2016 6th IEEE International Conference on Control System, Computing and Engineering, 25–27 November 2016, Penang, Malaysia

$\mathcal{H}_\infty$ Tuning Technique for PMSM Cascade PI Control Structure

Lukas Pohl* and Ludek Buchta†
CEITEC - Central European Institute of Technology,
Brno University of Technology
Purkynova 123, Brno, Czech Republic
Email: *lukas.pohl@ceitec.vutbr.cz
†ludek.buchta@ceitec.vutbr.cz

***Abstract*—In this paper the nonsmooth optimization technique is proposed for parameter tuning of a set of PI controllers. PI controllers connected in a cascade structure are controlling the stator currents and the rotor speed of PMSM (permanent magnet synchronous motor). Performance criteria are specified with weighting functions for each PI controller. Important aspects of weight design and its impact on cascade control of PMSM are presented. Nonlinear nature of PMSM is resolved by sampling LPV (linear parameter varying) model at key values of the rotor speed and creating set of design points. Tuning of cascaded PI controllers is performed as a two step procedure. In the first step the inner current loop is tuned for all the design points. As a second step to the procedure, the inner loop PI controller parameters are locked and outer speed PI controller is tuned around the q-axis current loop.**

Keywords*—$\mathcal{H}_\infty$, *fixed structured synthesis, PMSM, real-time implementation

I. Introduction

Efficiency, high torque ratio, compact size and reliability are the main qualities of PMSM (permanent magnet synchronous motor). These are also the reasons for increasing popularity of permanent magnet motors. Although the permanent magnet motors with sinusoidal back-EMF are mostly referred to as PMSM, they are often denoted as BLDC (brushless DC) motors by some manufacturers. Construction of these types of drives is very similar and other than the shape of generated voltages there is little difference between them. This similarity also allows using sinusoidal based control on trapezoidal (BLDC) motor and vice versa.

This article focuses on field oriented control (FOC) which has been proven to work on both PMS and BLDC motors. Basic idea of FOC is to transform multi phase (often 3-phase) currents and voltages into d-q synchronous frame, and consequently, decoupling the torque and flux control. Significant control over the rotor speed, stator voltages and currents can be achieved by using the FOC in cascade control structure. This architecture consists of two loops, inner loop for the stator currents and outer loop for the rotor speed. Cascade control structure is especially useful for a set of plants where the faster part of the plant can be controlled independently. In this case the feedback loop is constructed around the stator currents with stator voltages acting as a control effort. Current set point for the inner loop is produced by the outer speed loop. This approach also allows for easy implementation of current and voltage restriction. Control of d, q currents and rotor speed requires total of three controllers. The most basic type of controller that can achieve zero steady state error is the PI controller. With three of these controllers in a cascade control structure there are total of six parameters needed to be tuned for precision current and speed control. The task of tuning interconnected set of PI controllers can be difficult, especially when there is coupling between individual control loops. This is also the case of d-q current control where the amount of current cross-coupling depends on value of the rotor speed.

There are many different approaches to PI parameters tuning, but this article will focus on methods based on Robust Control Theory and $\mathcal{H}_\infty$ Synthesis. Most of these methods are designed to "tune" generic state-space controller. Publications describing simulations and experiments on real PMSM shows that well designed generic state-space controller can even outperform PI controllers. Example of robust controller $\mathcal{H}_\infty$ design, that treats rotor speed as and uncertainty, can be found in papers [1][2]. Similar, but less conservative approach to $\mathcal{H}_\infty$ synthesis utilizes the rotor speed as a scheduling variable to construct LPV (linear parameter varying) controller. This approach results in robust self-adjusting controller that performs well in whole operating range of the motor. Simulation and experimental results of this method tested on either PMSM or ACIM (induction motor) can be found in papers [3][4]. Most of these $\mathcal{H}_\infty$ controllers are computed as a result of the Riccati equation or an inequality derived from the Riccati equation. This inequality is often in a form of linear matrix inequality (referred to as bounded real lemma) which is convex and it can be solved efficiently via semidefinite programming. Problem of finding suitable controller becomes non-convex when the controllers structure is fixed (e.g., PI structure). Computation of non-convex optimization problem is more difficult and the result can vary for different initial conditions.

The method presented in [5] is not based on bounded real lemma. The problem of finding $\mathcal{H}_\infty$ controller is divided into two steps. The first step is pure stabilization of the plant and the second step is the $\mathcal{H}_\infty$ controller computation using non-smooth technique for structure-constrained controllers. Examples with different applications of this can be found in

978-1-5090-1178-0/16 /$31.00 ©2016 IEEE

recently published articles [6][7]. Main goal of this article is to apply the method presented in [5] to a cascade interconnection of PI controllers controlling the d-q currents and the rotor speed.

II. Controlled Plant Setup

A. PMSM Model

Under the assumption that i_d is kept close to a zero, the simplified PMSM model can be described by the following set of differential equations:

$$\begin{aligned}\frac{d\,i_d}{dt} &= -\frac{R_s}{L_d}i_d + \frac{\omega_m n_p L_q}{L_d}i_q + \frac{1}{L_d}\,u_d \\ \frac{d\,i_q}{dt} &= -\frac{\omega_m n_p L_d}{L_q}i_d - \frac{R_s}{L_q}i_q + \frac{1}{L_q}(u_q - \omega_m n_p\,\Psi_f) \\ \frac{d\,\omega_m}{dt} &= \frac{3}{2J_m}\Psi_f i_q - \frac{1}{J_m}T_L - \frac{b}{J_m}\omega_m \end{aligned} \tag{1}$$

There are several issues that needs to be addressed before these equations can be used for a $\mathcal{H}_\infty$ current/speed controller tuning. Both current equations have cross-coupling elements that are multiplied by the rotor speed. The speed dependence can be easily resolved by sampling the model at key values of the rotor speed. The second equation $\left(\frac{d\,i_q}{dt}\right)$ contains element Ψ_f which also linearly depends on the rotor speed. Even though the speed variable is replaced by a set of constant values, the resulting constant element $-\bar{\omega}_m\Psi_f/L_q$ cannot be simply represented using linear state space matrices. The value of permanent magnet flux Ψ_f (often described as motor voltage constant) does not change during the motor operation, so usually there is no reason to describe it as uncertain. In this case treating the Ψ_f as an uncertainty serves a single purpose. As an uncertainty, affected by an exogenous input w, the whole constant element $-\bar{\omega}_m\Psi_f/L_q$ can be moved to the exogenous input matrix B_w. This operation can be also applied to a load torque T_L element in the third equation $\left(\frac{d\,\omega_m}{dt}\right)$.

The modifications of the nonlinear differential model results in the following state space representation:

$$\begin{aligned}\dot{x} &= A(\omega_m)x + B_w(\omega_m)w + B_u u \\ z &= C_z x + D_{w\,z}w + D_{u\,z}u \\ y &= C_y x + D_{w\,y}w + D_{u\,y}u \end{aligned} \tag{2}$$

With state variables $x = [i_d, i_q, \omega_m]^T$, control inputs $u = [u_d, u_q]^T$, exogenous inputs $w = [\Psi_f, T_L]$ and measured outputs $y = [i_d, i_q, \omega_m]^T$. Exogenous outputs z will be utilized after the final feedback interconnection has been augmented with performance weights.

$$A = \begin{bmatrix} -\frac{R_s}{L_d} & \omega_m\frac{L_q}{L_d} & 0 \\ -\omega_m\frac{L_d}{L_q} & -\frac{R_s}{L_q} & 0 \\ 0 & n_p\frac{3}{2}\frac{\Psi_f}{J_m} & -\frac{b}{J_m} \end{bmatrix} \quad B_w = \begin{bmatrix} 0 & 0 \\ \omega_m\frac{1}{L_q} & 0 \\ 0 & -\frac{1}{J_m} \end{bmatrix}$$
$$B_u = \begin{bmatrix} \frac{1}{L_d} & 0 \\ 0 & \frac{1}{L_q} \\ 0 & 0 \end{bmatrix} \qquad C_y = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3}$$

Matrices C_z, D_{wz}, D_{uz}, D_{wy} and D_{uy} are zero matrices of appropriate sizes.

B. Cascade Control Structure

Implementation of cascade control structure greatly improves the rejection of load torque disturbance [8]. Two different outputs of PMSM allows the separation of fast dynamics (current) from the slow dynamics (rotor speed). Figure 1 shows the cascade structure controlling the d-q currents and the rotor speed. PI_{id} and PI_{iq} controllers, as part of the inner current loop, are responsible for fast current tracking and for the rejection of noise in measured currents. PI_ω controller should provide the correct setpoint value of q axis current in order to track the speed setpoint under different load conditions. In contrast to direct speed control with just the stator voltages, the stator currents can be easily restricted by limiting the setpoint of q-axis current controller.

C. PI Tuning Structure

Cascade structure for the controller tuning is shown on Fig. 2. Main difference between the tuning diagram and the cascade control diagram is that the tuning diagram includes additional inputs and outputs. These inputs/outputs, often called exogenous, are minimized by the optimization method. Setpoint values of current and speed are for the purpose of controller synthesis also considered to be exogenous.

III. PI Tuning Procedure

A. Weight Selection

Speed and current feedback control loops are augmented with performance weights W_P. Purpose of the performance weights W_P is to give the sensitivity function desired shape.

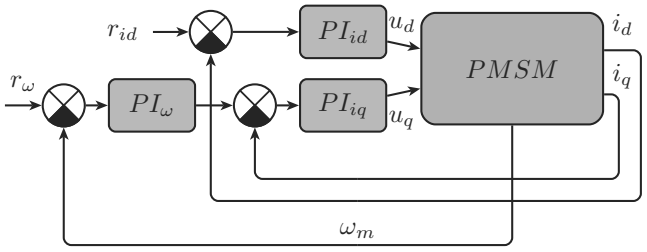

Fig. 1: Cascade Control Structure

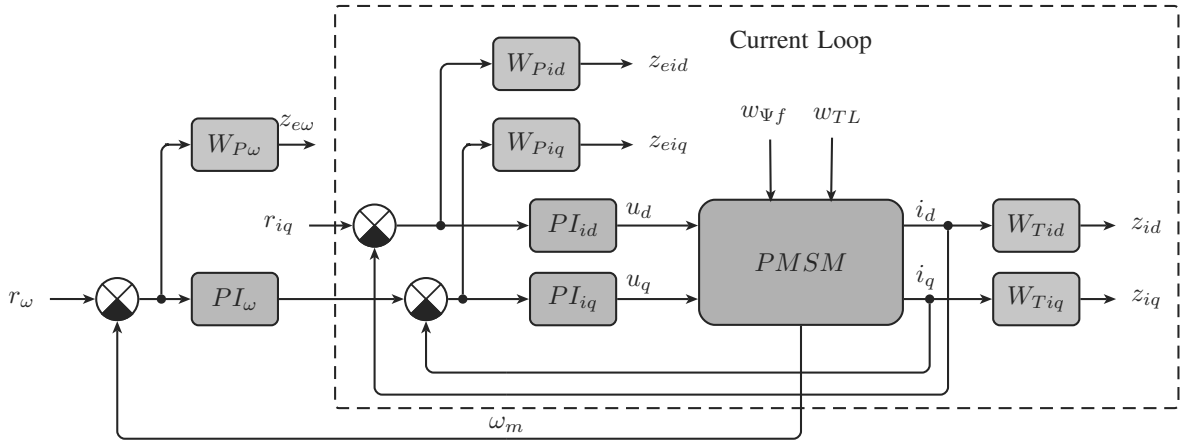

Fig. 2: Structure for PI tuning

Shape of the sensitivity function affects the disturbance rejection and overall control performance. Weight W_P was selected in following manner:

$$W_P = \begin{pmatrix} \frac{\frac{1}{M}s + \omega_S}{s + \omega_S A} & 0 \\ 0 & \frac{\frac{1}{M}s + \omega_S}{s + \omega_S A} \end{pmatrix} \tag{4}$$

Where M affects the robust stability, A the tracking error and ω_S the response time. For the PMSM with parameters listed in Table I the following values were selected for current PI optimization:

$$A = 0.0001 \qquad M = 2 \qquad \omega_S = 500$$

Parameters of the speed loop sensitivity weight were similar, only the value of ω_S was changed to 0.1.

The complementary sensitivity functions of both feedback loops are also shaped using the complementary sensitivity weights W_T. In theory, simple static weight (gain) would suffice for shaping, because the complementary sensitivity weight is primarily used to limit the setpoint overshot. In the case of cascade control structure it is often useful to slow down the outer loop so that the inner loop has faster dynamics by at least an order of magnitude. Following relation holds for the sensitivity and complementary sensitivity functions:

$$S + T = 1 \tag{5}$$

Where S is the sensitivity function and T is complementary sensitivity function. Slowing down the dynamics with sensitivity weight alone is not possible because it defines only the upper bound on frequency domain shape. The lower bound of sensitivity function is implicitly defined by the complementary sensitivity weight.

$$W_T = \begin{pmatrix} \frac{s + \omega_T}{A_T(\frac{1}{\epsilon}s + \omega_T)} & 0 \\ 0 & \frac{s + \omega_T}{A_T(\frac{1}{\epsilon}s + \omega_T)} \end{pmatrix} \tag{6}$$

The value of A_T limits the setpoint overshot, ϵ is the implementation constant of small value and ω_T defines the crossover frequency. The process of choosing correct crossover frequency ω_T is crucial for the speed feedback loop. Correct value will move the sensitivity function closer to its frequency domain upper bound, thus achieving the the desired value of ω_S. Following values were chosen for the current loop complementary sensitivity function weight:

$$A_T = 1.01 \qquad \epsilon = 0.000001 \qquad \omega_T = 500$$

The value of ω_T for the optimization of speed PI controller was chosen as 1000 in order to push the sensitivity function to lower crossover frequency.

B. Non-smooth Optimization

Tuning procedure of cascaded PI controllers for PMSM was carried out in following steps:

I) d and q axis current PI tuning
II) speed loop PI tuning with fixed current PI controllers

Tuning of current PI controllers was carried out using PMSM state space model with rotor speed defined as tunable object. The state space model was sampled at several key points of motor speed range, thus locking the tunable object to a fixed value. The sampled model substitutes the LPV system because the optimization methods for non-smooth synthesis are not suited for LPV state space models. Structure for the current PI tuning is shown on Fig. 2. Current controllers were defined as tunable LTI matlab objects. The feedback

interconnection of both current controllers and sampled PMSM model was augmented with sensitivity and complementary sensitivity weights. Resulting controllers were computed using `hinfstruct` matlab function for non-smooth $\mathcal{H}_\infty$ synthesis.

IV. Real-time Experiment

A. Control Performance Verification

Performance goals defined by W_P and W_T weights can be easily verified by plotting the singular values of frequency response. Frequency responses of S and T together with $1/W_P$ and $1/W_T$ were obtained for both current loop and current/speed cascade loop. Singular values of the current loop frequency response is shown on Fig. 3. Singular values of the speed/current cascade loop frequency response is shown on Fig. 4. S and T plots of both loops are close to their respective bounds which is also confirmed by γ having value close to 1. It is also important to verify that inner loop is approximately ten times faster than outer loop when using the cascade control structure.

B. Experiment Setup

Real-time experiments were performed on dSPACE DS1103 real-time platform that was connected to a 30V, 5A power stage. Rotor speed was measured using incremental encoder connected to the rotor shaft. Phase currents were sampled by shunt resistors located on the low side of the bridge. Precise timing of the current measurement is achieved with interrupt from the center aligned pulse width modulation (PWM). Control algorithm was implemented in Matlab Simulink and uploaded to the dSPACE platform through automated code generation and compilation. Simulation diagram is shown on Fig. 5.

C. Platform Considerations for Implementation

Mechanical parameters of the PMS motor and the platform itself allows fast current and speed tracking. When the experiment reaches the platform limits, several issues can arise. For example the harmonic distortion that can be observed on stator currents might influence the current control performance. This distortion is periodical and it is correlated to the phase current zero crossing. Source of this issue can be traced to the power stage. Voltage for each phase is supplied by a pair of unipolar transistors. Each pair is supposed to work in complementary mode, switching its state twice the PWM cycle. During the transition between on/off state it necessary to provide sufficient time between switching high side and low side transistors to prevent a shoot through. Time needed to safely transition between on/off state is often called Dead-Time. Duty cycles are modified by extending the pulse for one side and shortening the other to guarantee overlap of high side and low side signals. When small values of voltage are requested by the controller, the lower bound on the duty cycle is limited by this overlap. In order to achieve undistorted phase currents it is necessary to compensate the stator voltages every time the stator currents are crossing the zero value. The dead-time compensation is included in the cascade control structure shown on Fig. 5.

Sampling frequency of the dSPACE ds1103 was chosen to be 16kHz. Higher frequencies lead to higher switching loses of the power stage, requiring better heat dissipation. Sampling frequency of 16kHz is sufficient for the current control of PMSM, but it is recommended to check the size of largest singular value of S or T. Sampling frequency should be at least twice the size of largest singular value (sampling theorem) [9][10].

During the experiments it was also found out that fast dynamic changes of rotor speed can exceed dynamic range of the incremental encoder. Length of the speed step response was in some cases just few samples for lower speed sampling frequencies. Larger amount of samples can be achieved by faster sampling but it also causes amplification of measured speed noise. It is good practice to chose the speed sampling frequency as tradeoff between measurement speed and noise.

V. Results

Optimization of both current PI controllers and the speed PI controller resulted in following set of proportional and integral gains:

$$\begin{aligned} PI_{id} &: K_p = 0.771 & K_i &= 1000 \\ PI_{iq} &: K_p = 1.08 & K_i &= 1000 \\ PI_{\omega} &: K_p = 0.57 & K_i &= 0.058 \end{aligned}$$

Performance of the cascade control structure with PI controllers tuned by $\mathcal{H}_\infty$ optimization method was tested with speed step change from 0rad/s to 100rad/s. Two other PI tuning methods were chosen for performance comparison. The first method is based on simple tuning procedure described in book: Internal model control: PID controller design [11]. Integration time constant was chosen to cancel out time constant of the controlled system. Proportional gain is increased until the phase margin is reduced to 45 degrees (same as $\mathcal{H}_\infty$ controller with $M = 2$). Since the controlled system is a first order plant, instability is possible only when the time delay is taken into account. This assumption conforms to real control system, where small amount of delay is always present. The second method used for comparison is commonly known as Ziegler-Nichols method. PI constants are chosen from predefined table based on experimental measurements. Details about the design method can be found in book: Advanced PID control [12]. All the PI current controllers were tested under the same conditions with the same $\mathcal{H}_\infty$ speed controller. The current setpoint limit was set to 5A. Figures 6 and 7 shows significant improvement over the classic PI tuning methods. Both classic methods exhibit setpoint overshot, but the $\mathcal{H}_\infty$ tuning technique is able to achieve the desired current without an overshot. This result was expected, since the $\mathcal{H}_\infty$ method is based on mixed sensitivity design. Mixed sensitivity design takes into account not only the stability margin, but also performance as design criteria.

VI. Conclusion

This paper presented PI controller tuning method applied to the PMSM cascade control structure. Two step optimization procedure was demonstrated on current and speed equations of PMSM. Desired performance, specified by two different sets

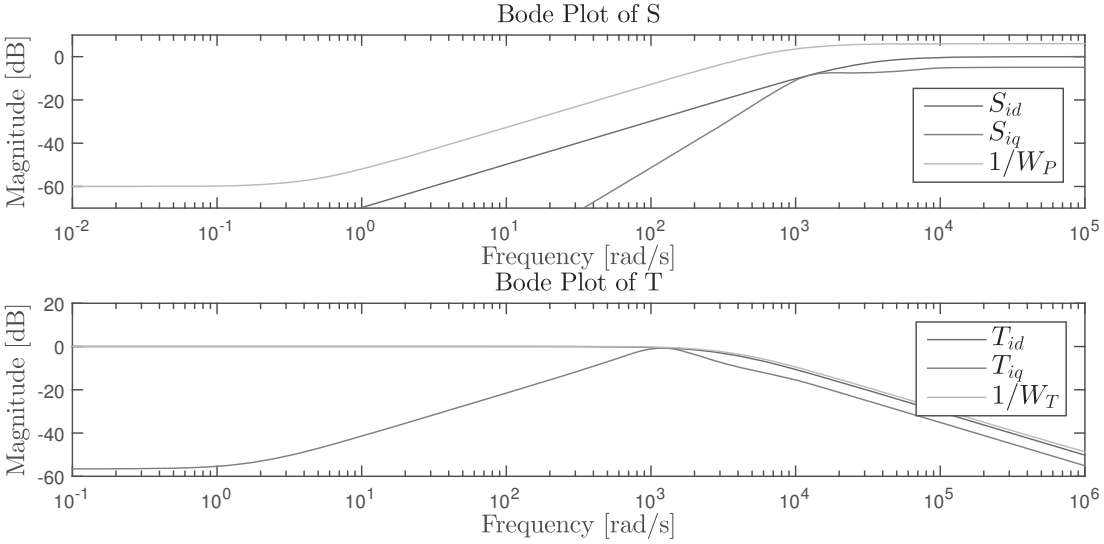

Fig. 3: Bode Plot of Sensitivity and Complementary Sensitivity Functions of Current Feedback Loop

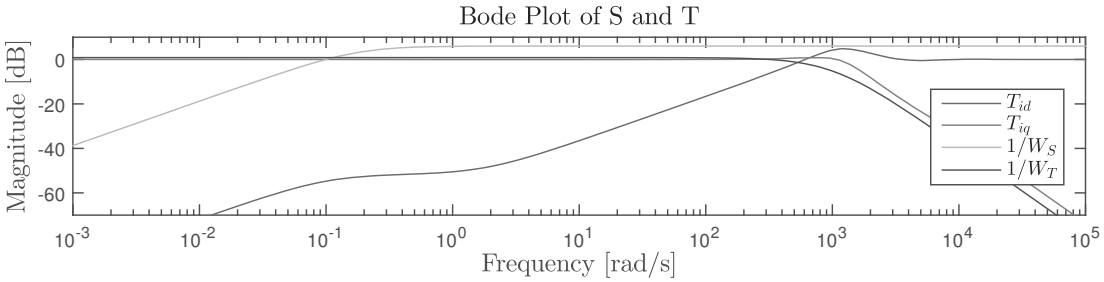

Fig. 4: Bode Plot of Sensitivity and Complementary Sensitivity Functions of Current Feedback Loop

of weighting functions for each loop, was achieved by locking current loop PI controllers for the second stage of optimization. The second optimization step tuned the speed PI controller on optimized current feedback loop, finalizing the cascade control structure.

Performance of the resulting cascade feedback loop was verified using singular values frequency plot. Frequency plot confirmed that the outer speed loop can be successfully slowed down by lowering the crossover frequency of sensitivity function weight. The optimization method was also forced to bring the sensitivity plot closer to its bound by constricting the complementary sensitivity function.

One of the main advantages of presented $\mathcal{H}_\infty$ optimization method is that the resulting controller has traditional PI(D) structure which has proven to be easily deployable on most digital controllers. Compared to the unstructured state space controller the PID control might not be suited for more complex controlled plats but it is more than sufficient for cascade control of PMSM. Real time testing proved that $\mathcal{H}_\infty$ optimization method has better performance than commonly used PI tuning methods. The future work will be focused on tuning different variants of PID controllers and improving the disturbance rejection of outer speed loop.

ACKNOWLEDGMENT

This research has been supported by the project CIDAM - Center for Intelligent Drives and Advanced Machine Control TE02000103 funded by the Technology Agency of the Czech Republic.

The research results were verified in simulation using AVL CRUISE M simulation SW provided by AVL within University Partnership Program.

REFERENCES

[1] J. Doyle, K. Glover, P. Khargonekar, and B. Francis, "State-space solutions to standard h2 and h infin; control problems," *Automatic Control, IEEE Transactions on*, vol. 34, no. 8, pp. 831–847, 1989.

[2] K. Zheng, A.-H. Lee, J. Bentsman, and P. T. Krein, "High performance robust linear controller synthesis for an induction motor using a multi-objective hybrid control strategy," *Nonlinear Analysis: Theory, Methods & Applications*, vol. 65, no. 11, pp. 2061–2081, 2006.

[3] S. Machmoum, P. Chevrel, and C. Darengosse, "A linear parameter variant Hinf controller design for a permanent magnet synchronous machine," *Power Electronics and Applications, 2005 ...*, pp. 1–11, 2005.

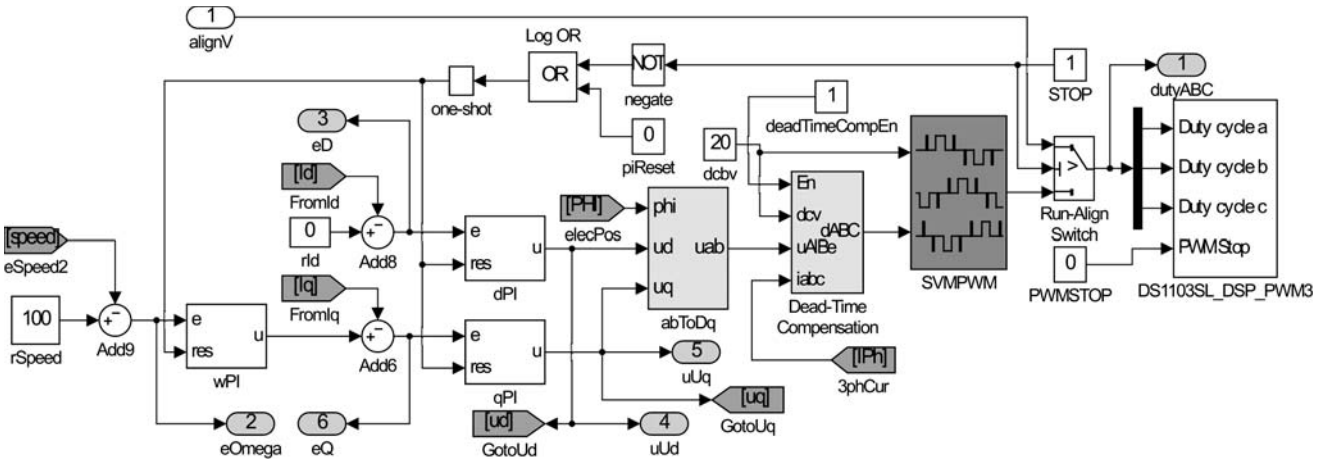

Fig. 5: Simulink Diagram of Cascade PMSM Control

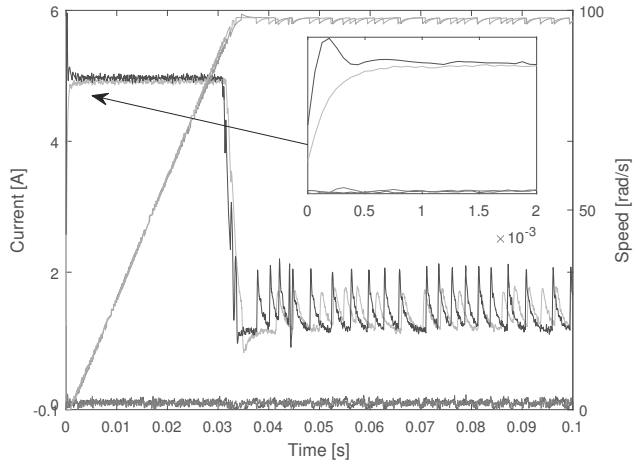

Fig. 6: Speed step response comparison between $\mathcal{H}_\infty$ tuning method and method from [11]. (— $i_{d\mathcal{H}\infty}$, — $i_{d\,classic1}$, — $i_{q\mathcal{H}\infty}$, — $i_{q\,classic1}$, — $\omega_{s\mathcal{H}\infty}$, — $\omega_{s\,classic1}$)

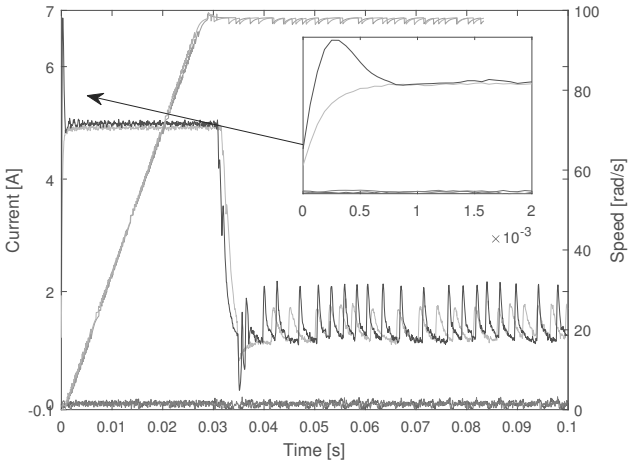

Fig. 7: Speed step response comparison between $\mathcal{H}_\infty$ tuning method and method from [12]. (— $i_{d\mathcal{H}\infty}$, — $i_{d\,classic2}$, — $i_{q\mathcal{H}\infty}$, — $i_{q\,classic2}$, — $\omega_{s\mathcal{H}\infty}$, — $\omega_{s\,classic2}$)

TABLE I: PMSM Nominal Parameters

Parameter Description	Parameter Symbol	Nominal Value
Number of pole pairs	p	3
Permanent magnet flux	Φ_f	0.0095 N.m/t
Stator resistance	R_s	0.35Ω
Direct axis inductance	L_d	240 μH
Quadratic axis inductance	L_q	320 μH
Moment of inertia	J_m	0.00163 kg.m^2
Viscous friction	b	0.00218N.m.s

[4] E. Prempain, I. Postlethwaite, and A. Benchaib, "A linear parameter variant Hinf control design for an induction motor," *Control Engineering Practice*, 2002.

[5] P. Apkarian and D. Noll, "Nonsmooth h synthesis," *Automatic Control, IEEE Transactions on*, vol. 51, no. 1, pp. 71–86, 2006.

[6] M. Abouzlam, R. Ouvrard, D. Mehdi, F. Pontlevoy, B. Gombert, N. K. V. Leitner, and S. Boukari, "A control for optimizing the advanced oxidation processescase of a catalytic ozonation reactor," *Control Engineering Practice*, vol. 44, pp. 1 – 9, 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0967066115001343

[7] G. Strub, S. Dobre, V. Gassmann, S. Theodoulis, and M. Basset, "Pitch-axis identification for a guided projectile using a wind-tunnel-based experimental setup," *IEEE/ASME Transactions on Mechatronics*, vol. 21, no. 3, pp. 1357–1365, June 2016.

[8] A. Visioli, *Practical PID Control*, 1st ed., ser. Advances in industrial control. Springer, 2006.

[9] H. Hanselmann, "Implementation of digital controllersa survey," *Automatica*, vol. 23, no. 1, pp. 7 – 32, 1987.

[10] P. Apkarian, "On the discretization of LMI-synthesized linear parameter-varying controllers," *Automatica*, 1997.

[11] D. E. Rivera, M. Morari, and S. Skogestad, "Internal model control: Pid controller design," *Industrial & engineering chemistry process design and development*, vol. 25, no. 1, pp. 252–265, 1986.

[12] K. J. Åström and T. Hägglund, *Advanced PID control*. ISA-The Instrumentation, Systems and Automation Society, 2006.