ROBUSTNESS ANALYSIS OF NONLINEAR SYSTEMS - APPLICATION TO INDUCTION MOTOR

Edouard Laroche

LSIIT (UMR CNRS-ULP 7005)
Parc d'innovation, bd S. Brant, 67400 Illkirch, France
Phone: 00 33 3 90 24 44 68, Fax: 00 33 3 90 24 44 80;
laroche@lsiit.u-strasbg.fr

Abstract: This paper is dedicated to stability analysis of a position control loop based on an induction motor with flux oriented control in the general framework of μ-analysis. The pertinence of using this methodology is discussed in the general case of a smooth nonlinear system when the equilibrium point varies slowly. A new linear fractional representation of the model is presented with low repetition indexes of the parameters. Several algorithms are used for evaluating μ bounds, allowing to compare them in term of accuracy and computation time.

Keywords: robustness, μ-analysis, linear fractional representation, nonlinear system, induction motor, flux oriented control

1. INTRODUCTION

Induction motor is a widely used actuator. Its most classical torque control strategy, called flux oriented control (FOC) requires the knowledge of the rotor parameters for tuning. If they are not well known, the torque cannot be well handled and parasitic dynamics appear that can affect an outer loop, such as a speed or position loop, and turn the system unstable. It then appears necessary to check the robustness of the system with respect to the variations of those parameters. This problem was observed in (Semail *et al.*, 1990). In (de Wit P.A.S. *et al.*, 1996), it is shown that FOC remains stable for large variations of the parameters. In (Cauët *et al.*, 2002), robustness analysis is performed via parameter-dependent Lyapunov functions and applied to the case of feedback linearization control. In (Laroche *et al.*, 2004) a linear fractional model was presented for evaluating robustness of an induction motor with FOC including a flexible link and μ-analysis results were provided. It included parameter variations of both motor and load. Nevertheless, the development of the linear fractional representation (LFR) was simplified as only one parameter was finally considered for the motor. Herein, this last approach is reconsidered. An improved model is developed including all the varying parameters. Several algorithms available in the literature are considered for evaluating the robustness margin, allowing to compare their efficiencies. Moreover, a general methodology is exhibited, which generalizes the considered case. Indeed, we discuss the use of μ-analysis for robustness analysis of smooth nonlinear systems in the particular case where nonlinear dynamics are inside an inner loop and can be considered as fast with respect to the outer loop.

A short description of the process in given in section 2. The general methodology is then discussed in section 3. The LFR model used for the robustness analysis is developed in section 4. The considered algorithms and numerical results are finally presented in section 5.

2. DESCRIPTION OF THE SYSTEM

The system considered herein is composed of an induction motor linked to an inertial load via a flexible link. The motor is torque controlled via the FOC algorithm and an outer loop is designed for position control. A brief presentation of the system follows. The reader may refer to (Laroche *et al.*, 2004) for complementary information.

2.1 Electromechanical system

The 3-phase induction motor is of 2.2 kW rated power and its voltage wave-forms are delivered by a voltage inverter. The motor is connected to a fixed inertial load by the mean of a flexible link. The dynamical model of the induction motor is well-known and can be found in the literature (Vas, 1990; Leonard, 1996).

2.2 Flux oriented control

Flux-oriented control is the most classical torque control strategy. It consists in choosing a change of frame so that the torque expressions simplifies into $T = PL_m i_m i_{sq}$, where P is the pole pair number, L_m is the magnetizing inductance, i_m is the magnetizing current and i_{sq} is the torque component of the stator current. Finally, FOC consists first in computing estimates $\hat{i}_{sq}$ of i_{sq} and $\hat{i}_m$ of i_m; second in controlling $\hat{i}_m$ at its nominal value and $\hat{i}_{sq}$ in order to impose the desired torque value T^*, which is the case with $i^*_{sq} = \frac{T^*}{P\hat{L}_m i^*_m}$.

This control strategy relies on the knowledge of rotor parameters: L_m and also the rotor resistance R_r through the rotor time constant $T_r = \frac{L_m}{R_r}$. When the parameters are known with accuracy, torque is well controlled and $T = T^*$ on a wide bandwidth (up to 1 kHz). Nevertheless, L_m varies with respect to magnetic saturation and R_r with respect to temperature. With estimation errors on R_r and L_m, the actual torque is not well-controlled any more. The system of input T^* and output T exhibits 2nd order dynamics.

2.3 H$_\infty$ controller

Assuming that the torque is perfectly controlled, transfer between torque reference T^* and the position was identified from experimental data. The position control loop is finally defined by $T^* = K(s)(\theta^* - \theta)$, where $K(s)$ is a H$_\infty$ controller designed by loop-shaping (McFarlane and Glover, 1992).

2.4 Robustness issue

The H$_\infty$ controller was designed for the nominal model and does assume that the torque is well-handled. In the case where rotor parameters vary, the parasitic dynamics of the torque modify the behavior of the loop that may become unstable. Checking whether the system remains stable for some parameter variation ranges appears to be crucial.

3. METHODOLOGY

3.1 Stability

Let us consider a controlled nonlinear system of reference signal r, output vector y and state model:

$$\dot{x} = f(x, r, p) \tag{1}$$

where p is the parameter vector, f being continuously derivable. For a given constant input r and a given parameter vector p, an equilibrium point x_0 is defined by the implicit relationship $f(x_0, r, p) = 0$. This equilibrium is stable if the Jacobian matrix:

$$A(x_0, r, p) = \frac{\partial f}{\partial x}(x_0, r, p) \tag{2}$$

is Hurwitz, i.e. if all its eigen values have strictly negative real parts. The system can be considered as locally stable if all the equilibrium points are stable, i.e. for any constant r in the admissible set.

Some remarks can be done concerning this definition of stability:

- Stability is guaranteed only for x being in a neighborhood of x_0.
- There is no guaranty that the regulation error $r - y$ is bounded for any trajectory.

In order to avoid these limitation, one may want to check that the Jacobian matrix:

$$A(x, r, p) = \frac{\partial f}{\partial x}(x, r, p) \tag{3}$$

is Hurwitz for any (x, r, p) of the admissible trajectories. In this case, stability is guaranteed in the case of low variations of the working point, namely of r. As the trajectory is generally not known in advance and as parameters are uncertain, trajectory of x is considered as uncertain and the Hurwitz criterion has to be checked for any x in a domain determined by simulation. Actually, matrix A generally depends on a limited number of entries of x_0 and r. These entries are considered as additive parameters included in an extended parameter vector $\tilde{p}$. This formulation may be pessimistic as variations of x_0 and r are considered as independent and the number of uncertain parameters is increased.

Nominal stability refers to the stability with the nominal value p^* of the parameter vector. *Stability robustness* is insured if the system remains stable for any value of p in the admissible domain. The robustness margin is the dilatation ratio that can be applied to the considered parameter domain while preserving stability. Thus, the system is robustly stable if the robustness margin is greater than one.

3.2 μ-analysis

If the linearized model $\dot{x} = A(\tilde{p})x$ is rational in $\tilde{p}$, a linear fractional representation can be obtained. It consists in a loop including a LTI model $M(s)$ closed with a static transfer $\Delta(\tilde{p})$. The parameter variation can be normalized so that $||\Delta(\delta)||_\infty \leq 1$ where δ is the normalized parameter vector. Assuming that $\tilde{p}$ (and then δ) is constant, stability robustness can be evaluated with the classical μ-analysis method. $\frac{1}{\mu}$ is then the size of the smaller Δ capable to turn the system unstable and is therefore the robustness margin.

μ-analysis has turned out to be a classical framework for robustness analysis of linear and smooth non-linear systems. It has been mainly applied to aeronautical and space applications (Döll *et al.*, 1998; Döll *et al.*, 1999) and also to other industrial systems (Laroche and Knittel, 2005).

3.3 Two-loops controlled system

Let us consider that the controlled system includes two loops:

- An inner loop controlling internal quantities that need to be limited; it has fast dynamics and is generally robust when used alone.
- An outer loop, with lower dynamics, forces the system output y to follow a desired reference trajectory r. The outer loop controller $K(s)$ is assumed to be linear.

The outer-loop controller is generally designed based on the nominal model of the inner loop. Nevertheless, uncertainties on the system may yield instability in the outer loop.

This situation is classical in electric engineering where an inner loop controls the torque (and the currents) of the electric motor whereas an outer loop controls the speed or position, which is the present case. It is also classical in robotics where an inner loop generally controls the joint speeds whereas an outer loop controls the end-effector position.

In this control scheme, the setting point depends on the outer-loop reference signal. If this reference signal varies slowly with respect to the inner dynamics, the inner loop can be considered as being around its equilibrium point. Assuming that the inner loop model is given by $\dot{x}_i = f_i(x_i, u, p)$ and $y = g(x_i, u, p)$, one can restrict the robustness evaluation to the equilibrium manifold defined by the implicit relationship: $f_i(x_{i0}, u_0, p) = 0$. The equilibrium point being given by $x_{i0} = \phi(u_0, p)$, the stability can be studied by considering the loop including $K(s)$ and the linearized model of the inner-loop:

$$\begin{cases} \dot{x}_i = A_i(\phi(u_0,p), u_0, p)x + B(\phi(u_0,p), u_0, p)u \\ y = C(\phi(u_0,p), u_0, p)x + D(\phi(u_0,p), u_0, p)u \end{cases}$$

where u_0 and p vary in their domains and with $A_i(x_i, u, p) = \frac{\partial f_i}{\partial x_i}$, $B(x_i, u, p) = \frac{\partial f_i}{\partial u}$, $C(x_i, u, p) = \frac{\partial g}{\partial x_i}$ and $D(x_i, u, p) = \frac{\partial g}{\partial u}$. The obtained extended parameter vector $\tilde{p}$ (including the components of u_0 that occur in the linearized model) is usually far smaller than the one obtained by keeping the dependence of x_i.

In the present case, the position loop can be considered as relatively slow with respect to the torque dynamics; the presented methodology is therefore applied.

4. LINEAR FRACTIONAL REPRESENTATION

Assuming that the stator currents are perfectly controlled ($\hat{i}_m = i_m^*$ and $\hat{T} = T^*$), the model of the induction motor with FOC can be obtained as in (Laroche *et al.*, 2004):

$$\begin{aligned} \frac{d\xi}{dt} &= \frac{R_r}{L_m i_m}(-i_m^* \sin(\xi) + \frac{T^* \cos(\xi)}{P\hat{L}_m i_m^*}) - \frac{\hat{R}_r T^*}{P\hat{L}_m^2 i_m^{*2}} \\ \frac{di_m}{dt} &= \frac{R_r}{L_m}(i_m^* \cos(\xi) + \frac{T^* \sin(\xi)}{P\hat{L}_m i_m^*} - i_m) \\ T &= PL_m i_m(-i_m^* \sin(\xi) + \frac{T^* \cos(\xi)}{P\hat{L}_m i_m^*}) \end{aligned}$$

where ξ is the angle error of the frame change. Following the method presented in section 3, the model linearized around the equilibrium point can be determined. This model was presented in the form of a transfer function in (Nordin *et al.*, 1985). After rearranging the parameters, it is possible to consider only 4 varying parameters: $1/T_r$, $\lambda = L_m/\hat{L}_m$, $\zeta = T_r/\hat{T}_r$ and $\iota = (\hat{i}_{sq}/\hat{i}_m)^2$, instead of the 5 parameters of the nonlinear model (L_m, $\hat{L}_m$, R_r, $\hat{R}_r$ and i_m^*).

As the parameter dependence is rational, the model can be put into the LFR of figure 1. This representation is not unique and different methods can be used, providing representations with different repetition indexes.

Factorizing the transfer function in order to lower the parameter repetition indexes, the LFR was

Table 1. Repetition indexes

	tf	tf red	ss	ss red
$1/s$	4	4	2	2
$1/T_r$	4	4	2	2
λ	1	1	1	1
ζ	10	8	6	5
ι	6	4	3	2

obtained with the LFR toolbox (Magni, 2001). The repetition indexes obtained by (Le Guennou, 2003) are given in the first column of table 1 ('tf' for transfer function). The model is of order 4 in state (instead of initial order 2) and the highest repetition index is 10. Using the reduction algorithms available in the mentioned toolbox, the order was sensibly reduced (column 'tf red').

A second way for obtaining the LFR is to use directly the state space model. Indeed, it is already a LFR of $\frac{1}{s}I_n$ with matrix:

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix}, \tag{4}$$

the repetition index of $1/s$ being $n = 2$. The change of state vector $x = [\xi \ i_m \frac{(1+\iota)(1+\iota\zeta^2)}{\sqrt{\iota}}]^T$ was first used, then factorization and commutation of multiplications between scalars; the following model was finally obtained:

$$\dot{x}_1 = \frac{1}{T_r}(w - x_1 - \iota\zeta x_2), \tag{5}$$

$$\dot{x}_2 = \frac{1}{T_r}(w + \zeta x_1 - x_2), \tag{6}$$

$$T = \frac{\lambda}{1+\iota\zeta^2}(-x_1 + \iota\zeta(x_2 + T^*) + T^*), \tag{7}$$

with $w = (1-\zeta)T^*$. The repetition indexes of the parameters are given in 'ss' column (for state-space) of table 1. One can notice that they are sensibly decreased; reduction algorithms allowed to even decrease the repetition indices (column 'ss red').

Finally, parameters λ and ζ are replaced by their expressions depending on R_r and L_m, the variations of those parameters being independent. The final model then includes 3 varying parameters; their variation ranges and repetition indexes are given in table 2.

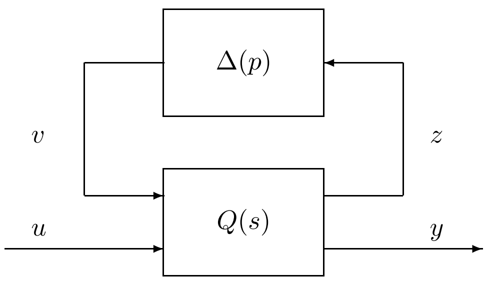

Fig. 1. LFR Model

Table 2. Repetition indexes

	range	index
ι	[0, 4]	2
L_m	± 50 %	6
R_r	± 50 %	5

5. COMPUTATION OF μ

Several algorithms available in the literature were used, allowing to compared their results and computation time. The final model is a LFR composed of $M(s)$ (including $Q(s)$ and $K(s)$) and $\Delta \in E_\Delta$; E_Δ being the set of matrices with the particular structure and Δ being normalized so that $||\Delta||_\infty \leq 1$ for the largest parameter deviations.

5.1 Upper bound algorithms

5.1.1. Upper bound with frequency grinding. The classical method for estimating μ is to consider the system in the frequency domain ($s = j\omega$) for a sampled set of frequencies. Finally, μ is the maximum value of the $\mu(\omega)$. The interest is that the complexity of the method is polynomial with respect to the repetition indexes. The main drawback is that peaks in the frequency plot may be underestimated. The most classical algorithm is due to (Young *et al.*, 1995).

5.1.2. Guaranteed upper bound. Let us consider the LFR of a LTI system of order n (composed of (4) and $\frac{1}{j\omega}I_n$). Quantity $\frac{1}{\omega}$ can be considered as an uncertain parameter varying over some interval. Including $\frac{1}{\omega}$ as an additional parameter in the LFR of $M(s)$, an upper bound of $\mu(\omega)$ over the interval can be computed. Thanks to this method, the high peaks are not underestimated. The change of variable $\frac{1}{\omega} = \frac{1-z}{1+z}$ allows to consider the whole range $[0, \infty]$ with z varying over $[-1, 1]$.

If μ is directly computed for a given interval of frequencies $[\omega_1, \omega_2]$, the robustness is finally evaluated over an interval of the same center $\frac{1}{2}(\omega_2 - \omega_1)$ but of size $\frac{\omega_2 - \omega_1}{\mu^*}$ where μ^* is the obtained μ value. The algorithm used herein is from (Friang *et al.*, 1998); it adds a tuning parameter with serial connection to $\frac{1}{\omega}$ that is tuned iteratively in order to obtain a value of μ corresponding to the prescribed interval $[\omega_1, \omega_2]$.

5.1.3. Topological separation. Let us consider the LFR composed of the operators M and Δ with:

$$\begin{cases} z = Mv \\ v = \Delta z \end{cases} \tag{8}$$

It is known that the interconnection is stable if the graph of M and the graph of Δ^{-1} are disjoint (Safonov, 1980); the graph of M denoting the set couples (v, z) verifying $y = Mv$. From this notion, Iwasaki and Hara have developed conditions allowing accurate evaluation of the structured singular value (Iwasaki and Hara, 1998):

$$\mu(M) = \inf_{\gamma>0, \Theta \in E_\Theta} \left\{ \gamma : [\gamma I \; M] \, \Theta \begin{bmatrix} \gamma I \\ M^* \end{bmatrix} < 0 \right\} \tag{9}$$

They show that the upper bound based on $D-G$ scaling is obtained with a particular form of Θ and they obtain a tighter upper bound with:

$$\begin{array}{c} \Theta = \begin{bmatrix} R & S \\ S^* & Q \end{bmatrix} : R_{ii} \leq 0, i = 1, ..., \alpha, \\ [\Delta_k \; I] \, \Theta \begin{bmatrix} \Delta_k^T \\ I \end{bmatrix} \leq 0, k = 1, ..., 2^\alpha \end{array} \tag{10}$$

where R_{ii} of size $k_i \times k_i$ is the i^{th} block of R, k_i being the repetition index of the i^{th} parameter and Δ_k are the vertices of E_Δ.

For a given γ, solving the inequalities is a LMI feasibility problem. As γ enters nonlinearly in (9), μ cannot be computed directly. A dichotomy can be made from an initial interval with the drawback of increasing the computational cost, which is already high and varies exponentially with respect to the number of parameters.

5.2 Lower bound algorithms

5.2.1. Root locus. A simple way to evaluate the robustness margin is to check the roots of the system for parameters varying in their domains. Let us consider the variations of the maximum real part of the roots with respect to a dilatation ratio ρ applied to the parameter domain. This quantity is negative for $\rho = 0$, as the nominal model is assumed to be stable. It is non decreasing and the value of ρ for which it is zero is the robustness margin. As only a finite subset of the parameter domain is considered for computing the roots, a lower bound of μ is therefore obtained. More details on the use of this method can be found in (Laroche and Knittel, 2005).

5.2.2. Root migration. An original method consists in looking for the smaller parameter variation that moves a given pole toward the imaginary axis (Döll *et al.*, 1998). In a first step, the parameter displacement is chosen in order to minimize Frobenius norm of Δ, as an analytic expression is available. In a second step, the H_∞ norm of Δ is minimized. These steps are processed with several poles of the systems and the smallest Δ is finally chosen.

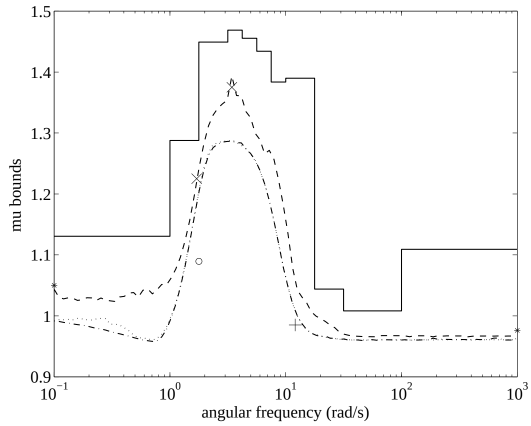

Fig. 2. μ plot (plain line: guaranteed with DG scalings, dashed: DG with Uc, dots: DG with UC option and DG from LMI control toolbox, $\times$: top. sep., o: root locus, +: root migration)

5.3 Numerical results

The different methods presented in this section were used for computing μ. In table 3 are given the corresponding results. For the different methods and the options used in the algorithm are given μ lower or upper bounds, the corresponding angular frequencies and the CPU times on an Athlon-1.4 GHz PC. The results are also graphically summarized in figure 2. Among the lower bound methods, one can notice that the root migration method (column 3) is notably faster than the root locus method (column 2), but it was not able to find a better lower bound. The worst case is obtained for the maximum value of L_m and the minimum value of R_r.

Among the different algorithms based on DG scalings, one can notice that changing the option 'Uc' for 'UC', in the function of the μ-analysis and synthesis toolbox allows a slight improvement in the accuracy (8 %) but with a prohibitive computation time. The methods from the μ-analysis and synthesis toolbox with option 'UC' and the LMI control toolbox provides similar results with similar computation time. More interesting is the use of the guaranty method ('DG guar.'); the result is quite accurate and the computation cost is not increased too much.

The upper bounds were validated with the topological separation method (last column). Two particular frequencies were chosen: the one of the upper bound peak and of the root-locus. In the present case, no improvement is found over the other methods, for a higher computational time.

Finally μ is guaranteed to be in interval [1.095, 1.5], which is not very accurate. Simulation of the nonlinear model with the worst admissible case obtained from the lower bound ($\delta_2 = 1/\mu_{\min} = 0.91$, $\delta_3 = -1/\mu_{\min}$) were processed. For compar-

Table 3. Numerical results

	lower bounds		upper bounds				
method	root locus	root mig.	DG scal.		DG LMI	DG guar.	top. sep.
options	1000 values	meth.='imd'	Uc	UC	100 pts	Uc	4 iter
			100 pts	100 pts		$\omega_{\max}/\omega_{\min} = 1.5$	
μ	1.095	0.985	1.40	1.29	1.29	1.47	1.375
ω (rad/s)	1.73	12.1	3.43	3.43	3.51	[3.16, 4.22]	3.4
CPU time (s)	54	4.46	37	2040	8930	64	1020

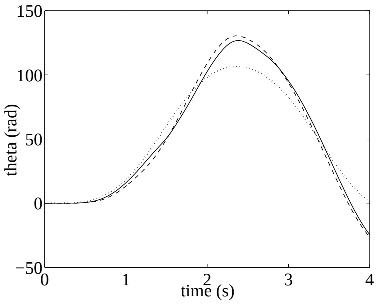

Fig. 3. Simulation results: position (plain: worst case, dashed: nominal case, dots: reference)

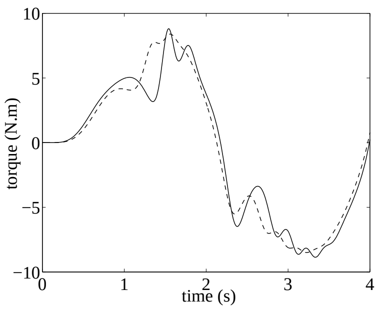

Fig. 4. Simulation results: torque (plain: worst case, dashed: nominal case)

ison, results of the nominal case are also given. Angular position are shown in figure 3; the torque in shown in figure 4. The system appears to follow the trajectory satisfactorily. Conjecturing that that the actual value of μ is 1.095, the corresponding admissible variations on L_m and R_r are $\pm$ 45 %.

REFERENCES

Cauët, S., L. Rambault, O. Bachelier and D. Mehdi (2002). Parameter-dependent lyapunov functions applied to analysis of induction motor stability. *Control Engineering Practice* **10**(3), 337–345.

de Wit P.A.S., Ortega R. and Mareels I. (1996). Indirect field-oriented control of induction motors is robustly globally stable. *Automatica* **32**(10), 1393–1402.

Döll, C., Magni J.F., Looye G. and Bennami S. (1998). Robustness analysis applied to autopilot design: Evaluation of a new tool for μ-analysis. In: *Proc. of ICAS Congress*. Vol. 1.

Döll, C., G. Ferreres and J.F. Magni (1999). μ tools for flight control robustness assessment. *Aerospace Sciences and Technologies* **3**(3), 177–189.

Friang, J.P., G. Duc and J.P. Bonnet (1998). Robust autopilot for a flexible missile: loop-shaping H_∞ design and real ν-analysis. *International Journal of Robust and Nonlinear Control* **8**, 129–153.

Iwasaki, T. and S. Hara (1998). Well-posedness of feedback systems: Insights into exact robustness analysis and approximate computations. *IEEE trans. Automatic Control* **43**(5), 619–630.

Laroche, E. and D. Knittel (2005). An improved linear fractional model for robustness analysis of a winding system. *Control Engineering Practice* **13**(5), 659–666.

Laroche, E., Y. Bonnassieux, H. Abou-Kandil and J.L. Louis (2004). Controller design and robustness analysis for induction machine-based positioning system. *Control Engineering Practice* **12**, 757–767.

Le Guennou, P. (2003). *Robustness of the induction motor - μ-analysis and sensivity analysis (in French)*. Master thesis, Université Louis Pasteur. Strasbourg/France.

Leonard, W. (1996). *Control of Electrical Drives*. Springer-Verlag.

Magni, J.F. (2001). Linear fractional representation with a toolbox for use with matlab. Technical report. ONERA.

McFarlane, D.C. and K. Glover (1992). A loop shaping design procedure using H_∞ synthesis. *IEEE trans. on Automatic Control* **37**(6), 759–769.

Nordin, K.B., D.W. Novotny and D.S. Zinger (1985). The influence of motor parameter deviations in feedforward field orientation drive systems. *IEEE trans. on Industry Applications* **21**(4), 1009–1015.

Safonov, M.G. (1980). *Stability and Robustness of Multivariable Feedback Systems*. MIT Press.

Semail, B., E. Mendes, F. Bouillault and A. Razek (1990). Vector controlled induction machine simulation-parameter sensitivity analysis. In: *proc. ElectrIMACS*.

Vas, P. (1990). *Vector Control of AC Machines*. Oxford university press.

Young, P., M. Newlin and J. Doyle (1995). Computing bounds for the mixed μ problem. *Int. J. Robust Nonlinear Control* **5**, 573–590.