DC Motor Speed and Position Control Using Discrete-Time Fixed-Order H_∞ Controllers

[1]Mohamed A. Darwish, [2]Hossam S. Abbas
[1], Assiut University, Mohamed.darwish@eng.au.edu.eg
[2], Egypt-Japan University of Science and Technology, h_abbas@au.edu.eg

Abstract

This paper describes the design and experimental implementation of a discrete-time fixed-order H_∞ controller for a DC motor speed and position control. To provide a model for the DC motor, two system identification techniques are employed. In the first one a model for DC motor speed control is identified in open-loop based on black box modeling whereas in the other one a model for position control is identified in closed-loop based on grey box modeling. An extension of HIFOO toolbox to discrete-time controller design developed recently is used to synthesize the controller. The performance of the designed controller in comparison with various control strategies is demonstrated. The paper aims at demonstrating simple modeling and control synthesis techniques with the help of available software tools to design low-complexity controllers in terms of design and implementation. Consequently, cheap hardware can be utilized for several applications.

Keywords: *Discrete-Time H_∞ Control, DC Motor Position Control, DC Motor Speed Control, Fixed-Order Controllers, Microcontroller*

1. Introduction

DC motors are widely used in industrial applications, robotic manipulators, etc. PID controllers are commonly used for DC motor speed and position control due to simple structures and comprehensible interpretation of operation. However, designing and tuning PID controllers using conventional methods or with the help of optimization tools [1] might not be adequate to achieve a specified design objectives. Moreover, intelligent control approaches such as Fuzzy Logic Controllers (FLCs) have been used extensively in the literature to control the position of DC motors [2]; however computational complexity [2] may prevent these approaches from being practical in practice. Furthermore, a common problem of all these approaches is that no standard way to specify design objectives in terms of reference tracking, disturbance and noise rejections and control effort constraints. Therefore the design procedures of most of these approaches are usually tedious.

On the other hand H_∞ control framework is well suited for designing optimal and robust controllers for linear time-invariant (LTI) systems. It provides a straightforward methodology to achieve all design objectives that guarantee stability and optimal performance [3]. If no constraints are imposed on the controller order and structure the design problem is convex and can be efficiently solved (e.g. using linear matrix inequalities (LMIs)), but delivers controllers of the same order as the generalized plant [3]. For control applications such as DC motor speed and position control, simple controllers are preferred, as it saves memory and computations efforts in implementation which allow the use of low cost microcontrollers. One can use model order reduction techniques to reduce the order of the controller; however it is not guaranteed that a reduced order controller can achieve the same performance as the full order one (the original controller) or even preserve the stability. Therefore designing low-order controllers directly is preferred in this sense; nevertheless, this makes the synthesis problem non-convex with multiple local minima, which is hard to be solved. Recently, in [4] gradient-based approaches have been proposed for solving the problem in the continuous-time case and readily implemented as a toolbox for Matlab – HIFOO [5].

Nowadays most of controllers are implemented digitally on microcontrollers. Therefore a controller design in continuous-time might be misleading since potential problems like frequency warping [6] are not considered or it might not be possible to impose desired restrictions on the controller, such as properness. As a consequence, an extension of HIFOO to the design of fixed-order discrete-time

doi: 10.4156/ijim.vol1.issue1.1

controllers has been introduced recently in [7], where the whole design is carried out in discrete-time domain.

In this paper a mathematical model of a DC motor is identified experimentally for speed and position control based on black box and grey box modeling, respectively. Then a discrete-time H_∞ first-order controller is designed using the HIFOO toolbox. All the desired system performance specifications are specified in frequency domain that provides a controller which achieves perfect tracking with acceptable control effort and the ability of perfect disturbance as well as high frequency noise rejections. The performance of the controller is compared experimentally with other controllers designed based on different approaches. The approach adopted here gives simple modeling and control synthesis techniques which are able to provide simple controllers that can achieve all desired design objectives without intensive tuning and keep the lowest possible computation effort and storage memory in implementation. This will lead to utilize cheap hardware for several applications.

The paper is organized as follows: Section 2 introduces the modeling approach adopted in this work to model the DC motor for both speed and position control. Section 3 presents the control synthesis technique to control the position and speed of the DC motor, respectively. In Section 4 the experimental results are demonstrated. Finally, conclusions are drawn in Section 5.

2. Modeling

In this section, the modeling approach adopted to identify a mathematical model for the motor is demonstrated. The DC motor used here is an ironless rotor servo motor manufactured by McLennan Servo Supplies Ltd [8]. Its rated torque and speed are 1.5 N.cm and 2500 rpm, respectively.

2.1. Identification of DC motor for speed control

For simplicity the electrical part is neglected, i.e. first order model for DC motor is considered. Identification procedures for linear time invariant (LTI) models could be chosen to estimate grey box model directly in discrete form. In the following the subspace identification method N4SID, provided by Matlab's System Identification Toolbox [9] is used to estimate the parameters A, B, C, and D of a state space model.

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) \\ y(k) &= Cx(k) + Du(k) \end{aligned} \tag{1}$$

where u(k) and y(k) denote the system input and output respectively and x(k) is the state vector of the system at time k.

The data used for identification is a multi sine signal with 1 *HZ* frequency range, 6000 samples, and 0.01 sampling time. The experiment is done in open loop. The obtained data is divided equally in an estimation and a validation part.

The identified discrete-time model is shown below:

$$\begin{aligned} x(k+1) &= 0.9605\,x(k) + 0.0001014\,u(k) \\ y(k) &= 340.3x(k) \end{aligned} \tag{2}$$

To evaluate the estimated model outcomes, the Best Fit Rate (BFR) is used [10]:

$$BFR\% = 100\,\% \,.\, \max(1 - \frac{\|y-\hat{y}\|_2}{\|y-\bar{y}\|_2}, 0)\,, \tag{3}$$

where $\bar{y}$ is the mean of y and $\|\bullet\|_2$ is the 2-norm of a sequence. The BFR measures how much better the model describes the process compared to the mean of the output.

In Figure 1, part of measured vs. simulated outputs is shown for the identified 1st order model. It shows that the model perfectly captures most of the dynamics of the system. The identified discrete-time model here gives BFR of 96.85%.

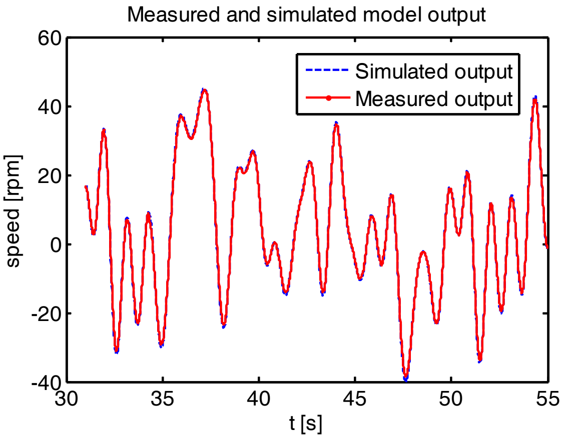


Figure 1. Measured and simulated model output

2.2. Identification of DC motor for position control

Conventionally, a DC motor for position control can be represented by a third order LTI system; however for simplicity it can be represented by a cascaded second order system consists of a first order system and an integrator. This is by neglecting the dynamics of its electrical part which is significantly faster than the mechanical one. Therefore a discrete-time model for the DC motor can be represent by

$$G(z) = \frac{a}{z+b} \cdot \frac{T_s}{z-1}, \tag{4}$$

where T_s is the sampling time, here it is 0.01 sec, a and b are constant parameters of the model. Next the parameters a and b are identified. A gray box modeling is considered based on the closed-loop configuration shown in Figure 2. In this configuration the parameters a and b are estimated with the help of the parameter estimation feature provided by Matlab's Simulink Design Optimization toolbox [11]. This requires gathering a set of input-output data of the system.

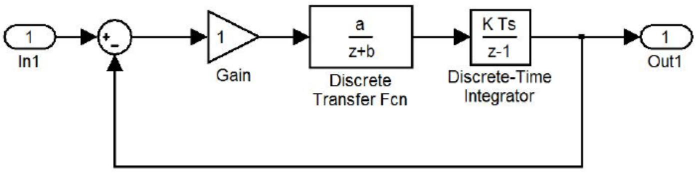


Figure 2. Block diagram for identification of a position control model

Gathering the data is performed in real-time in closed-loop as the open-loop model contains an integrator as shown in Figure 2 where a proportional gain is used to stabilize the closed-loop system. An input signal is designed to excite the dynamics of the DC motor. It is a random multi step signal with 6000 samples with each step has 0.4 sec duration. The obtained data is divided equally into two parts, one part is used for estimation and the other is used for validation. Then with the estimation data the identification process is performed using the Simulink Design Optimization toolbox. Figure 3 shows the convergence of the estimated parameters to their final values. For validation, Figure 4 shows measured versus simulated outputs for the validated part of the data, which has not used in estimation, this demonstrates good estimates. To evaluate the estimated model outcomes, the Best Fit Rate (BFR) criterion is used.

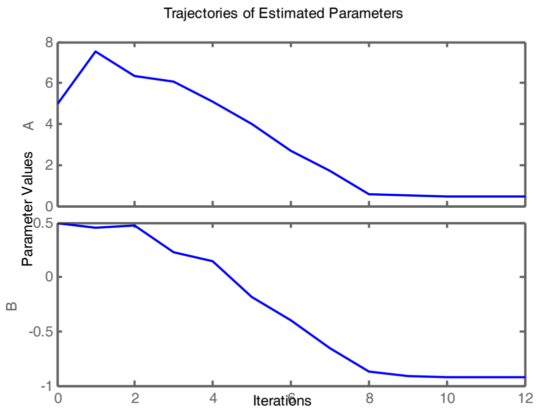


Figure 3. Convergence of the identified parameters to the estimated values

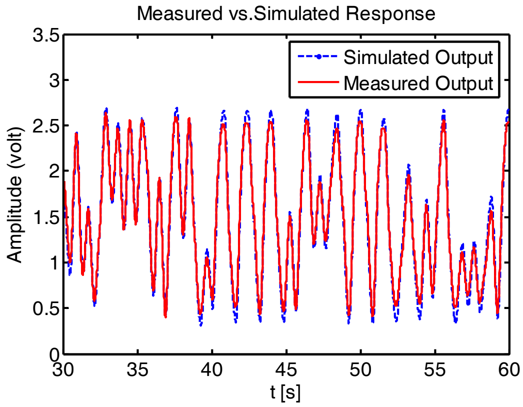


Figure 4. Measured and simulated model output

The identified model gives BFR = 92.65% and the values of a and b are estimated as 0.39855 and -0.93913, respectively. The state-space discrete-time representation of the identified model in reachability form is given by

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \end{bmatrix} = \begin{bmatrix} 1.939 & -0.9391 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} u(k) \qquad (5)$$

$$y(k) = \begin{bmatrix} 0 & 0.003986 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix}$$

where $u(k)$ and $y(k)$ denote the model input and output, respectively, and $x(k)$ is the state vector of the system at time instant k.

3. Control synthesis

In this section H_∞ is introduced for DC motor speed and position control. For position control the classical mixed sensitivity H_∞ design approach is considered in Section 3.1, whereas the four-block H_∞ design approach is used for speed control in Section 3.2.

3.1. Position control

For the control system shown in Figure 5 where $G(z)$ is a given discrete-time LTI model for the DC motor and $K(z)$ is the controller to be designed. H_∞ mixed sensitivity loop shaping design approach is adopted here.

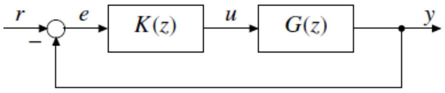


Figure 5. Closed-loop system

For H_∞ design based, a generalized plant is constructed as shown in Figure 6 to translate the closed-loop desired performance.

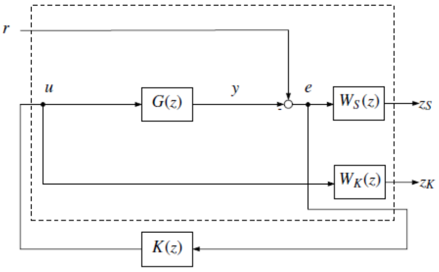


Figure 6. Generalized plant for H_∞ design

The blocks W_S (*z*) and W_K (*z*) shown in Figure 6 indicate weighting filters that are used to shape the closed-loop sensitivity *S* (*z*) and control sensitivity *KS* (*z*) functions, respectively. They are chosen to reflect desired performance of the closed-loop system. The time domain design objectives are given in Table 1. They are translated into the following sensitivity and control sensitivity weighting filters

$$W_S(z) = \frac{0.09}{z-1} \quad \text{and} \quad W_K(z) = \frac{0.1z - 0.0999}{z} \tag{6}$$

Table 1. The control performance objectives for both speed and position control

Rise time (sec)	Settling time (sec)	Overshoot %	E_{ss}	Permissible Voltage
< 1 sec	Minimum	< 5 %	< 1 %	10 v

These filters generate fictitious output signals z_S and z_K, see Figure 6. With the help of *WS* (*z*) and *WK*(*z*), *S* (*z*) and *KS* (*z*), respectively, are shaped in order to achieve desired reference tracking as well as output disturbance rejection, i.e. fast response with small overshoot and small steady state error, to be capable to reject high frequency measurement noise and at the same time to preserve control input limitations.

For more details about loop shaping based design see [12]. The controller design problem can be formulated by the minimization problem

$$\gamma = \min_K \left\| \begin{matrix} W_S S \\ W_K KS \end{matrix} \right\|_\infty \tag{7}$$

where $\left\| \bullet \right\|_\infty$ denotes H_∞-norm of an LTI system. If the minimization problem results a controller *K* (*z*) that yields

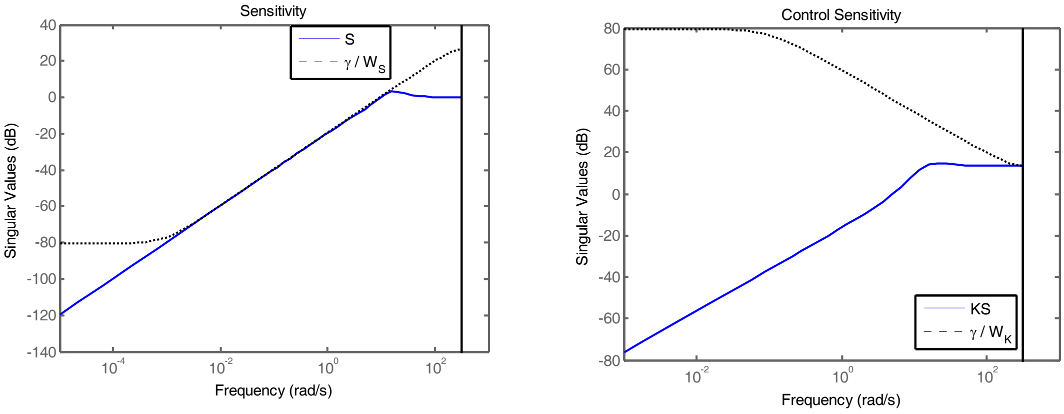
$$\gamma = \min_K \left\| \begin{matrix} W_S S \\ W_K KS \end{matrix} \right\|_\infty < 1$$

the controller *K*(*z*) does satisfy all the required design objectives. If no constraints are imposed on the controller order or structure, problem (7) can be expressed as a minimization of a linear cost function subject to linear matrix inequality (LMI) constraints which can be efficiently solved by the available LMI solvers [13]. This produces a controller with order equals the order of the generalized plant (here its is a fourth order). In this work it is desired to achieve all the design objectives via a first order controller, therefore problem (7) can only expressed as an optimization problem subject to bilinear matrix inequality (BMI) constraint which is a non-convex optimization problem that is difficult to solve. Different direct optimization techniques and iterative LMI techniques have been introduced in the literature [14][15] to attack this problem with varying success and performance. The main difficulties of these approaches is that the optimization problem formulated for controller design consists of a large number of decision variables, this makes the optimization algorithms slow and cannot achieve good local minimum.

Based on the relation between "complex stability radius" and H_∞ norm of transfer functions, a gradient based optimization technique has been developed in [4][5] to design a fixed-order H_∞ controllers where the number of decision variables are significantly reduced and consequently faster and better solutions can be obtained. The proposed technique has been offered as a free toolbox named HIFOO, which has been developed with Matlab. Recently this approach has been extended in [7] for H_∞ discrete-time design. For the DC motor position control, consider the same problem formulated in (7) with design objectives interpreted in frequency domain in terms of the shaping (weighting) filters in (6). The generalized plant shown in Figure 6 is constructed. A first-order discrete-time H_∞ controller is desired. Next, HIFOO toolbox is used to solve problem (7) to obtain the controller. This results controller

$$K(z) = \frac{4.504Z - 4.297}{Z - 0.8579} \tag{8}$$

It is worth to mention that the minimum value of H_∞ norm, in (7) with the controller (8) is $\gamma = 0.9433$. This shows that all the required specifications in terms of the filters in (6) have been achieved. For further demonstrations, the singular value plots of the sensitivity and control sensitivity with the inverse of the corresponding weighting filters are plotted in Figure 7. These plots emphasize that the required design objectives in frequency domain have been met.



(a) (b)

Figure 7. Sensitivity (a) and control sensitivity (b) plots with inverse of weighting filters

3.2. Speed control

Next, we discuss the design strategy for DC motor speed control. Now, the design objectives include reference tracking as well as input disturbance rejection, see Figure 8. These design objectives are in contradiction over all frequencies. Therefore, a balanced trade-off between them cannot be found using the classical approach discussed in the previous section [16]. In order to achieve both good reference tracking and input disturbance rejection a standard four-block mixed sensitivity approach is employed here. Consider the generalized plant shown in Figure 9 where V_r(s) and V_d(s) are pre-filters for reference and disturbance signals, respectively.

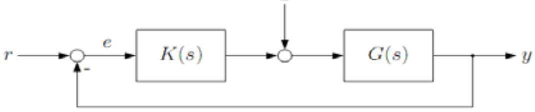


Figure 8. Closed loop system with input disturbance

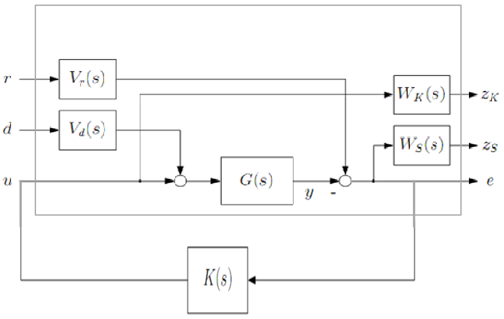


Figure 9. Four-block generalized plant for mixed-sensitivity design.

Now in the H_∞ design, the filters $W_S(s)$, W_K(s), $V_r(s)$ and $V_d(s)$ all are used to shape the transfer functions *S, KS, SG* and *KSG*, see Figure 9. As a consequence, the synthesis problem is expressed as

$$\gamma = \min_K \left\| \begin{matrix} W_S S V_r & W_S S G V_d \\ W_K K S V_r & W_K K S G V_d \end{matrix} \right\|_\infty . \tag{9}$$

The required design objectives will be achieved if a controller *K* which minimizes the H_∞ norm in (9) is found such that

$$\gamma = \left\| \begin{matrix} W_S S V_r & W_S S G V_d \\ W_K K S V_r & W_K K S G V_d \end{matrix} \right\|_\infty < 1 \cdot$$

For the DC motor speed control the identified model in Section 2.1 is used, see (2). Again the design objectives shown in Table 1 are considered. The shaping filters as well as the prefilters are tuned to reflect these design objectives and expressed as follows:

$$W_S(z) = \frac{0.2}{z-1},\ W_K(z) = \frac{0.01z - 0.00999}{z}\ ,\ V_r = 2 \text{ and } V_d = 4 \tag{10}$$

The generalized plant in Figure 9 is constructed with the filters in (10). Again, HIFOO toolbox is used to solve problem (9) to obtain the controller. This produces the discrete-time controller given as

$$K(z) = \frac{15.17(Z - 0.88717)}{Z - 1} \tag{11}$$

The minimum value of the H_∞ norm of (9) with the controller (11) is 0.8164, which proves that all the design objectives have been attained. The singular value plots of the shaped functions *S, KS, SG* and *KSG* and simulation result are shown in Figure 10, which confirm the obtained results.

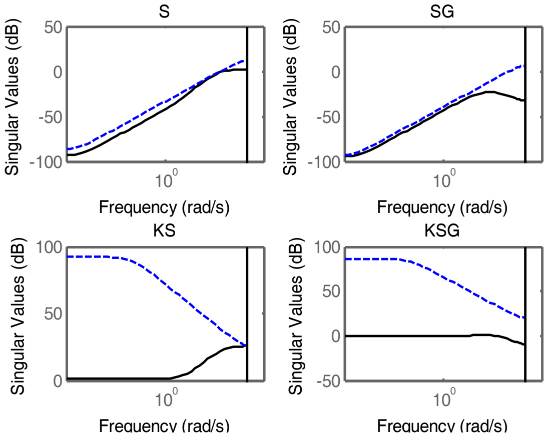


Figure 10. Singular values plots of *S, KS, SG* and *KSG* and their weighting functions

4. Experimental results

This section describes the experimental setup and the implementation of the first-order H_∞ controller designed in the previous section for both speed and position control, and a comparison of results with a full-order H_∞ controller, reduced order of the full-order one, an optimized PID controller and a Fuzzy Logic Controller (FLC), all designed based on the derived model in Section 2.

4.1. Experimental setup

The experimental setup used here is the analogue servo trainer CA06 manufactured by LJcreate Instrumentation Ltd [17], see Figure 11 for the hardware setup, its block diagram is shown in Figure 12. The experimental setup consists of the DC motor and a potentiometer fixed on the motor shaft to measure the angular position. The microcontroller PIC16F877A is used to implement digital controllers [18]. It runs at an internal clock speed of 5 MHz with 200 nsec instruction cycle. The microcontroller receives the measured position from the potentiometer and the measured speed from the tacho-generator via a built in A/D converter. An external D/A converter has been built to convert the digital output from the microcontroller into analogue voltage.

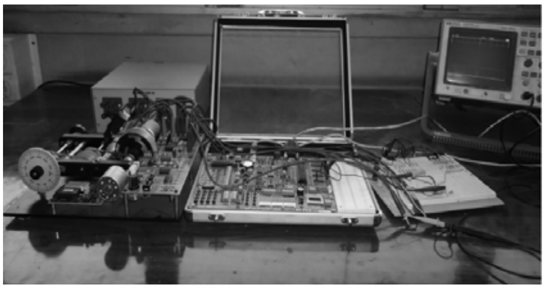


Figure 11. Hardware setup

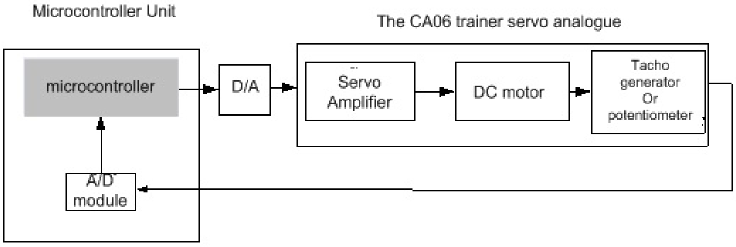


Figure 12. Block diagram of the hardware setup

4.2. Position controller implementation

Next the capability of the DC motor to track a position step reference of 132 degree in closed-loop is tested with different controllers as follows: In addition to the first-order H_∞ controller designed in Section 3 with the help of HIFOO toolbox, a full-order (fourth-order) H_∞ controller has been designed based on the model identified in Section 2 with the help of Matlab command **dhinflmi** with the same weighting functions in (6) in order to consider the same deign objectives. Moreover, the order of the full-order controller has been reduced with the help of Matlab command **balred** to provide simpler controller for comparison. It results a third-order one. It is worth to mention that it is not possible to further reduce the order of this controller, otherwise it brings unstable closed-loop. An FLC controller has been designed as well based on Mamdani architecture for fuzzy control [19], for more details about this controller see [20]. Moreover, an optimized PID controller based on the identified model has been designed with the help of Matlab SISOtool. The performances of the DC motor to track the step reference with all designed controllers are demonstrated in Figure 13. In terms of rise time, all controller achieve the desired time. The FLC controller gives the highest overshoot and the optimized PID controller gives the highest settling time and both violate the desired design objectives whereas the

H_∞ based controllers achieve the desired performance and all have almost the same performance, see Table 2.

Table 2. Response performance of position control

	Optim. PID	FLC	4th order H_∞	1st order H_∞	4th order H_∞
Rise time (sec)	0.69	0.36	0.49	0.4	0.52
Settling time (sec)	6.33	0.94	0.72	0.47	0.7
Overshoot%	7.27	17.95	0	0	0
Max. control signal (v)	8.81	3.31	8.775	8.931	8.853

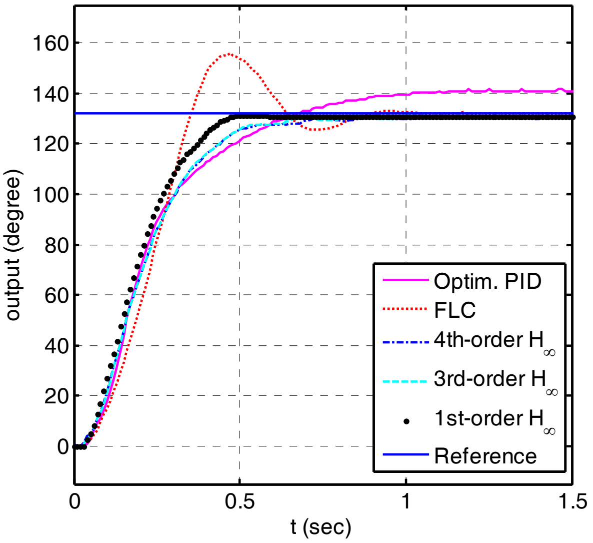


Figure 13. Tracking performance of all controllers for position control

4.3. Speed controller implementation

Again, the capability of the DC motor to track a speed step reference of 13.966 rad/sec and to reject a sudden disturbance applied for 4.5 sec is compared and tested with different controllers as follows: In addition to the first-order H_∞ controller designed in Section 3 with the help of HIFOO toolbox, a full-order (third-order) H_∞ controller has been designed based on the model identified in Section 2 with the help of Matlab command **dhinflmi** with the same weighting functions in (10) in order to consider the same deign objectives. Moreover, the order of the full-order controller has been reduced with the help of Matlab command **balred** to provide simpler controller for comparison. It results a first-order one. Also, A FLC and an optimized PI controller are designed as in the case of position control.

These controllers have been designed to achieve the same objectives given in Table 1. The tracking performance of the PI, FLC, full order H_∞, reduced order H_∞ and the fixed-order H_∞ controllers are shown in Figure 14. The disturbance rejection ability and controller voltage in response to the step reference in case of the first order fixed-order H_∞ controller based on HIFOO are shown in Figure 15.

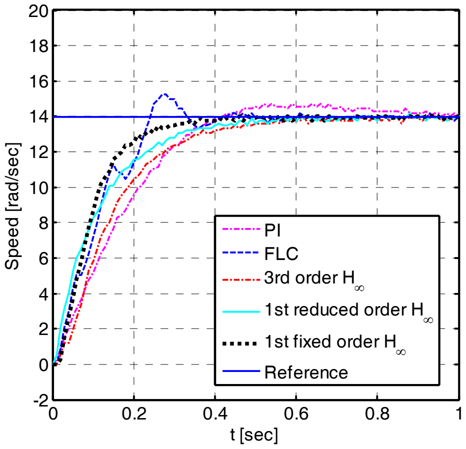
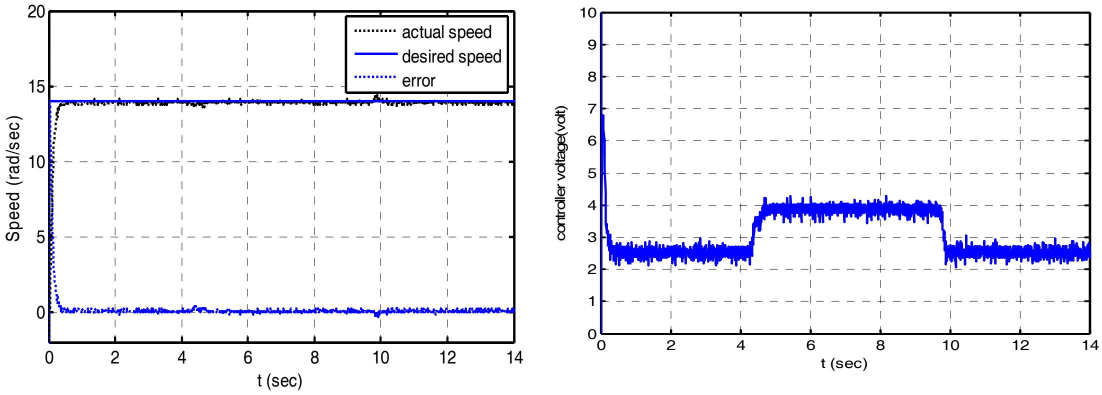


Figure 14. Tracking performance of all controllers for speed control



(a) (b)

Figure 15. First order fixed-order controller synthesis HIFOO: (a) Reference tracking and disturbance rejection (b) Control signal

Table 3 summarizes the tracking performance of the implemented controllers. It is clear that the fixed-order H_∞ controller achieves the best settling time among the others, whereas the FLC leads to the smallest rise time in the expense of the highest overshoot. On the other hand, the PI controller provides the smallest control signal. We conclude that the fixed order H_∞ controller gives the best compromise of the control objectives among the others.

Table 3. Response performance of speed control

	PI	FLC	3rd order H_∞	1st order H_∞ Fixed-order	1st order H_∞ Reduced
Reference tracking specification					
rise time (sec)	0.42	0.24	0.56	0.32	0.46
Settling time (sec)	0.98	0.46	0.61	0.38	0.55
Overshoot%	4.29	9.16	0	0	0
Max. control signal (volt)	2.8	9.7	7.878	6.747	7.332
Disturbance rejection specification					
Overshoot%	18.11	4.4	4.6	4	4.4
Time to remove the disturbance (sec)	0.77	0.3	0.32	0.29	0.37

For further demonstration, the performance of the DC motor with the first-order H_∞ controller to track a multi step speed and position reference is shown in Figure 16 and Figure 17, respectively. It shows that the controller gives almost the same performance regardless the value of the step reference.

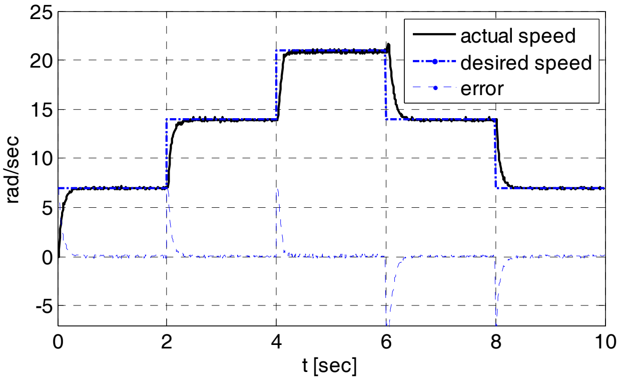


Figure 16. Multi step speed reference tracking with the first-order H_∞ controller

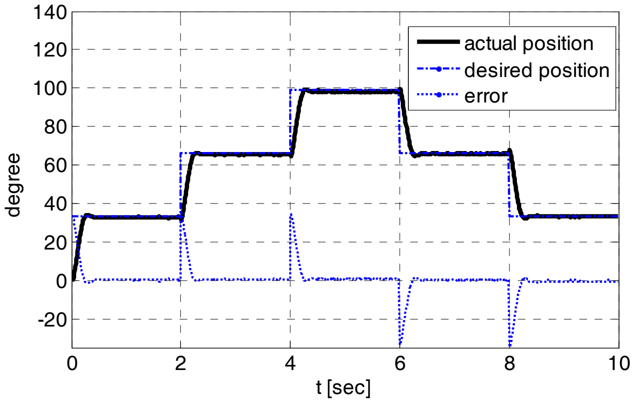


Figure 17. Multi step position reference tracking with the first-order H_∞ controller

4.4. Practical issues

In this section we demonstrate the performance of the microcontroller unit in terms of the storage memory and execution time requirements when it implements the algorithms of the PID, FLC and the H_∞ controllers.

Table 4 shows, the Ram, Program memory and execution time required to implement the mentioned algorithms in one sampling interval.

Table 4. Microcontroller resources usage

Algorithm	Ram (Byte)	Program memory word (KB)	Execution time (msec)
PID	69.92	1.44	1.9
FLC	69.92	3.36	6.2
3rd order H_∞	88.32	1.6	2.2
1st order H_∞	66.24	1.28	1.6

It is important to mention that the performance of the optimized PID or the FLC controllers could be improved further; however it is tedious to tune both as there is no standard or systematic procedures for that. Whereas the synthesis of H_∞ based controllers approaches provide a systematic way to express desired design objectives directly as an optimization problem and to design the controller with few tuning parameters. This is in addition to the availability of efficient tools which result optimal controllers. On the other hand, the main advantage of the first-order H_∞ controller is that its ability to provide all the desired design objectives with low complex structure. Table 4 demonstrates the performance of the microcontroller unit in terms of the Ram, Program memory and execution time required to implement the algorithm of all considered controllers. It is shown that the first-order H_∞ controller demonstrates the lowest microcontroller resources usage, namely, Ram, Program Memory and execution time. It requires the least RAM, Program Memory and minimum execution time among the other controllers. It consists of just a first-order difference equation that can be implemented easily on a microcontroller which has very low specifications. This means that the current microcontroller (PIC16F877A) could be replaced by a simpler one such as PIC16F876A that costs half the price of PIC16F877A. Therefore, one can claims that first-order H_∞ controller is the best among the others as it achieves all the required objectives with the simplest structure.

5. Conclusion

The modeling and design of a fixed-order discrete-time H_∞ controller for a DC motor speed and position control has been demonstrated with available modeling and control synthesis tools. A comparison with other control algorithms has been carried out experimentally. The results show that the first-order H_∞ controller has attained all desired performance with the simplest hardware requirements for implementation. The design has been executed with the help of the recently developed HIFOO toolbox. The main features of the design are that it gives more flexibility to the designer to specify generally design objectives to design very simple controllers provided a good model for the plant.

6. References

[1] N. Thomas and D. Poongodi, “Position control of DC motor using genetic algorithm based PID controller,” in Proceedings of the World Congress on Engineering, London, U.K., 2009, p. Vol II.

[2] N. Amlashi, “Design and implementation of fuzzy position control system for tracking applications and performance comparison with conventional pid,” IAES International Journal of Artificial Intelligence, vol. 1, no. 1, pp. 31–44, 2012.

[3] S. Boyd, L. E. Ghaoui, E. Feron, and V. Balakrishnan, Linear Matrix Inequalities in System and Control Theory. SIAM, Philadelphia, 1994, PDF.

[4] J. Burke, D. Henrion, A. Lewis, and M. Overton, “Stabilization via nonsmooth, nonconvex optimization,” IEEE Transactions on Automatic Control, vol. 51, no. 11, pp. 1760–1769, 2006.

[5] J. Burke, D. Henrion, A. Lewis, and M. Overton, “HIFOO - a Matlab package for fixed-order controller design and H_∞ optimization,” in IFAC Symposium on Robust Control Design, Toulouse, France, 2006.

[6] K. Astr¨om and B. Wittenmark, Computer-Controlled Systems: theory and design, 3rd ed. Upper Saddle River, NJ: Prentice-Hall, 1997.

[7] A. Popov, H. Werner, and M. Millstone, “Fixed-structure discrete-time H_∞ controller synthesis with HIFOO,” in Proc. 49th IEEE Conference on Decision and Control, Atlanta, Georgia, USA, 2010.

[8] Low cost 2-12 Watt ironless rotor servo motors, Mclennan Servo Supplies Ltd.

[9] The Mathworks, System Identification Toolbox User’s Guide, The Mathworks, Inc., Natick, 2004.

[10] L. Ljung, System Identification Toolbox 7 Getting started guide, 2010.

[11] Simulink Design Optimization Toolbox. The MathWorks, 2010.

[12] D. Gu, P. Petkov, and M. Konstantinov, Robust Control Design with MATLAB. Springer Verlag, London, 2005.

[13] G. Balas, R. Chiang, A. Packard, and M. Safonov, Robust Control Toolbox for Use with Matlab. The MathWorks, 2010.

[14] A. Hassibi, J. How, and S. Boyd, “A path-following method for solving BMI problems in control,” in Proc. of the American Control Conference, 1997, pp. 1385–1389.

[15] H. Abbas, S. Chughtai, and H. Werner, “A hybrid gradient-LMI algorithm for solving BMIs in control design problems,” in Proc. 17th IFAC World Congress, Seoul, South Korea, 2008, pp. 14 319–14 323.

[16] Yu Huang, Timo Pe, Andrey P. Popov, Herbert Werner and Frank Thielecke, "Control of a Two-Load-Path Trimmable Horizontal Stabilizer Actuator of an Aircraft – Comparison of H_∞ Design Approaches ", 49th IEEE Conference on Decision and Control, USA, December, 2010.

[17] Analog and Digital Motor Control, Curriculum Manual CA06, LJcreate Ltd.

[18] PIC16F87X Data Sheet 28/40-Pin 8-Bit CMOS FLASH Microcontrollers, Microchip Technology Incorporated, USA,, 2001.

[19] J. Yen and R. Langarig, Fuzzy Logic Intelligence, Control, and Information. Prentice Hall, 1999.

[20] M. Darwish, H. Abbas, A. Saleh, and M. Hassan, “FLC implementation on a 8-bit microcontroller for DC motor speed and position control,” Journal of Engineering Sciences, Assiut University, vol. 39, no. 2, pp. 405–423, 2011.