUAL GENERALISED PREDICTIVE CONTROLLER FOR THE SPEED CONTROL OF PMBLDC MOTOR

4.1 INTRODUCTION

The speed and position of PMBLDC motor is normally controlled in a multi loop structure with inner current loop and outer speed loop. The loop delay occurs in the current loop and speed loop, when the motor is controlled by a single system. Similarly, when the motor is operating in a distributed control environment, the delay arises in the communication network. When the loop delay is not dominant, proportional plus integral control is very effective. However, the difficulties caused by the delay time cannot be recognised by the PI controller. So, the time management between inner current loop and the outer speed loop plays an important role in the industrial motor control systems. Because of loop delay, the noise is created in the switches and fluctuations in the motor power supply and hence the motor settles at the desired set point with more oscillations. If the control loop delay is properly compensated, lot of power can be saved because of reduction in losses in the internal components of the drive. Therefore, it is desired to develop a controller that has the ability to manage the time delay and adapt online, according to the environment in which it works to yield satisfactory control performance.

To overcome these drawbacks, advanced control schemes such as Smith predictor performs well only if the motor model is accurate but the performance degrades with inaccuracy in motor system gain and delay time. Self-tuning regulators can be used to control motors with unknown dynamics. Nevertheless, adaptive or self-tuning algorithms lack robustness when applied to system with incorrectly modeled motor delay or model order. So, an alternative design approach is required for conventional cascade speed control scheme.
4.2 CONVENTIONAL CONTROLLER

The prominent controller in the motor control drive is the cascade PI/PID controller. The controller design and tuning of PI controller coefficients, to provide optimal performance is always a contentious issue. Another issue is that because of its structure, a PID controller acts only after a disturbance has moved the process output from its desired trajectory. Thus PI controller is practically complex and requires lot of manipulations and retuning methods. The cascade control scheme is the common standard for the control of electric drive systems as shown in Fig.4.1.

![Block diagram of cascade PI speed control for PMBLDC motor system](image)

Fig. 4.1. Block diagram of cascade PI speed control for PMBLDC motor system

The speed controller $G_i(s)$ computes an output signal that is the torque needed to accelerate the motor to the desired speed. The desired current $I_d^*(s)$ that the motor needs to produce the torque is calculated from the mathematical model of the motor. The inner loop controls the current that is needed to produce the torque. The output of the controller $G_i(s)$ is used as set point to the power converter which produces the necessary input voltage to the motor. The transfer function from the rotor current set point $I_d^*(s)$ to the rotor current is the closed loop transfer function of the inner loop and is given by

$$G_{cl}(s) = \frac{I_d(s)}{I_d^*(s)} = \frac{G_e(s)G_i(s)G_{me}(s)}{1 + G_e(s)G_i(s)G_{me}(s)} \quad (4.1)$$

where $G_i(s)$ is transfer function model of the power converter

$G_{me}(s)$ is the electrical part of the PMBLDC motor.

$G_{pn}(s)$ is the mechanical part of the PMBLDC motor.
If the gain of the speed controller is large, then the inner closed loop controller will approach to unity and will also be quite insensitive to variations in the power converter and/or motor transfer functions. Non-linear behaviour of the motor and converter can often be modeled by transfer functions with variable coefficients. From the output of the speed controller there are three quite simple systems in series, the speed controller, the current control loop $G_{\text{ct}}(s)$ and the mechanical part of the motor model $G_{\text{me}}(s)$. Thus the cascade structure eliminates many of the inherent complexities in the power converter and motor dynamics. However, windup problem in cascade control systems needs special attention. Also, both speed and torque tracking objectives are achieved in matched and mismatched parameter case. The measured and unmeasured disturbances are to be effectively rejected and the motor must run at desired speed at constant load. In addition, non-minimum phase characteristics of the motor and motor constrain lead to a problem. To overcome these drawbacks, this work applies a dual generalised predictive controller (GPC) to the PMBLDC Motor system in an existing cascade control structure. GPC control is an online optimisation approach to satisfy multiple, changing performance criteria, under existing PMBLDC Motor control hardware scheme. Therefore, the multirate based general predictive control law is proposed for the conventional cascaded PI-PI scheme, in which both the current and speed control loops are configured with new general predictive control algorithm.

4.3 PROPOSED CASCADED DUAL GPC

4.3.1 Basics of GPC

Generalised predictive control developed by Clarke et al [48,49,50,70] is one of the most popular predictive control strategies. Predictive control, commonly grouped as model predictive control (MPC), uses a model of the plant to predict the output in the future. The interest in developing a multirate cascade control system using GPC is the possibility to control the speed and current together. To realise this, two GPC control algorithm is computed as shown in the Fig.4.2 where

- $y_\text{o}$ is the resulting control signal applied to the motor.
- $u_\text{in}$ is the inner signal coming from the minimisation of GPC$\omega$. 

60
The speed and current loops of the cascaded structure require the two GPC algorithms and consequently the minimisation of cost function. It is also necessary to express the numerical models of GPC to the individual loops.

4.3.2 Numerical Model of GPC

A cascaded predictive strategy first requires the definition of the numerical model for both speed and current loops of the PMBLDC Motor system. Often, the time varying system dynamics can be described by a controlled autoregressive and integrated moving average (CARIMA) model for a general ‘r’ inputs and ‘n’ outputs system with \( u_d \) measurable input disturbances. This can be expressed as

\[
A(q^{-1})y(t) = B(q^{-1})u(t-d) + \frac{C(q^{-1})e(t)}{\Delta} + D(q^{-1})u_d(t-d)
\]  

(4.2)

where

\( q^{-1} \) is delay operator

\( y(t) \) is the predictive controller output vector

\( u(t-d) \) is the manipulated variable with delay

\( u_d(t-d) \) is the manipulated variable with delay

\( e(t) \) is uncorrelated random noise

\( \Delta \) is the difference operator \( 1-q^{-1} \). In most of the cases \( C(q^{-1}) = 1 \)
A, B, C and D are matrices of polynomials in the delay operator \((q^{-1})\) with dimensions \(n \times n\), \(n \times r\), \(n \times n\) and \(n \times n\), respectively.

The output vector for 'n' output system is expressed as
\[
y(t) = [y_1(t) \ldots y_n(t)]^T
\]
The manipulated variable vector for 'r' input system is expressed as
\[
u(t) = [u_1(t) \ldots u_r(t)]^T
\]
The 'n' manipulated variable disturbance vector is expressed as
\[
u_d(t) = [u_{1n}(t) \ldots u_{nn}(t)]^T
\]
The uncorrelated random noise vector for 'n' output is expressed as
\[
\varepsilon(t) = [\varepsilon_1(t) \ldots \varepsilon_n(t)]^T
\]

### 4.3.3 Controller Formulation for PMBLDC motor

The main objective of the GPC is to detect the changes in manipulated variable with a system delay time of one sampling instant is

\[
\Delta u(t) = u(t) - u(t - d)
\]

This would make the output best match to a target value \(y_r(t + N)\) in the presence of disturbances and system constraints. In long range predictive control, a predicted projection of outputs \(y_r(t)\) over p-future time intervals \((t + N_1)\) to \((t + N_2)\) is matched to the set point trajectory \(\omega\), by prescribing the sequence of m-future moves

\[
\Delta u(t), \ldots, \Delta u(t + m - 1)
\]

where

- \(N\) is future control increments
- \(N_1\) is the minimum costing horizon
- \(N_2\) is the maximum costing horizon
The GPC model of the current control loop is expressed as

\[
A_i(q^{-1})y_i(t) = B_i(q^{-1})u(t - d) + \frac{\varepsilon_i(t)}{\Delta}
\]  

(4.4)

The GPC model of the speed control loop is expressed as

\[
A_\omega(q^{-1})y_\omega(t) = B_\omega(q^{-1})y_\omega(t) + \frac{\varepsilon_\omega(t)}{\Delta}
\]  

(4.5)

where

- \(A_\omega, A_i, B_\omega, B_i\) are polynomials in the backward shift operator \(q^{-1}\).
- \(\varepsilon_\omega(t)\) and \(\varepsilon_i(t)\) are uncorrelated random sequences with zero mean for speed and current GPC models.
- \(y_\omega\) is the speed controller output of GPC.
- \(y_i\) is the current controller output of GPC.
- \(A_\omega(q^{-1}) = 1 + a_{\omega 1}q^{-1} + a_{\omega 2}q^{-2} + \ldots + a_{\omega m}q^{-m}\).
- \(A_i(q^{-1}) = 1 + a_{i 1}q^{-1} + a_{i 2}q^{-2} + \ldots + a_{i m}q^{-m}\).
- \(B_\omega(q^{-1}) = b_{\omega 0} + b_{\omega 1}q^{-1} + b_{\omega 2}q^{-2} + \ldots + b_{\omega m}q^{-m}\).
- \(B_i(q^{-1}) = b_{i 0} + b_{i 1}q^{-1} + b_{i 2}q^{-2} + \ldots + b_{i m}q^{-m}\).

Since speed control PMBLDC motor is a SISO system, the predicted outputs \(\hat{y}(t + N_i)\) and \(\hat{y}(t + N_\omega)\) can be obtained by recursively iterating equation (4.2).

The predicted output can be expressed in vector notation as follows

\[
\hat{Y} = QU + P
\]  

(4.6)

where

- \(P\) is called free response = \([p(t + N_i), \ldots, p(t + N_\omega)]^T\).
- \(U\) is the manipulated variable vector = \([\Delta u(t), \ldots, u(t + N_u - 1)]^T\).
- \(\hat{Y}\) is the predicted output = \([\hat{y}(t + N_i), \ldots, \hat{y}(t + N_\omega)]^T\).
- \(Q\) is the step response matrix.
- \(N_u\) is the control horizon.
P is affected by past control action only. P (t+j) can be easily calculated for all j values by iterating PMBLDC motor model and a future control equals the previous control variable \( u(t-1) \). Considering multistage cost function \[49\]

\[
J_{GPC} = \sum_{j=N_1}^{N_2} \left[ y(t+j) - w(t+j) \right]^2 + \sum_{j=1}^{N_2} \lambda \Delta u^2(t+j-1) \tag{4.7}
\]

where

- \( w(t+j) \) is a future reference trajectory, which is a pre-specified set point \( y_r(t) \)
- \( \lambda \) is a control - weighting sequence.

For \( j=1 \) to \( N \), the future reference trajectory vector can be written as

\[
W=[w(t+1), w(t+2), \ldots \ldots \ldots \ldots \ldots w(t+N)]^T
\]

Thus the cost function can be written as

\[
J_{GPC} = (QU + P - W)^T (QU + P - W) + \lambda U^T U \tag{4.8}
\]

The solution minimising \( J_{GPC} \) gives an optimal value, which is suggested for control increment sequence. The optimum value for the prediction sequence will be

\[
U_{opt} = (Q^T Q + \lambda I_N)^{-1} Q^T + (w - P) \tag{4.9}
\]

From the above equations (4.7), (4.8) and (4.9), the cost function for the inner GPCi model and the external GPCω model are derived as
where

\[ N_{1\omega} \] is the minimum prediction horizon for speed loop
\[ N_{1i} \] is the minimum prediction horizon for current loop
\[ N_{2\omega} \] is the maximum prediction horizon for speed loop
\[ N_{2i} \] is the maximum prediction horizon for current loop
\[ N_{u\omega} \] is the control horizon for the speed loop
\[ N_{ui} \] is the control horizon for the current loop

As a standard generalised predictive control \([49, 50]\), Eqns. (4.10) and (4.11) can be written in the matrix form

\[
\begin{align*}
\hat{Y}_e &= Q_e U_e + P_e \\
\hat{Y}_i &= Q_i U_i + P_i
\end{align*}
\]

where

\[
\begin{align*}
\hat{Y}_e &= \begin{bmatrix} y_e(t + N_{1\omega}) & \cdots & y_e(t + N_{1\omega} + N_{2\omega} - 1) \end{bmatrix}^T \\
\hat{Y}_i &= \begin{bmatrix} y_i(t + N_{1i}) & \cdots & y_i(t + N_{1i} + N_{2i} - 1) \end{bmatrix}^T \\
U_e &= \begin{bmatrix} \Delta u_e(t) & \cdots & \Delta u_e(t + N_{uem} - 1) \end{bmatrix}^T \\
U_i &= \begin{bmatrix} \Delta u_i(t) & \cdots & \Delta u_i(t + N_{uin} - 1) \end{bmatrix}^T \\
P_e &= \begin{bmatrix} p_e(t + N_{1\omega}) & \cdots & p_e(t + N_{2\omega}) \end{bmatrix}^T \\
P_i &= \begin{bmatrix} p_i(t + N_{1i}) & \cdots & p_i(t + N_{2i}) \end{bmatrix}^T
\end{align*}
\]

The \(Q_e\) and \(Q_i\) are the step response matrices for the speed loop and current loop system respectively. The future reference trajectory \(w_e\) and \(w_i\) are expressed as.
Substituting Eqns. (4.12.a) and (4.12.b) in (4.11) and (4.10) respectively,

\[ w_a = [w_a(t + N_{1a}), \ldots, w_a(t + N_{2a})]^T \]
\[ w_i = [w_i(t + N_{1i}), \ldots, w_i(t + N_{2i})]^T \]

Substituting Eqns. (4.12.a) and (4.12.b) in (4.11) and (4.10) respectively,

\[ J_{GPC_a} = (Q_a U_a + P_a - w_a)^T (Q_a U_a + P_a - w_a) + \lambda_a U_a^T U \]  \hspace{1cm} (4.13)
\[ J_{GPC_i} = (Q_i U_i + P_i - w_i)^T (Q_i U_i + P_i - w_i) + \lambda_i U_i^T U \]  \hspace{1cm} (4.14)

The simulation diagram for the cascade dual GPC is shown in Fig. 4.3. It is noted from the figure that the control variable GPC\(_a\) acts as set point to the GPC\(_i\) and the speed loop tracks speed setpoint.

![Simulation diagram for cascade dual GPC](image)

**Fig.4.3. Simulation diagram for cascade dual GPC**

Through computing \( \partial J_{GPC_a} / \partial U_a = 0 \) and \( \partial J_{GPC_i} / \partial U_i = 0 \), the optimal control variables for the optimal speed control system are obtained as follows:

\[ U_{opt_a} = (Q_a^T Q_a + \lambda_a I N_{1a})^{-1} Q_a^T (w_a - P_a) \]  \hspace{1cm} (4.15)
\[ U_{opt_i} = (Q_i^T Q_i + \lambda_i I N_{1i})^{-1} Q_i^T (w_i - P_i) \]  \hspace{1cm} (4.16)
The proposed cascade dual GPC algorithm is implemented as follows

Step 1: Set a value for $k$ and set a sampling time for the speed and current loops as $T_\omega$ and $T_i$.

Step 2: Set a maximum prediction horizon, minimum predictive horizon and control horizon for the two loops.

Step 3: Estimate the CARIMA model to yield $Q_\omega$, $Q_i$ and $P_\omega$, $P_i$.

Step 4: Compute matrix $Q_\omega$, $Q_i$ and $(Q_\omega^T Q_\omega + \lambda_\omega IN_{1\omega})^{-1}$, $(Q_i^T Q_i + \lambda_i IN_{2i})^{-1}$.

Step 5: Determine the variables $U_{\text{open}}$ and $U_{\text{opti}}$ based on eqns. (4.15) and (4.16).

Step 6: Set the increment as $k=k+1$.

Step 7: Check the set value of $k$. If $k<\text{set value}$, go to Step 3, else end.

4.4 SIMULATION RESULTS AND DISCUSSION

A generalised predictive control algorithm has been proposed and its application to a PMBLDC motor has been investigated. The motor is characterised by fast dynamic non-minimum phase behaviour with nonlinearities. The time constant of the current loop is $\tau_i=20\text{ms}$, while that of the outer loop is almost $\tau_\omega = 230\text{ms}$. In order to show the interaction effects, multiple rates sample time should be considered [71]. The sampling time for speed control loop ($T_\omega$) is 8 ms and sampling time for current control loop ($T_i$) is 2 ms. The GPC parameters have been chosen to design controllers as given in Table 4.1.

<table>
<thead>
<tr>
<th>$N_{1\omega}$</th>
<th>$N_{2\omega}$</th>
<th>$N_{1i}$</th>
<th>$N_{2i}$</th>
<th>$N_{ui}$</th>
<th>$\lambda_\omega$</th>
<th>$\lambda_i$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>12</td>
<td>1</td>
<td>2</td>
<td>20</td>
<td>0.0000002</td>
<td>0.0000004</td>
</tr>
</tbody>
</table>

Table 4.1. GPC parameters of PMBLDC motor system
The GPC parameters are chosen for the following conditions.

\( N_u = 1 \) and \( N_m \) equal to 1 being \( N_u < N_2 \).

\( N_{1u} \) can often be taken as 1.

\( N_2 \) is set to approximate the rise-time of the motor.

The performance of cascade dual GPC is analysed with different types of inputs and their results are compared with conventional PI controller. When the set point equal to the step sequence, the control variables of current loop follow the speed loop without any oscillation and tracking their path smoothly with a settling time of approximately 40ms as shown in Figs. 4.4 and 4.5 for speed control loop and current control loop respectively. Fig. 4.6 depicts the output responses of speed and current for both conventional PI control and DGPC. From this figure, it is noted that the PI controller response is sluggish with delay in their speed current loops and it is compensated by the proposed DGPC. For the conventional controller, the speed and current are settled at different instants whereas, in the proposed method, the speed and current are settled almost at the same instant.

![Fig. 4.4. Response of the speed control variable for the given step reference](image)

68
Similarly, to examine the track performance along with change in operating condition, a square wave input is applied to the speed and current loop of the motor and their responses are shown in Figs. 4.7 and 4.8 respectively. It is observed that the speed and current loops are settled without any fluctuation. In this case the settling time is approximately equal to 40ms.
Figs. 4.7 and 4.8 depict the set point tracking performance for both speed and current loops for the given ramp input. From these figures it is observed that the conventional PI controller has a time lag and little overshoot during the transient
period. Whereas the proposed controller compensates the overshoot and delay without any tracking error.

Fig. 4.9. Output of the speed loop with increasing ramp reference

Fig. 4.10. Output of the current loop with ramp reference
Fig. 4.11 shows the high speed response of the motor for variable step set points with noise in the inner current loop. From the figure, it is clear that the conventional speed PI controller is sensitive to noises and its speed oscillates with maximum of 10% overshoot. Whereas the DGPC tackle the model uncertainty problem and rejects current loop disturbance. The DGPC achieves fast and non oscilator convergence of the motor output. The outer loop controller not only tackles model uncertainty problems, also the rejects the current loop disturbances.

Fig. 4.11. Comparison of cascade PI algorithm with DGPC algorithm for variable step

The impact of this controller is also simulated for the speed reference, speed error and torque signals responses when the motor is running slowly as shown in Fig. 4.12. The simulation shows that the controller provides a fast tracking behaviour, the speed error is very small, but the influence of the measurement noise on the control signal appears clearly.
Fig. 4.12. Speed response and control signal for the PI controller under slow speed

Figs. 4.13 and 4.14 depict the speed, speed error and control signal of the PMBLDC motor. For the DGPC controller, the speed error is smaller with nearly the same noise in the control signal. It is also noted that DGPC efficiently rejects the noise on control signal. In fact, the PI controller provides the same effect with respect to noise and disturbance rejection with a slower tracking dynamic.
Fig. 4.13. Speed response and control signal for the PI controller under nominal inertial condition.

Fig. 4.14. Speed response and control signal for the dual-GPC controller under nominal inertial load condition.
Figs. 4.15 and 4.16 show the results obtained under the same conditions for the dual GPC and the PID controllers. With the inertia $J=0.5J_n$, the speed tracking is quite similar for both cases, but the disturbance rejection appears to be better for the dual GPC controller. The control signal has no oscillation for the GPC controller and has oscillations for the PI controller. This oscillation in the control signal is induced by a neglected dynamic that appears in the system due to the additional inertia. This neglected dynamic affects the PI controlled system, which is not the case with the dual GPC controller.

Consequently, the GPC controller can accept higher additive uncertainties without loss of stability. The simulated results show that the proposed method allows robustness of the initial controller according to several constraints. It also guarantees some robustness performance when the inertia of the system varies and guarantees a desired dynamic for the disturbance rejection. It is concluded that cascade GPC makes full use of advance knowledge of future requirements to
achieve improved performance over the well tuned cascade PI controller. The performance comparison of cascade PI and DGPC are summarised in Table 4.2.

![Graph showing speed response and control signal for the dual-GPC controller under perturbation](image)

Fig. 4.16. Speed response and control signal for the dual-GPC controller under perturbation

<table>
<thead>
<tr>
<th>Table 4.2 Performance comparison of cascade PI and DGPC</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Type of controller</strong></td>
</tr>
<tr>
<td>------------------------</td>
</tr>
<tr>
<td>Error in rad/s</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Control signal variation in Nm</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Oscillation</td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>
4.5 SUMMARY

A cascade GPC for speed control of PMBLDC motor is proposed in this chapter. The inner loop uses GPC, exploiting information conveyed by accessible disturbances, while outer loop used a GPC to restrain the error from nonlinear identification of the generalised system based on PMBLDC motor models. Simulation results showed that cascade GPC outperformed than the well tuned cascade PI controller. Also, the results demonstrated the satisfactory system output and smooth feasible control actions of the dual GPC. The proposed dual GPC scheme successfully replaces the well tuned cascade PI control algorithm.