Discrete-Time LPV Current Control of an Induction Motor

Bendtsen, Jan Dimon; Trangbæk, Klaus

Publication date:
2001

Document Version
Også kaldet Forlagets PDF

Link to publication from Aalborg University

Citation for published version (APA):
Abstract—In this paper we apply a new method for gain-scheduled output feedback control of nonlinear systems to current control of an induction motor. The method relies on recently developed controller synthesis results for linear parameter-varying (LPV) systems, where the controller synthesis is formulated as a set of linear matrix inequalities with full-block multipliers. A standard nonlinear model of the motor is then constructed and written on LPV form. We then show that, although originally developed in continuous time, the controller synthesis results can be applied to a discrete-time model as well without further complications. The synthesis method is applied to the model, yielding an LPV discrete-time controller. Finally, the efficiency of the control scheme is validated via simulations as well as experimentally on the actual induction motor, both in open-loop current control and when an outer speed control loop is closed around the current loop.

Index Terms—Induction Motors, LPV Control, Linear Fractional Transformations, Linear Matrix Inequalities

I. INTRODUCTION

Gain scheduling is a well-known and common approach to control of well-behaved nonlinear systems. The classical approach to gain-scheduling control has been to linearise the plant model in some set of operating points and design one or more linear, possibly robust, controllers for the system in said points. The gains of these individual linear controllers are then interpolated between the different operating points. This approach has been used in a multitude of applications and often works well as long as the scheduling variable, i.e., the variable according to which the controllers are interpolated, varies slowly. However, as pointed out in [17], the rate of change of the parameter variation imposes fundamental limitations on the achievable performance of classical gain scheduling controllers. Also, the classical gain scheduling methods are generally somewhat ad hoc.

More recent work on linear parameter varying (LPV) control has addressed these issues by devising rigorous methods in which it is possible to compensate for known, fast parameter variations directly in the control design [7], [12], [14], [16]. Linear parameter-varying systems are linear systems whose system matrices depend on some time-varying parameter vector that is either fully known or at least known to be contained in some known set. In LPV control design this knowledge is employed to provide systematic gain scheduling in which stability and performance of the closed loop can be guaranteed. The controller synthesis is cast as a set of matrix inequalities based on the varying system matrices and the plant-controller interconnection, along with a set of multipliers, which must satisfy these matrix inequalities.

One problem with these types of approaches has so far been that it can be difficult to obtain non-conservative controllers for a given plant if the plant parameter variations are considerable and restrictions are placed on the controller synthesis in the form of pre-imposed structures in the aforementioned multipliers. In [2] a controller synthesis was achieved for parameter dependencies entering the system via a linear fractional transformation (LFT) description. The structural restrictions on the multipliers were dealt with in case of affine parameter dependencies in [3], but it is only recently that it has been shown how they can be lifted in case of more general, rational parameter dependencies, i.e., in LFT descriptions, as well. The resulting synthesis matrix inequalities yielding the controllers can be solved by using the so-called full-block S-procedure [15], [16]. In essence, this results in an automated controller design method for nonlinear systems which permit an LPV description. To the best of our knowledge, this paper presents the first actual implementation of an LPV controller designed via the full-block S-procedure.

In this paper we will use this novel technique to design and test a rotor flux oriented current controller for an induction motor. Induction motors have been used in a wide range of industrial as well as everyday applications over a number of decades, but with their highly nonlinear, fast dynamics they remain a challenge to control. A number of different current control schemes for three-phase systems have been employed in order to deal with the problem, such as classical linear controllers, predictive control and schemes based on neural networks/fuzzy logic (see [10] and the references therein). Several continuous-time control schemes that take the induction motor dynamics into account have been applied, including simple Lyapunov-based approaches [19], minimum-time control [6], sliding mode control [18], as well as decoupling with special attention paid to robustness [8]. Recently, backstepping techniques have been applied to current and rotational speed control simultaneously [13].

A drawback of most of these methods is, however, that they require a considerable amount of tuning and engineering insight. In this paper we will demonstrate that the LPV controller synthesis can be applied to the problem and achieve satisfying performance basically without any ad hoc tuning. Another general problem with these schemes is that it is unclear whether or not they will work well when implemented in discrete time at a sampling frequency which is not considerably faster than the motor dy-
The three-phase voltages and currents are transformed from into a single complex voltage and current representation in a rotating reference frame, respectively, according to the relations
\[ u_s = u_{sd} + j u_{sq} = \frac{2}{3} \left( u_{sA} + u_{sB} e^{j \frac{2\pi}{3}} + u_{sC} e^{j \frac{4\pi}{3}} \right) e^{-j \rho} \]
\[ i_s = i_{sd} + j i_{sq} = \frac{2}{3} \left( i_{sA} + i_{sB} e^{j \frac{2\pi}{3}} + i_{sC} e^{j \frac{4\pi}{3}} \right) e^{-j \rho} \]
where \( \rho \) is the angular position of the chosen reference frame. \( u_{sd} = \Re \{ u_s \} \), \( u_{sq} = \Im \{ u_s \} \), \( i_{sd} = \Re \{ i_s \} \), and \( i_{sq} = \Im \{ i_s \} \) are all real-valued signals. The aim will pursue in this paper is to design an inner current control loop which can be placed in a cascade coupling with an outer shaft speed control loop, as indicated in Figure 1.

![Schematic drawing of the induction motor setup.](image)

With standard assumptions, the motor model is given by (see e.g., [11] or [13]):
\[
\frac{d i_s}{dt} = -\left( \frac{R_s + R_r}{L_s} + j \omega \right) i_s + \left( \frac{R_s}{L_s} - j \frac{L'_m}{L_s} Z_p \omega_{\text{mech}} \right) i_m + \frac{u_s}{L_s} \tag{1}
\]
\[
\frac{d i_m}{dt} = \frac{R'_r}{L_m} i_s - \left( \frac{R'_r}{L_m} + j (\omega - Z_p \omega_{\text{mech}}) \right) i_m \tag{2}
\]
in which \( u_s \) and \( i_s \) are the complex stator voltage and current, respectively, and \( i_m \) is the complex magnetising current associated with the rotor flux \( \Psi_r = L_m i_m \). The equations above are given in a reference frame which rotates with the rotational speed \( \omega = \omega_r \). \( L_m \) is the magnetising inductance. \( R'_r = (L_m/L_r)^2 R_r \), \( L'_s = L_s - L_m^2/L_r \) and \( L'_m = L_m^2/L_r \) are the referred parameters used in the model, found based on identified values of the stator and rotor resistances and inductances \( R_s, R_r, L_s \) and \( L_r \). \( Z_p \) is the number of pole pairs, while \( \omega_{\text{mech}} \) is the motor shaft speed. The motor develops the following electromagnetic torque:
\[
m_e = \Re \left\{ \frac{3}{2} Z_p L_m i_s i_m^* \right\}
\]
while the load torque \( m_L \) acts as a disturbance via the mechanical relation
\[
\frac{d \omega_{\text{mech}}}{dt} = m_e - m_L
\]
where $J$ is the mechanical moment of inertia. We choose a reference frame rotating at the same angle as the magnetising current, since in this frame the steady-state signals are constant. Since, in reality, the magnetising current cannot be measured, we will use the following simple estimator. Let $T_r = L'_m/R_p$ and $\omega_r = Z_p\omega_{\text{mech}}$ and compute the estimate of $i_m$, $\hat{i}_m$, based on (2) as

$$\omega = \omega_r + i_{sq}/(T_{rd})$$  \hspace{1cm} (3)

$$\frac{d\hat{i}_m}{dt} = \frac{1}{T_r} \left( i - (1 + j(\omega - \omega_r))\hat{i}_m \right).$$  \hspace{1cm} (4)

In this reference frame, $\hat{i}_m$ is real. We choose the complex state vector $x = [i_m^* \ i_m]^*$ and insert (3) in (1)–(2) obtaining

$$\dot{x} = (A_0 + \delta_1 A_1 + \delta_2 A_2) x + Bu_s, \quad i = Cx$$  \hspace{1cm} (5)

in which

$$A_0 = \begin{bmatrix} -\frac{R_i + R'_i}{L_m} & \frac{R'_i}{L_m} \\ -\frac{R'_i}{L_m} & -\frac{1}{L_m} \end{bmatrix}, \quad B = \begin{bmatrix} \frac{1}{L_m} \\ 0 \end{bmatrix} \quad \text{and} \quad C = [1 \ 0]$$

represent the nominal model, which is a linear time invariant system, and

$$A_1 = \begin{bmatrix} -j & -j \frac{1}{T_r} \\ 0 & 0 \end{bmatrix} \quad \text{and} \quad A_2 = \begin{bmatrix} 0 & j \frac{1}{T_r} \\ 0 & -j \frac{1}{T_r} \end{bmatrix}$$

represent the effects of parameter variations in the linear system. These parameter variations symbolise the nonlinearities caused by $\delta_1 = \omega$ and $\delta_2 = i_{sq}/i_{md}$, where all signals vary with time and $i_{md} > 0 \forall t$. This particular choice of parameterisation has the advantage over the other obvious choice, $\delta_1 = \omega$ and $\delta_2 = \omega_r$, that $\omega$ will typically be close to $\omega_r$, the parameterisation chosen above is a straightforward way to exploit this knowledge. The system (5) can then be written on an LFT form and can be meaningfully discretised, for instance according to the method presented in [1]. By considering the parameter variations as being caused by external effects, we are able to employ the LPV control synthesis that will be described in the following section. It should be noted, as already mentioned in the introduction, that the main reason why we discretise the model at this point is to be able to address limitations on the sample rate already in the synthesis phase, before the actual implementation.

III. LPV Controller Synthesis

In the synthesis, we consider the discrete-time system

$$\begin{bmatrix} x_{k+1} \\ z_{u,k} \\ z_{p,k} \\ y_k \end{bmatrix} = \begin{bmatrix} A & B_u & B_p & B \\ C & D_{uu} & D_{up} & E_u \\ C_p & D_{pu} & D_{pp} & E_p \\ C & F_u & F_p & 0 \end{bmatrix} \begin{bmatrix} x_k \\ u_{u,k} \\ u_{p,k} \\ u_k \end{bmatrix}$$  \hspace{1cm} (6)

with $x_k \in \mathbb{C}^n$, $u_k \in \mathbb{C}^m$ and $y_k \in \mathbb{C}^q$ representing states, inputs and outputs at sample instant $k$, respectively. All the matrices are assumed to be complex, constant and of appropriate dimensions. $w_{u,k} \in \mathbb{C}^{nu}$ and $z_{u,k} \in \mathbb{C}^{nu}$ are used to specify performance and $w_{u,k} \in \mathbb{C}^{nu}$ and $z_{u,k} \in \mathbb{C}^{nu}$ are channels which connect a set of residual gains collected in the mapping $\Delta_k$ with the nominal linear system as follows:

$$w_{u,k} = \Delta_k z_{u,k}.$$  \hspace{1cm} (7)

$\Delta_k$ is a time-varying mapping that represents the nonlinearities in the system. We will assume that $\Delta \in \Delta$, where $\Delta$ is a compact, path-connected set containing $0$, and that the interconnection between the nominal system model $M(k)$ and $\Delta$ is well-posed, that is, $I - \Delta D_{uu}$ is nonsingular for all $\Delta \in \Delta$.

![Fig. 2. The interconnection of the nominal system $M(k)$, the residual gains $\Delta$, and the controller $K(k)$.

We then consider the controller-system interconnection depicted in Figure 2. The controller is of the form

$$\begin{bmatrix} x_{c,k+1} \\ u_k \\ z_{c,k} \\ y_k \end{bmatrix} = \begin{bmatrix} A_c & B_{c1} & B_{c2} \\ C_c & D_{c11} & D_{c12} \\ C_p & D_{c21} & D_{c22} \end{bmatrix} \begin{bmatrix} x_{c,k} \\ u_k \\ z_{c,k} \end{bmatrix}$$  \hspace{1cm} (8)

with $w_{u,k} = \Delta_c(\Delta_k) z_{c,k}$ where $\Delta_c$ is a nonlinear function of $\Delta$. If we interconnect the controller and the nominal system as depicted in the left part of Figure 2 we get the closed-loop LTI system $M_c(k)$ described by

$$\begin{bmatrix} \chi_{k+1} \\ z_{u,k} \\ z_{c,k} \\ z_{p,k} \end{bmatrix} = \begin{bmatrix} A & B_u & B_p & B \\ C & D_{uu} & D_{up} & E_u \\ C_p & D_{pu} & D_{pp} & E_p \\ C & F_u & F_p & 0 \end{bmatrix} \begin{bmatrix} \chi_k \\ w_{u,k} \\ w_{c,k} \\ w_{p,k} \end{bmatrix}$$  \hspace{1cm} (9)

subject to the parameter dependency

$$\begin{bmatrix} w_{u} \\ w_{c} \end{bmatrix} = \begin{bmatrix} \Delta_k & 0 \\ 0 & \Delta_c(\Delta_k) \end{bmatrix} \begin{bmatrix} x_{u} \\ z_{c} \end{bmatrix}$$  \hspace{1cm} (10)

and with the state vector $\chi_k = [x_k^* \ x_{c,k}^*]^*$. $\Delta_c$ and the controller matrices must be chosen such that the interconnection with the system and controller is well-posed, i.e., $I - [\Delta \ 0 \ \Delta_c] [D_{u} \ D_{c}]$ is nonsingular for all $\Delta \in \Delta$. More
explicitly, the gains of $M_c(k)$ in (9) are given by

$$M_c = \begin{bmatrix} A & B_u & B_c & B_p \\ C_u & D_{au} & D_{uc} & D_{up} \\ C_c & D_{cu} & D_{cc} & D_{cp} \\ C_p & D_{pu} & D_{pc} & D_{pp} \end{bmatrix}$$

$$= \begin{bmatrix} 0 & B_u & 0 & B_p \\ C_u & 0 & 0 & 0 \\ C_c & 0 & 0 & 0 \\ C_p & 0 & 0 & 0 \end{bmatrix} K \begin{bmatrix} 0 & I & 0 & 0 \\ C & 0 & F_u & 0 \\ 0 & 0 & I & 0 \\ 0 & F_p & 0 & 0 \end{bmatrix}$$

where $K$ is the matrix of controller gains given by (8). It can be shown that the trajectories of (9) are identical to those of the nonlinear system [15]

$$\chi_{k+1} = \bar{A}(\Delta_k)\chi_k + \bar{B}(\Delta_k)w_{p,k}$$

$$z_{p,k} = \bar{C}(\Delta_k)\chi_k + \bar{D}(\Delta_k)w_{p,k}$$

(11)

where

$$\begin{bmatrix} \bar{A}(\Delta_k) & \bar{B}(\Delta_k) \\ \bar{C}(\Delta_k) & \bar{D}(\Delta_k) \end{bmatrix} = \begin{bmatrix} A_m & B_{m2} \\ C_{m2} & D_{m22} \end{bmatrix} + \begin{bmatrix} B_{m1} \\ D_{m21} \end{bmatrix}\begin{bmatrix} D_{au} & D_{uc} \\ D_{cu} & D_{cc} \end{bmatrix}(I - [\Delta_k \Delta_k] \begin{bmatrix} D_{au} & D_{uc} \\ D_{cu} & D_{cc} \end{bmatrix})^{-1}\begin{bmatrix} C_m & D_{m12} \end{bmatrix}$$

The objective is, if possible, to find a gain-scheduled control law $K(k)$ and a scheduling function $\Delta(\Delta)$ such that the closed loop system (9) fulfills a robust quadratic performance specification (RQP), which is defined as follows.

- The interconnection of system and controller is well-defined.
- The unforced system is uniformly asymptotically stable, i.e., positive constants $K$ and $\alpha$ exist such that $\|\chi_k\| \leq ||\chi_0||K e^{-\alpha k}$ for $k \geq 0$ and all $\Delta \in \Delta$ if $w_{p,k} \equiv 0$.
- The following performance specification holds for $\chi_0 = 0$:

$$\exists \varepsilon > 0 : \limsup_{k \to \infty} \sum_{k=0}^{\infty} w_{p,k}^T P_p w_{p,k} \leq \varepsilon \sum_{k=0}^{\infty} w_{p,k}^T w_{p,k}$$

(12)

for some $P_p = \begin{bmatrix} Q_p & S_p \\ S_p^T & R_p \end{bmatrix}$, $R_p \geq 0$, specified a priori.

As can be seen, this formulation is equivalent to the continuous-time formulation of the notion of RQP (see for instance [16]). The following result shows that the discrete-time version of the full-block S-procedure yields a synthesis procedure that will guarantee (discrete-time) RQP for (9).

**Theorem 1:** Robust quadratic performance is achieved for the system (9)-(10) if one of the following two equivalent properties holds.

1. (9)-(10) is well-posed and there exists a Hermitian $\Delta > 0$ such that

$$\begin{bmatrix} \bar{A}(\Delta) & \bar{B}(\Delta) \\ \bar{C}(\Delta) & \bar{D}(\Delta) \end{bmatrix} < 0$$

(13)

for all $\Delta \in \Delta$.
2. There exists a Hermitian multiplier

$$P_e = \begin{bmatrix} Q & S \\ S^* & R \end{bmatrix}$$

(14)

which fulfills the matrix inequality

$$\begin{bmatrix} \Delta & 0 \\ 0 & \Delta_c(\Delta) \end{bmatrix}^* P_e \begin{bmatrix} \Delta & 0 \\ 0 & \Delta_c(\Delta) \end{bmatrix} > 0$$

(15)

for all $\Delta \in \Delta$ and a Lyapunov matrix $\Delta > 0$ such that

$$\begin{bmatrix} -\Delta & 0 \\ 0 & \Delta \end{bmatrix}^* P_e \begin{bmatrix} \Delta & 0 \\ 0 & \Delta_c(\Delta) \end{bmatrix} > 0$$

(16)

where

$$\tau = \begin{bmatrix} I & 0 & 0 \\ A & B_u & B_c \\ 0 & I & 0 \\ C_u & D_{au} & D_{uc} \\ 0 & 0 & I \\ C_c & D_{cu} & D_{cc} \\ 0 & 0 & 0 \\ C_p & D_{pu} & D_{pc} \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

(17)

**Proof:** Inspection reveals that the only difference between the continuous-time case and the discrete-time case is the upper left block in the central factors in (16) and (13). The equivalence between 1. and 2. hence follows from a direct application of the full-block S-procedure, Theorem 8 in [15].

We thus just need to show that requirement 1. yields RQP. Let $w_{p,k} \equiv 0$ in (11) and choose $V_k = \chi_k^T \chi_k$ as a Lyapunov candidate for the unforced system. The difference from sample to sample is $V_{k+1} - V_k = \chi_k^T \bar{A}(\Delta) \chi_k - \chi_k^T \chi_k$, which implies that the system is uniformly exponentially stable if $\bar{A}(\Delta)^T \chi_k^T \bar{A}(\Delta) < \chi_k^T \chi_k$. This is immediately deduced from the upper left block in (13), which can be written as $\bar{A}(\Delta)^T \chi_k^T \bar{A}(\Delta) - \chi_k^T \chi_k < 0$. Since $R_p \geq 0$ it is seen that if
\( X \) renders (13) satisfied, the unforced system is uniformly exponentially stable.

Furthermore, due to continuity and strictness of (13), we can add a small perturbation \( \mathcal{G} = \begin{bmatrix} 0 & \varepsilon I \end{bmatrix} \) to the left-hand side of the inequality without rendering it unsatisfied. Multiplying from the left and right with \( \xi_k = [\chi_k^* \ w_{p,k}^*]^T \)
then gives

\[
\xi_k \left[ \begin{array}{l}
\tilde{\mathcal{A}}(\Delta)^* \tilde{\mathcal{A}}(\Delta) \quad \chi(\Delta) \\
\tilde{\mathcal{B}}(\Delta)^* \chi(\Delta)
\end{array} \right] \xi_k + \\
\xi_k \left[ \begin{array}{l}
0 \\
I \tilde{\mathcal{D}}(\Delta)
\end{array} \right] P_p \left[ \begin{array}{l}
0 \\
I \tilde{\mathcal{D}}(\Delta)
\end{array} \right] \xi_k + \xi_k^T G \xi_k \leq 0
\]

which reduces to

\[
(\chi_{k+1} - \chi_k)^* X (\chi_{k+1} - \chi_k) + \\
\begin{bmatrix} w_{p,k}^* \\ z_{p,k}^* \end{bmatrix} P_p \begin{bmatrix} w_{p,k} \\ z_{p,k} \end{bmatrix} + \varepsilon w_{p,k}^* w_{p,k} \leq 0
\]

Summing from \( k = 0 \) to \( k = \infty \) with \( \chi_0 = 0 \) and \( \lim_{k \to \infty} \chi_k = 0 \) then yields (12), and hence requirement 1. implies RQP.

It is observed that, as in the continuous-time case, (16) is an LMI in the unknowns \( X \) and \( P_p \). The discrete-time controller synthesis can therefore be completely analogously with the continuous-time synthesis in [15]. The extended multipliers \( P_e \) in (14) is constructed from multipliers \( P \) and \( \tilde{P} \) of lower dimension such that

\[
P_e = \begin{bmatrix} P & * \\ * & * \end{bmatrix}, \quad P_e^{-1} = \begin{bmatrix} \tilde{P} & * \\ * & * \end{bmatrix}
\]

(18)

where \( P \) and \( \tilde{P} \) must fulfill the following requirements:

\[
P = \begin{bmatrix} Q & S \\ S^* & R \end{bmatrix}, \quad \begin{bmatrix} \Delta \end{bmatrix}^* \begin{bmatrix} \Delta \\ I \end{bmatrix} > 0 \quad \forall \Delta \in \Delta
\]

(19)

and

\[
\tilde{P} = \begin{bmatrix} \tilde{Q} & \tilde{S} \\ \tilde{S}^* & \tilde{R} \end{bmatrix}, \quad \begin{bmatrix} I \end{bmatrix}^* \begin{bmatrix} \tilde{P} & I \\ -\Delta^* \end{bmatrix} < 0 \quad \forall \Delta \in \Delta
\]

(20)

We will, due to a technicality in the controller construction, require \( \tilde{P} \) to have as many negative eigenvalues as the dimension of \( w_u \). It is now possible to construct the extended multipliers as

\[
P_e = \begin{bmatrix} P & U \\ U^* & N^{-1} \end{bmatrix}
\]

(21)

where the matrix \( U \) is chosen such that its columns form an orthogonal basis of the image of \( P - \tilde{P}^{-1} \), and such that

\[
U^* (P - \tilde{P}^{-1}) U = N = \begin{bmatrix} N_- & 0 \\ 0 & N_+ \end{bmatrix}
\]

(22)

with \( N_- < 0 \) and \( N_+ > 0 \).

\[
\begin{bmatrix} V_-(\Delta) & V_+(\Delta) \end{bmatrix} = \begin{bmatrix} \Delta^* & I \end{bmatrix} U
\]

(23)

with \( V_- \) and \( V_+ \) having \( \dim \{ N_- \} \) and \( \dim \{ N_+ \} \) columns, respectively. We will then construct the scheduling function \( \Delta_c(\Delta) \) in the following way:

\[
\Delta_c(\Delta) = N_- V_-(\Delta)^* \\
\left( [\Delta^* P [\Delta] - V_-(\Delta) N_- V_-(\Delta)^*]^{-1} \right) V_+(\Delta).
\]

(24)

The scheduling function can thus be constructed based on knowledge of the mapping \( \Delta \) and the multiplier submatrices \( P \) and \( \tilde{P} \). These multipliers are found by solving the following three coupled LMIs:

\[
\begin{bmatrix} X & I \\ I & Y \end{bmatrix} \succeq 0
\]

(25)

\[
\begin{bmatrix}
-\gamma \ I \\
\gamma \ I
\end{bmatrix} \succeq 0
\]

(26)

\[
\begin{bmatrix}
-\gamma \ I \\
\gamma \ I
\end{bmatrix} \succeq 0
\]

(27)

where

\[
\psi = \begin{bmatrix} I & 0 & 0 \\ A & B_u & B_p \\ 0 & D_{uu} & D_{up} \\ C_u & D_{pu} & D_{pp} \end{bmatrix}
\]

and

\[
\phi = \begin{bmatrix}
-A^* -C_u^* -C_p^* \\
B_u^* -D_{uu}^* -D_{up}^* \\
0 & I \\
0 & 0 \\
-B_p^* & -D_{pp}^* & I
\end{bmatrix}
\]

(28)

In the above, \( \tilde{P}_p = \begin{bmatrix} \tilde{Q}_p & \tilde{S}_p \\ \tilde{S}_p^* & \tilde{R}_p \end{bmatrix} \) denotes the inverse of the performance specification matrix \( P_p \).

To sum up, the synthesis progresses as follows. Assuming that matrices \( X, Y, P, \) and \( \tilde{P} \) solving (25)-(27) have been found, it is possible to construct the extended multiplier \( P_e \) and the scheduling function \( \Delta_c \) as given in (21) and (24), respectively. If \( X - Y^{-1} \) is of full rank, the Lyapunov matrix \( X \) can be calculated as

\[
X = \begin{bmatrix} X & I \\ I & (X - Y^{-1})^{-1} \end{bmatrix}
\]
in which case the controller will be of the same order as the
system. If, on the other hand, \( X - Y^{-1} \) is close to losing
rank, then \( X \) can instead be constructed as

\[
X = \left[ X \quad Z \right]
\begin{bmatrix}
Z^* (X - Y^{-1}) Z^{-1}
\end{bmatrix}^{-1}
\]

where the columns of \( Z \) form an orthonormal basis of
the image of \( X - Y^{-1} \). In this case the order of the
controller will be reduced by a corresponding number of
orders. (17) is a linear function of the controller matrices
\( (A_c, B_c, C_c, D_c) \), which means that (16) becomes a
quadratic matrix inequality (QMI) in \( (A_c, B_c, C_c, D_c) \). A
solution method for the QMI problem (16) based on the
number of positive and negative eigenvalues of the central
factor, can for instance be found in [15].

IV. CONTROLLER SYNTHESIS

In this section we apply the synthesis method outlined in
the previous section to the discrete-time motor model.
The model (5) was employed, using the following parameter
values previously identified from the actual motor setup:

\[
A_0 = \begin{bmatrix}
-320.7 & 140.0 \\
10.5 & -10.5
\end{bmatrix}, \quad B = \begin{bmatrix} 42.0 \end{bmatrix}, \quad C = \begin{bmatrix} 1 & 0 \end{bmatrix}
\]

and

\[
A_1 = \begin{bmatrix} j & -13.3 \end{bmatrix}, \quad A_2 = \begin{bmatrix} 0 & 140.0 \end{bmatrix}
\]

The contributions of the parameter variations to the
state equation could be described by \( B_u w_u = [A_1, A_2] [\sigma_1 \sigma_2] \). However, since \( A_1 \) and \( A_2 \) have rank 1, we
may write

\[
A_1 = U_1 \Sigma_1 V_1^* = \begin{bmatrix} u_1 \quad u_2 \end{bmatrix} \begin{bmatrix} \sigma_1 & 0 \\
0 & \sigma_2
\end{bmatrix} \begin{bmatrix} (u_1)^* \\
(u_2)^*
\end{bmatrix}
\]

and

\[
A_2 = U_2 \Sigma_2 V_2^* = \begin{bmatrix} u_1 \quad u_2 \end{bmatrix} \begin{bmatrix} \sigma_1 & 0 \\
0 & \sigma_2
\end{bmatrix} \begin{bmatrix} (u_1)^* \\
(u_2)^*
\end{bmatrix}
\]

and let \( B_u = [u_1 \sigma_1 \quad u_2 \sigma_2] \). It then follows that, with
\( z_u = c_u x, c_u = [v_1 \quad v_2]^* \), the parameter variation
can be written as \( \delta_1 A_1 x + \delta_2 A_2 x = B_u [\sigma_1 \sigma_2] z_u \), which is
advantageous in terms of implementation. The parameter variation channel \( z_u \rightarrow w_u \) was hence defined as follows:

\[
w_u = \Delta z_u = \begin{bmatrix} \omega & 0 \\
0 & i \omega / 4 m_d
\end{bmatrix} z_u.
\]

The performance and noise/reference channels were denoted \( z_p \) and \( w_p \). The performance output

\[
\begin{bmatrix} i_s - i_{s,ref} \\
\sigma u_s
\end{bmatrix}
\]

consisted of the control error and the control signal
weighted by a factor \( \sigma_u \). The performance input (or noise
channel)

\[
w_p = \begin{bmatrix} i_{s,ref} \\
u_m
\end{bmatrix}
\]

consisted of the stator current reference and measurement
noise. Finally, the measurement \( y \) was defined as the
control error corrupted by random measurement noise
\( \nu_m \in [-\sigma_v; \sigma_v] \), i.e.

\[
y = i_s - i_{s,ref} + \nu_m.
\]

The following nominal system could thus be constructed:

\[
\begin{bmatrix}
\dot{x} \\
\dot{z}_p \\
y
\end{bmatrix} = \begin{bmatrix}
A_0 & B_u & 0 \\
C_u & 0 & 0 \\
0 & D_{pp} & E_p
\end{bmatrix} \begin{bmatrix}
x \\
w_u \\
w_p
\end{bmatrix}
\]

The matrices \( C_p, D_{pp} \) and \( E_p \) were used to define the
weightings of the state, noise, reference, and control signal
contributions to the performance and measurement output,
respectively. Correspondingly, \( F_p \) accounted for the
weightings of the noise and reference contribution to the
measurement output. We chose \( \sigma_u = 10^{-6} \) and \( \sigma_v = 10^{-6} \).

The control error part of the performance channel was
augmented by a first-order filter that allowed frequency
tuning of the controller; the pole was placed in \( s = -100 \).
This system was then discretised with a sampling period of
600 Hz, using the bilinear transformation as described in
[1], yielding the discrete-time nominal system (6). As
discussed earlier, this sampling frequency was imposed by the
hardware setup. The eigenvalues of the discretised system
matrix were located at \( z = 0.5815, z = 0.9903 \) and
\( z = 0.9990 \).

The next step was to solve the LMIs (25)–(27) in order to
calculate a controller. The performance specification

\[
P_p = \begin{bmatrix} -\gamma I & 0 \\
0 & \frac{1}{\gamma} I
\end{bmatrix}
\]

providing a bound \( \gamma \) on the induced 2-norm

\[
\sup_{w_p \neq 0} \frac{\|z_p\|}{\|w_p\|} \leq \gamma
\]

was chosen and a bisection search for the smallest \( \gamma \) for
which (25)–(27) were feasible could then be performed. We
allowed \( \delta_1 = \omega \) and \( \delta_2 = 3 \{i_s\} / R \{i_m\} \) to obtain values in the
intervals \([-800 ; 800] \) and \([-10 ; 10] \), respectively.
Under these circumstances, a performance of \( \gamma = 0.0011 \)
could be achieved and a controller on the form (8) could be
obtained by solving the QMI (16). When solving the
synthesis QMI, it was found that the controller order could
be reduced by one, yielding a second-order controller with
eigenvalues of \( A_c \) in \( z = 0.9990 \) and \( z = 0.8202 \).
Fig. 3. LPV current control, simulation. The top figure shows the real and imaginary components, $u_{eq}$ and $u_{aq}$, of the control voltage generated by the controller. The middle figures show the real and imaginary components, $i_{rd}$ and $i_{qd}$, of the controlled currents, plotted with full lines $(-)$ along with their reference signals, plotted with dash-dotted lines $(- -)$. The bottom plot shows $\delta_1$ $(-)$ and $\delta_2$ $(- -)$ scaled to the interval $[-1; 1]$. As can be seen, the tracking of the current reference satisfies the performance requirement except when the control voltage saturates (at around 4 sec).

Fig. 4. LPV current control, simulation without scheduling. The top figure shows the real and imaginary components, $u_{eq}$ and $u_{aq}$, of the control voltage generated by the controller. The lower figure shows the real and imaginary components, $i_{rd}$ and $i_{qd}$, of the controlled currents, plotted with full lines $(-)$ along with their reference signals, plotted with dash-dotted lines $(- -)$. Without the gain scheduling the system becomes unstable.

Fig. 5. LPV current control, experimental results. The top figures show the real and imaginary components, $u_{eq}$ and $u_{aq}$, of the control voltage generated by the implemented LPV controller. The lower figures show the controlled current $i_{eq}$. The reference signals are shown with dash-dotted lines $(- -)$, while the measurements are shown with full lines $(-)$. The left figures are without load, while the right figures are recorded with a load torque of $m_L = 4$Nm.

V. SIMULATIONS

In the simulations the reference sequence was chosen as a series of steps, each with a duration of 250 samples. For each step, the reference for $i_{aq}$ was allowed to take random values in the interval $[-10; 10]$, while the reference for $i_{rd}$ was chosen from the interval $[1; 3]$. The system was disturbed by a load torque $m_L$, which was a sequence of uniformly distributed white noise filtered through a first-order filter with a time constant of $1/2$ second. Subject to these external signals, the nonlinear model generated the $\delta_1$ and $\delta_2$ sequences based on which the controller scheduling function was calculated. Motivated by limitations of the hardware of the experimental setup, the control voltage $u_s$ was made to saturate at 600V.

Figure 3 shows a simulation of the closed loop system. It is seen that the control loop achieves good tracking, in accordance with the performance value achieved for all values of the parameter variations, except when the control signal saturates. The parameter variations are shown in the bottom plot in Figure 3, scaled to the interval $[-1; 1]$. It is noted that the generated stator voltage compensates for the parameter variations throughout the allowed range. This scheduling is crucial to successful control; Figure 4 shows a simulation carried out under similar circumstances, but with the scheduling signal set to $w_s = 0$ (only voltage and current are shown). As can clearly be seen, the system becomes unstable in certain regions of the operating range if the gain scheduling is switched off.

VI. PRACTICAL EXPERIMENTS

The controller presented above was implemented in C without modifications on a standard PC. The power device is a voltage-sourced inverter controlled directly from the PC. The induction motor is a 1.5kW, two pole-pair motor with a rated torque of 10Nm. The first two experiments were open-loop current control experiments where the aim was to keep the magnetising current constant and make the imaginary part of the stator current follow a series of steps. The first experiment was conducted without load, while in the second experiment the motor shaft was subjected to a load torque of 4Nm. The results are shown in Figure 5, where it is observed that the current tracks the reference steps adequately well. Looking at the stator...
Fig. 6. LPV current control in cascade with rotational speed controller, experimental results. The top figure shows the real and imaginary components, $u_{v_d}$ and $u_{v_q}$, of the control voltage generated by the implemented LPV controller. The middle and lower figures show the controlled current $i_{v_d}$ and the rotational speed $\omega_r$, respectively. The reference signals are shown with dotted lines (---), while the measurements are shown with full lines (- - -). The current reference signal was generated by an outer loop speed controller.

Voltage, it is noted that the imaginary part of the voltage is significantly different between the two experiments. This is due to the two different disturbance load torques, which cause the scheduling controller to yield significantly different control signals. Some variation can be noted in the real part of the voltage as well, caused by cross couplings between the stator and magnetising currents.

In the third experiment the shaft speed loop was closed using an outer PI-controller. In this case the stator current reference signals were thus generated by the PI-controller, and the LPV controller had to track these signals. The results of this experiment is shown in Figure 6. As can be seen, the control loop performs satisfactorily.

VII. CONCLUDING REMARKS

In this paper a recently developed procedure for LPV controller synthesis, the so-called full-block S-procedure, has been applied to stator current control of an induction motor, which is a highly nonlinear dynamical system. This method required the model to be written on a linear fractional form, where the nonlinearity entered the model as parameter variations. Due to hardware limitations on the sample rate, it was chosen to discretise the system model. Hereafter, it was demonstrated that the controller synthesis can be formulated in discrete time as well.

A controller was constructed such that its dynamics depended on a scheduling function calculated from the parameter variation part of the model. The synthesis of the controller and scheduling function was then achieved by solving a set of linear matrix inequalities constructed from the model parameters. It was found via simulations that the gain-scheduled closed-loop system fulfilled a robust quadratic performance specification throughout the operating range, but that it would become unstable if the scheduling was switched off. The gain-scheduling is thus an integral part of the controller. The main contribution of this paper was then to show that a systematic non-conservative control design with more than one scheduling parameter could be implemented on a real-life system with fast dynamics.

Finally, some suggestions for further research could be to construct an entirely LPV-based controller from speed reference to measured speed, as well as to try to include robustness with respect to poorly known parameters such as the rotor resistance and inductance.

REFERENCES
