Design of $\mathcal{H}_\infty$ Control and Simplified Instrument Fault Detection on DC Motor

Lintang Erlangga¹ Bagas Budhi Permana² Ulynnuha Setiyadi³ Samiadji Herdjunto⁴ Sujoko Sumaryono⁵

Abstract—In the presence of uncertainty, a closed-loop system must be able to maintain its stability and performance. Uncertainty can be categorized into two meanings, disturbance from external systems and perturbation in a dynamical system. In this paper, we will present a robust control design by considering a disturbance from external factors, and analyze its robustness using Structured Singular Value respect to parametric uncertainty and external disturbance. Beside that, every control system has their own region to operate, outside that region a performance degradation will occur, and led into instability. One factor to that catastrophic is a fault in instrument reading, to anticipate it, this paper also implement Simplified Instrument Fault Detection method to detect a sensor fault in order to give an alarm for further action.

Index Terms—robust control, uncertainty, fault, fault detection

I. INTRODUCTION

DC motor is one of the most versatile electromechanical system. There are various implementations in a diversity of subjects, e.g. robotics, industrial process, and so forth. To operate it, it is necessary to design a controller in order to regulate it reaches a set point. A challenging problem arises when one is naturally do not know the exact value of the physical parameters, e.g. resistance, rotor inertia etc. It is from an internal perspective of the system, there is also a disturbance come from the external of the system. Indeed, the objective of the controller will be agitated. To deal with it, it is important to consider at least one kind of uncertainty in a controller design step.

Due to various of DC motor applications, it is considerable to make it more safety, instrument fault is one factor that can lead into the malfunction of the system. The human eyes are not being able to observe directly the presence of degradation in system performance, there must be an "alarm" that warn a human operator so that corrective action can be taken. The fault term here means undesired deviation in system behaviour, it is limited into a such problem that tolerated yet as in [6] stated. This paper also provide some useful tool program written in C++ for dealing with robust control and fault diagnosis, which is called Just Another Control Library (JACL).

The subsequent sections are organized as follows. In section II, we discuss several approaches that attempt to make a

controller is robust respect to uncertainty. In section III, we present our DC motor model being used. In section IV, we explain our consideration when designing SIFD. In section V, we discuss our discrete $\mathcal{H}_\infty$ controller. Last, we report our results on SIFD performance, then analyzing the controller performance based on analysis with μ -analysis.

II. RELATED WORK

There are several approaches to maintain a closed-loop system performance in the presence of uncertainty. The first is adaptive control, the main idea is to utilize an "estimator like" for estimating every state variable in a system. Because the input and output signals carrying information about the state, an adaptive control system is most likely trying to "learn" every change on the physical parameter by utilizing the available signals, as in [9] which using the most commonly known adaptive control method which called Model Reference Adaptive Control (MRAC). But the irony is, when the available signals is not reliable and the real parameters is also not known. Another possible approach is to use Lyapunov control. At first, the Lyapunov term is used to refer stability analysis, that can be considered as an energy function. Recall that every action needs an energy, if the energy is dissipated, then the action is gone, and enable to a system to reach its equilibrium state e.g. pendulum. For design purposes, it can be considered as choosing an energy function which called Lyapunov function, then attempts to find a controller that makes the derivative of the Lyapunov function is definite negative as in [10], the main difficulty is, there is no concrete solution to find the Lyapunov function. Next approach is using fuzzy control, a fuzzy control approach can be taken to tackle a combination of varying parameter of a system, roughly speaking by trying to combine the every possible system parameter values with a certain weight as in [11]. Indeed, the combination become larger and complex due to proportional with parameter variation interval. Hence, our approach is using a robust control method, although the parameter variation cannot be included to certain methods, but at least in the design process this approach can be dealing with external disturbances. Then, we analyze the controller respect to uncertainty using Structured Singular Value.

To deal with an instrument fault, the proposed method from [4], which called Simplified Instrument Fault Detection (SIFD) was used, there is a predecessor of SIFD which called the Instrument Fault Detection (IFD). The consideration taken here is due to the SIFD simplicity which only needs one dedicated observer, and another reason is, it is possible

^{1,2,3}Department of Electrical and Information Engineering, Universitas Gadjah Mada, Yogyakarta, Indonesia, {lintang.e, bagas.budhi.permana, ulynnuha.setiyadi}@mail.ugm.ac.id

^{4,5}Department of Electrical and Information Engineering, Universitas Gadjah Mada, Yogyakarta, Indonesia, {samiadji, sujoko}@ugm.ac.id

Fig. 1. DC motor model.

the dynamics of one dedicated observer in the IFD become unobservable.

III. MODEL OF DC MOTOR

The used model of the DC motor is separated each for a controller and a fault detection process. Since the SIFD at least needs three sensors to work. Both the controller and the fault detection dynamics can be produced by the same dynamics as in “Fig. 1” to satisfy the need of the comparison in SIFD. The only difference is on the number of the output used. For controlling purpose the model has three states and single-input-single-output (SISO), and for fault detection, the model has three states and single-input-multi-output (SIMO), the three states consist of position, velocity and armature current. As a first step to our work, we derive the model in continuous time domain and ended with a discretized model.

A. Continuous-time model

Start with two differential equations based on mechanical and electrical perspective.

$$\frac{d\dot{\theta}}{dt} = \frac{1}{J_m} (-\bar{B}_m \dot{\theta} + \bar{K}_i i) \quad (1)$$

$$\frac{di}{dt} = \frac{1}{L_a} (-\bar{K}_b \dot{\theta} - \bar{R}_a i + v_{in}) \quad (2)$$

where

- $\bar{B}_m$ (viscous friction) = $4.36 \times 10^{-5} \text{ Nm/s/rad}$
- $\bar{J}_m$ (rotor inertia) = $5.71 \times 10^{-7} \text{ kgm}^2$
- $\bar{K}_i$ (torque constant) = $3.07 \times 10^{-1} \text{ Nm/A}$
- $\bar{L}_a$ (armature inductance) = $1.97 \times 10^{-3} \text{ H}$
- $\bar{K}_b$ (electromotive force constant) = $3.07 \times 10^{-1} \text{ Vs/rad}$
- $\bar{R}_a$ (armature resistance) = 21.6Ω

The physical parameters were obtained from Maxon 2322.983-11.225-200 model datasheet [5], and note that, bar notation denotes the nominal value of the parameters. In a state-space form, after we substitute the nominal value, we got

$$\frac{d}{dt} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & -76.357 & 53765 \\ 0 & -15.584 & -10964 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 507.61 \end{bmatrix} u \quad (3)$$

where $x_1 = \theta$ (rotor position), $x_2 = \dot{\theta}$ (rotor velocity) and $x_3 = i_a$ (armature current). The output matrix for fault detection

$$y_{FD} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \quad (4)$$

and for controller design

$$y_K = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \quad (5)$$

B. Discrete-time model

Due to the availability of the computational resources are digital computer. The model should be transformed into their equivalent model in discrete-time. By using our tools, JACL, by means compute a certain exponential matrix, the result is

$$\sigma \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 1 & 0.0009 & 0.0042 \\ 0 & 0.8636 & 4.2948 \\ 0 & -0.0012 & -0.0062 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} 0.001 \\ 2.1233 \\ 0.0436 \end{bmatrix} u \quad (6)$$

where σ is a shift operator. Both output matrix stay equal for both the controller and the fault detection.

IV. SIMPLIFIED INSTRUMENT FAULT DETECTION (SIFD)

The main principle of the SIFD is analogous to its predecessor [3], instead of using multiple instrument on each reading (e.g. use multiple sensor to observe an armature current), this method use functional redundancy term, that is utilizing several dedicated observers to get an estimation for each instrument output, the process was illustrated in “Fig. 2”. The difference is located in the number of a fault occurred, the SIFD is only capable to handle one fault at a time.

The concept of the dedicated observer is similar to separation principle, partitioning a state into an observed and unknown states. The dedicated term expresses the estimated value on a certain states from an observer is equal with an instrument reading. Then, a fault in each instrument can be indicated by observing a remarkable deviation between estimated and raw output. Based on (4) and (6), hence there are two state variables which need to estimate, since one of them is considered as a dedicated variable. Here, the dedicated state variables were selected based on how noisyless they are (i.e. by inspecting how steady is the raw data by giving a step signal), in this case is the rotor position. To curious readers, you can refer to [3] for further information. The robust pole placement method as proposed in [7] was used here to seek a dedicated observer gain, the obtained gain is

$$L = \begin{bmatrix} 1016.8 \\ 40.373 \end{bmatrix}$$

which is correspond to poles at 0.8 and 0.5.

V. DISCRETE $\mathcal{H}_\infty$ CONTROL

The main purpose of the $\mathcal{H}_\infty$ synthesis process, based on [1] is trying to find a stabilizing controller that make a particular metric between disturbance and performance signals is suppressed, the stabilizing controller means the controller is fulfill the internally stability condition. The metric used is $\mathcal{H}_\infty$ -norm. For facilitating the synthesis process, every control design must be transformed into Linear Fractional Transformation (LFT), in $\mathcal{H}_\infty$ synthesis the special form of LFT is used which called Lower LFT, see “Fig. 3”. Entire exogenous signals are represented by w comprise of disturbance, noise, reference, etc. The signal z can be regarded as our performance benchmark such as tracking error, input action etc. Then, signals y_G and u_G are represent the input and output signals that directly connected to the controller in design process, respectively. The G is called an inter-connection matrix which usually consists of plant, weighting function, etc. And K is our stabilizing controller, i.e. our main subject to find.

Assume G is defined as

$$G := \begin{bmatrix} G_{11} & G_{12} \\ G_{21} & G_{22} \end{bmatrix} \quad (7)$$

. Then the synthesis process is attempted to seek a stabilizing controller K so that $\|\mathcal{F}_l(G, K)\|_\infty < \gamma$, where $\|\cdot\|_\infty$ denotes infinity-norm and $\mathcal{F}_l(G, K)$ represents a transfer function from w to z . The approach taken here is formulated in [2] by solving two discrete algebraic Riccati equations, which are a part of the solution to $\mathcal{H}_\infty$ control problem in state-space form. To begin the process, first, find the realization of (7) as follows

$$G := \left[\begin{array}{c|cc} A & B_1 & B_2 \\ \hline C_1 & D_{11} & D_{12} \\ C_2 & D_{21} & D_{22} \end{array} \right] \quad (8)$$

, then check the necessary condition which requires (A, B_2, C_1, D_{12}) has no invariant zeros on the unit circle and is left invertible. The second necessary condition is also similar which assume that (A, B_1, C_2, D_{21}) has no invariant zeros on the unit circle and is right invertible. Then we verify both conditions using a theorem which equivalent each other.

Fig. 2. Illustration process of SIFD.

Our closed-loop design is illustrated in “Fig. 4”, instead of finding (7) in frequency domain, then back to time domain to get (8), we used a proposed method from [1] by defining a new state vector, then put together every signal which corresponds to w , also apply to all signals which correspond to z , and by the same token to y_G and u_G . Assume the realization of P

$$P := \left[\begin{array}{c|c} A_P & B_P \\ \hline C_P & D_P \end{array} \right] \quad (9)$$

then our resulting realization as follows

$$G := \left[\begin{array}{c|c|c|c|c} A_P & B_P & 0 & 0 & B_P \\ \hline -C_P & 0 & 0 & I & 0 \\ \hline 0 & 0 & 0 & 0 & I \\ \hline -C_P & -D_P & -I & I & -D_P \end{array} \right] \quad (10)$$

where (9) is corresponds to (6).

To check whether the assumptions as stated in [2] is satisfied or not, we use a technique from [1] by checking a certain pair of matrix system (e.g. (A, C) and (A, B)), whether it has uncontrollable or unobservable mode in the unit circle by PBH test. As notified previously, there are two discrete algebraic Riccati equations required to seek a stabilizing solution. To solve those equations, we used a method from [8] by finding an invariant spectral subspace by means find a generalized eigenvector based on a particular symplectic matrix in Riccati domain. The generalized eigenvectors can be obtained by using Schur decomposition along with ordering the eigenvectors based on its corresponding eigenvalue location, in this circumstance whether the eigenvalues inside the unit circle or not. After that, the rest of the computation can be calculated directly. Assume the realization of K

$$K := \left[\begin{array}{c|c} A_K & B_K \\ \hline C_K & D_K \end{array} \right] \quad (11)$$

our controller result from the synthesis process is as follows

$$\begin{aligned} A_K &= \begin{bmatrix} 0.998432 & 0.000099 & 0.000192 \\ -0.108719 & 0.988804 & 3.243435 \\ -0.030702 & -0.001133 & 0.331035 \end{bmatrix} \\ B_K &= \begin{bmatrix} -0.003858 \\ -0.028090 \\ 0.000155 \end{bmatrix} \\ C_K &= [-1.156189 \quad -0.007224 \quad -0.035401] \\ D_K &= [0.004672] \end{aligned} \quad (12)$$

Fig. 3. $\mathcal{H}_\infty$ synthesis framework.

Fig. 4. Block diagram of closed-loop system.

Fig. 5. Three types of fault considered in SIFD.

with $\gamma = 2.7$.

VI. RESULT

The SIFD framework was tested in two conditions, in the presence of fault and nothing, both conditions are stimulated by a unit step signal, with $\theta = 97^\circ$. To generate the fault, as in [3], we use three types of fault: bias, scale, and dead-zone, as illustrated in “Fig. 5”. We are also present comparison between simulation and real execution.

Input Step without Fault

Even in the absence of fault, the error is still appear as in “Fig. 7, and 9” shown, this condition is normal, caused due to delay in the dynamics of dedicated observer when trying to converge to the goal estimate. In an ideal situation (e.g. simulation) the dedicated observer perfectly converges to a certain value. The peak values were summarized in “Tab. I”

Fig. 6. The speed error given input step without fault (simulation).

Fig. 7. The speed error given input step without fault (reality).

TABLE I
PEAK ERRORS WITHOUT FAULT

Sensor Name	Peak Value (absolute)	
	Maximum	Minimum
Speed (simulation)	0	0
Speed (reality)	$\approx 2.7 \times 10^3$	$\approx 7.2 \times 10^2$
Current (simulation)	0	0
Current (reality)	$\approx 6.8 \times 10^2$	$\approx 6 \times 10^2$

Input Step with Fault

In this test, we limit only one type of fault that occurred at a time. In “Fig. 10, 11, 12, and 13” we present errors response under scale fault $k = 1.2$ condition. In “Fig. 14, 15, 16, and 17” we present error responses under bias fault $b = 30$ condition. In “Fig. 18, 19, 20, and 21” we present errors response under dead-zone fault $w = 10$ condition. It is very clear that under the fault condition, errors are very differ significantly. The peak values were summarized in “Tab. II”.

We only need the order of the peak values, due to no exist exact methods to determine a threshold. Then, based

Fig. 8. The current error given input step without fault (simulation).

Fig. 9. The current error given input step without fault (reality).

Fig. 12. The current error given input step with scale fault $k = 1.2$ in position reading (simulation).

Fig. 10. The speed error given input step with scale fault $k = 1.2$ in position reading (simulation).

Fig. 13. The current error given input step with scale fault $k = 1.2$ in position reading (reality).

Fig. 11. The speed error given input step with scale fault $k = 1.2$ in position reading (reality).

Fig. 14. The speed error given input step with bias fault $b = 30$ in position reading (simulation).

Fig. 15. The speed error given input step with bias fault $b = 30$ in position reading (reality).

Fig. 18. The speed error given input step with dead-zone fault $w = 10.0$ in position reading (simulation).

Fig. 16. The current error given input step with bias fault $b = 30$ in position reading (simulation).

Fig. 19. The speed error given input step with dead-zone fault $w = 10.0$ in position reading (reality).

Fig. 17. The current error given input step with bias fault $b = 30$ in position reading (reality).

Fig. 20. The current error given input step with dead-zone fault $w = 10.0$ in position reading (simulation).

Fig. 21. The current error given input step with dead-zone fault $w = 10.0$ in position reading (reality).

TABLE II
PEAK ERRORS WITH FAULT IN POSITION

Sensor Type	Peak Value (absolute)	
	Maximum	Minimum
Speed (simulation)	$\approx 6 \times 10^3$	≈ 8
Speed (reality)	$\approx 4 \times 10^3$	$\approx 2.6 \times 10^2$
Current (simulation)	$\approx 2 \times 10^5$	≈ 5
Current (reality)	$\approx 3 \times 10^3$	$\approx 5 \times 10^2$

on previous data, it is reasonable to set the threshold for the error in the speed sensor, $T_2(\text{speed}) = 4$ in simulation and 4.4×10^3 in real execution, and for the error in the current sensor, $T_3(\text{current}) = 2.5$ in simulation and 7.5×10^2 in real execution. That is what we had done to set the thresholds, by comparing a possible maximum peak error in the absence of a fault with a possible minimum peak error when a fault is appear, so that the SIFD scheme is quite robust under a control action, even though there is a few intervals which ignored where a fault is present.

Controller Performance

Similarly with previous test, we also present simulation and real execution results to evaluate the controller performance. The response from our closed-loop system can be seen at “Fig. 22, and 23” for simulation and real execution, respectively. The performance parameters were summarized in “Tab. III”.

There is a significant difference between the theoretical and the reality performance – due to difficulty to encompass a

TABLE III
PERFORMANCE PARAMETER

Control Method	Time (s)			Overshoot (%)
	Rise	Peak	Settling	
$\mathcal{H}_\infty$ (simulation)	0.0629	0.1417	0.4999	14.03
$\mathcal{H}_\infty$ (reality)	0.004	0.0134	0.2515	62.22

Fig. 22. The transient response of the closed-loop system given step signal, y-axis unit is radian (simulation).

Fig. 23. The transient response of the closed-loop system given step signal, y-axis unit is degree (reality).

lot of uncertainty in the hardware setup, especially in the overshoot performance. In the other hand, the rise, peak, and settling time performance showed up as a superior than the simulation one, it is not absolutely true, because the presence of a noise in raw output reading from encoder – it is as if the system is responsive. Later it will be clarified why was this happened in the next subsection.

Robustness Analysis

Analyzing robustness of closed-loop system is quite complex due to calculation of Structured Singular Value. We use MATLAB `mu` function to calculate it. The concept of Structured Singular Value can be seen as finding a smallest perturbation in terms of largest singular value of perturbation matrix $\Delta \in \mathbb{C}^{q \times p}$ such that the system being perturbed become unstable. The Structured Singular Value is defined as

$$\mu := \frac{1}{\min\{\bar{\sigma}(\Delta) : \Delta \in \mathbf{\Delta}, \det(I - M\Delta) = 0\}} \quad (13)$$

where Δ is a set of perturbation matrix. To facilitate the computation of μ in our design, another special case of LFT framework was used, that is an upper LFT as in “Fig. 24” shown, where the controller had integrated inside the interconnection matrix G . There are six physical parameters in our plant, in our analysis each parameter has its own uncertainty term. The physical parameters are described by

$$\begin{aligned} B_m &= \bar{B}_m + \delta_{B_m} \\ J_m &= \bar{J}_m + \delta_{J_m} \\ K_i &= \bar{K}_i + \delta_{K_i} \\ L_a &= \bar{L}_a + \delta_{L_a} \\ K_b &= \bar{K}_b + \delta_{K_b} \\ R_a &= \bar{R}_a + \delta_{R_a} \end{aligned}$$

where δ represents an uncertainty that assumed $\delta \in [-c, c]$, $c \in \mathbb{R}$.

Recall that, the dynamics in (1) and (2) become.

$$\frac{d\dot{\theta}}{dt} = \frac{1}{\bar{J}_m + \delta_{J_m}} \left(-(\bar{B}_m + \delta_{B_m}) \dot{\theta} + (\bar{K}_i + \delta_{K_i}) i \right) \quad (14)$$

$$M := \left[\begin{array}{ccc|cccc|cc|c} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -\frac{B_m}{J_m} & \frac{K_i}{J_m} & -\frac{1}{J_m} & -\frac{1}{J_m} & \frac{1}{J_m} & 0 & 0 & 0 & 0 \\ 0 & -\frac{K_b}{L_a} & -\frac{R_a}{L_a} & 0 & 0 & 0 & -\frac{1}{L_a} & -\frac{1}{L_a} & -\frac{1}{L_a} & \frac{1}{L_a} \\ \hline 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -\frac{B_m}{J_m} & \frac{K_i}{J_m} & -\frac{1}{J_m} & -\frac{1}{J_m} & \frac{1}{J_m} & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -\frac{K_b}{L_a} & \frac{R_a}{L_a} & 0 & 0 & 0 & -\frac{1}{L_a} & -\frac{1}{L_a} & -\frac{1}{L_a} & \frac{1}{L_a} \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array} \right] \quad (16)$$

Fig. 24. Robust analysis framework.

Then followed by discretization (16), since the equation defined in the continuous time domain.

To get the $G_K - \Delta$ relationship which needs, integrating discrete version of (16) with controller (11), Redheffer Star Product was used, for more information, the reader can refer to [1]. For convenience, we didn't display the resulting matrix, we just define it as G_K , the subscript denotes that the controller was integrated. The MATLAB `mussv` function

$$\frac{di}{dt} = \frac{1}{\bar{L}_a + \delta_{L_a}} \left(-(\bar{K}_b + \delta_{K_b}) \dot{\theta} - (\bar{R}_a + \delta_{R_a}) i + v_{in} \right) \quad (15)$$

After that, we define a new “fictitious output” so that we can “pulling out” the uncertainties δ as in [1]. The “fictitious output” $v_i, i \in \{x \in \mathbb{N} : 1 \leq x \leq 6\}$ is actually correspond to the plant outputs. The following is how “fictitious outputs” are related to the uncertainties.

$$\begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \end{bmatrix} = \begin{bmatrix} \delta_{B_m} & 0 & 0 & 0 & 0 & 0 \\ 0 & \delta_{J_m} & 0 & 0 & 0 & 0 \\ 0 & 0 & \delta_{K_i} & 0 & 0 & 0 \\ 0 & 0 & 0 & \delta_{L_a} & 0 & 0 \\ 0 & 0 & 0 & 0 & \delta_{K_b} & 0 \\ 0 & 0 & 0 & 0 & 0 & \delta_{R_a} \end{bmatrix} \begin{bmatrix} \eta_1 \\ \eta_2 \\ \eta_3 \\ \eta_4 \\ \eta_5 \\ \eta_6 \end{bmatrix}$$

Hence the dynamics in (14) and (15) along with the output matrix can be written as follows.

actually requires the frequency response of G_K , so that it is more convenient to define it as

$$G_K := \begin{bmatrix} G_{K11} & G_{K12} \\ G_{K21} & G_{K22} \end{bmatrix} \quad (17)$$

Robust Stability

Robust stability test only requires the submatrix of G_K , G_{K11} . The theorem in [1] can be said that, a system achieves its robustness in term of stability if G_{K11} is well-posed and internally stable for a specific set of uncertainty. Let say the element is Δ with $\|\Delta\|_\infty < \frac{1}{\beta}$, then the stability is maintained if and only if

$$\sup_{\omega \in \mathbb{R}} \mu(G_{K11}(j\omega)) \leq \beta$$

our result was shown in “Fig. 25”.

Robust Performance

Under robust performance test, we must include an effect from disturbance signals w to our performance signals z . Hence, the uncertainty block is needed to represent them. The

Fig. 25. Structured singular value bounds for robust stability.

Fig. 26. Structured singular value bounds for robust performance.

block is full size, different with the parametric uncertainty which is a diagonal matrix. The theorem in [1] can be said that, a system achieve its robustness in term of performance if G_K is well-posed and internally stable for a specific set of uncertainty, let say the element is Δ with $\|\Delta\|_\infty < \frac{1}{\beta}$, then the performance is maintained if and only if

$$\sup_{\omega \in \mathbb{R}} \mu(G_K(j\omega)) \leq \beta$$

our result was shown in “Fig. 26”.

As a summary, the controller is able to maintain its stability under approximately 95% change of the parameter value, but the controller was failing to maintain its performance since the synthesis process objective bound is not satisfied. That is the reason why the presented performance in reality differs from the theoretical one.

VII. CONCLUSION AND FURTHER WORKS

The fault detection scheme can be concluded that, it is quite robust under a unit step control action, it means under its operation for a specific purpose, the fault detection scheme will not resulting a false alarm, unless a fault is actually present.

The performance presented by the controller in reality and real execution under nominal condition is proficient regarding

to its time performance (e.g. settling time etc.), but the result has overshoot. The $\mathcal{H}_\infty$ controller was able to maintain its stability under specific circumstance, but contrary, with its robustness in term of performance. For future work, it is suggested to design a weighting function to suppress a perturbation effect under particular frequency interval. Also, it is recommended to use μ synthesis technique, since $\mathcal{H}_\infty$ synthesis cannot enclose internal uncertainty in a system. As additional information, the entire code of this paper can be accessed at <https://github.com/koseng-lc/jacl> including the JAACL tools.

This project was funded by Department of Electrical Engineering and Information Technology, Universitas Gadjah Mada. And conducted to fulfill writer bachelor’s degree as a complement to writer’s thesis.

REFERENCES

- [1] K. Zhou, and J. C. Doyle, Essentials of robust control. Upper Saddle River, NJ: Prentice Hall, 1998.
- [2] A. A. Stoorvogel, The H_∞ control problem: a state space approach. NY: Prentice Hall, 1992.
- [3] R. N. Clark, “Instrument Fault Detection,” IEEE Transactions on Aerospace and Electronic Systems, vol. AES-14, no. 3, pp. 456-465, May 1978.
- [4] R. N. Clark, “A Simplified Instrument Failure Detection Scheme,” in IEEE Transactions on Aerospace and Electronic Systems, vol. AES-14, no. 4, pp. 558-563, July 1978.
- [5] Maxon, “S 2322 22 mm, Graphite Brushes, 6 Watt,” 2322.983-11.225-200 model datasheet, Mar. 2020.
- [6] R. J. Patton, P. M. Frank, and R. N. Clark, Issues of fault diagnosis for dynamic systems. Springer Publishing Company, Incorporated, 2010.
- [7] J. Kautsky and N. K. Nichols, “Robust pole assignment in linear state feedback,” Int. J. Control, vol. 41, no. 5, pp. 1129-1155, May 1985.
- [8] K. Zhou, J. C. Doyle, and K. Glover, Robust and optimal control. Upper Saddle River, NJ: Prentice Hall, 1996.
- [9] D. Dolinar, J. Ritonja, B. Grear and K. Jezernik, “Model reference adaptive control of servodrives,” International Conference on Control 1991. Control ’91, Edinburgh, UK, 1991, pp. 98-103 vol.1.
- [10] J. Choi, C. Park, S. Rhyu and H. Sung, “Development and control of BLDC Motor using fuzzy models,” IEEE Conference on Robotics, Automation and Mechatronics, 2004., Singapore, 2004, pp. 1180-1185 vol.2.
- [11] T. Senjyu, H. Kamifurutono and K. Uezato, “Robust speed control of DC servo motor based on Lyapunov’s direct method,” Proceedings of 1994 Power Electronics Specialist Conference - PESC’94, Taipei, Taiwan, 1994, pp. 522-527 vol.1, doi: 10.1109/PESC.1994.349687.