DC Motor Servo Controller Synthesis and Robustness Analysis

Yingxin Wei
Department of Electrical and Computer Engineering
University of Minnesota
wei00120@umn.edu

Xiang Zhang
Department of Electrical and Computer Engineering
University of Minnesota
zhan6668@umn.edu

***Abstract*—In this project, the physical system that we study is the servo controller for the DC motor. We synthesize a Proportional-Integral (PI) controller using root locus, and two controllers using optimal control techniques (LQR, H_∞). We model the uncertain systems and assess robust stability and robust performance. Eventually we manage to synthesize three kinds of controllers and each controller has its own strengths.**

***Index Terms*—Servo Controller, DC Motor, Proportional-Integral (PI), Optimal Control, Linear-Quadratic Regulator (LQR), H_∞ Robust Control Design.**

I. Introduction

A. DC Motor Model

As shown in Figure 1, we select a DC motor model modified from a demo provided by a MATLAB document [1]. A DC motor belongs to a class of rotary electrical motors that converts direct current electrical energy into mechanical energy. In armature-controlled DC motors, the applied voltage V_a controls the angular velocity ω of the shaft [3]. The goal of the controller is to provide tracking to step changes in reference angular velocity [1].

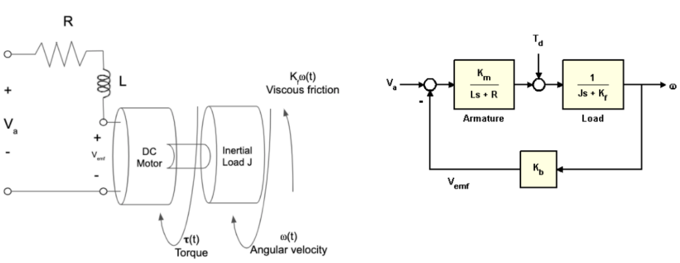

Fig. 1. Selected DC motor and the simplified model.

In our motor model, the initial physical constants are:

- $R = 2.0\ Ohms$
- $L = 0.5\ Henrys$
- $K_m = 0.1\ Torque\ constant$
- $K_b = 0.1\ Back\ emf\ constant$
- $K_f = 0.2\ Nms$
- $J = 0.02\ kg\ m^2/s^2$

We also have a Simulink model for the DC motor as shown in Figure 2. [1].

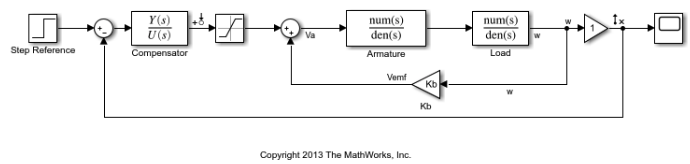

Fig. 2. Simulink model for the chosen DC motor.

B. DC Motor Model with Parameter Uncertainty and Unmodeled Dynamics

Based on the DC motor model introduced above, as we are inspired by the topics about uncertainty and optimal control in class, we decide to add parameter uncertainty and unmodeled dynamics so that we can investigate the robustness of the system [2]. We aim to synthesize three kinds of controllers and evaluate the margins as well as the disturbance rejection quality of three corresponding systems.

C. Controllers for armature-controlled DC motors

Different DC motor control techniques can be used to track setpoint commands and reduce sensitivity to load disturbances [3]. In order to reduce the sensitivity of ω to load variations (changes in the torque opposed by the motor load), we propose three possible control techniques that we can apply, which are: Proportional-Integral (PI) control, Linear-Quadratic Regulator (LQR) control, and H_∞ control.

A PI controller is a feedback control loop that calculates an error signal by taking the difference between the output of a system and the set point, whose output power equals the sum of proportion and integration coefficients [4] [5]. An LQR controller is often designed by using a mathematical algorithm that minimizes a cost function with weighting factors supplied by designers. H_∞ methods are used in control theory to synthesize controllers to achieve stabilization with guaranteed performance. To use H_∞ methods, a control designer expresses the control problem as a mathematical optimization problem and then finds the controller that solves this optimization [6].

II. Methods and Analysis

With the initial constant parameters, taking this DC motor as our plant, we are able to use traditional methods to generate a PI controller and also apply optimal control techniques such as LQR and H_∞. We choose to add uncertainty in the parameters R, L, K, K_f, based on which we test and analyze our synthesized controllers.

The nominal values of K_f and K_b are 0.015 with a range between 0.012 and 0.019. The resistance (R) and inductance (L) constants range within $\pm 40\%$ of their nominal values. We set the nominal value of the rigid body inertia (J) to 0.02 and we include 15% dynamic uncertainty in multiplicative form.

A. Controller Synthesis

PI Control: The proportional gain K_p should be set to the reciprocal of the DC gain from V_a to ω. To enforce zero steady-state error, we also use integral control of the form $\frac{K_i}{s}$. To determine the integral gain K_i, we use the root locus technique applied to the open-loop $\frac{1}{s} * transfer(V_a \rightarrow \omega)$. Then we get the transfer function of the PI controller: $\frac{4.10s+5}{s}$.

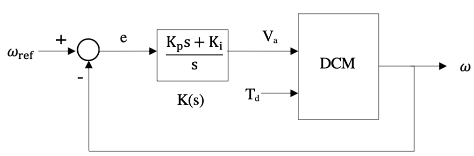

Fig. 3. Block diagram of applying PI control.

LQR Control: In addition to the integral of error, the LQR scheme also uses the state vector $x = (i, \omega)$ to synthesize the driving voltage V_a [2]. The resulting voltage is of the form

$$V_a = K_1 * \omega + K_2 * \frac{\omega}{s} + K_3 * i \quad (1)$$

where i is the armature current. For better disturbance rejection, we use a cost function that penalizes large integral error, e.g., the cost function

$$C = \int_0^\infty (20q(t)^2 + \omega(t)^2 + 0.1V_a(t)^2)dt \quad (2)$$

where $q = \frac{\omega(s)}{s}$. For this cost function we can compute the optimal LQR gain in MATLAB.

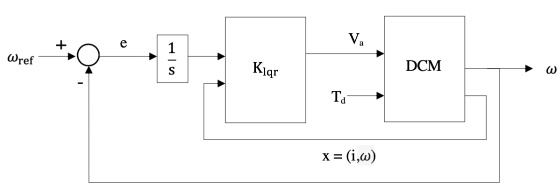

Fig. 4. Block diagram of applying LQR control.

H_∞ **Control**: We design our structure using the H_∞ method and apply it to the DC motor as shown in Figure 5. This is a simplified model, where noise is neglected. The model is constructed using the DC motor and the ideal controller. The tools in **Control System Designer** in MATLAB can help design a controller for the nominal plant model while simultaneously visualizing the effect on the other plant models [1]. Given a properly set up structure, we can solve the controller directly using *hinfsyn* in MATLAB. The H_∞ synthesized controller almost behaves the same as the ideal controller generated by the tools in **Control System Designer** in MATLAB.

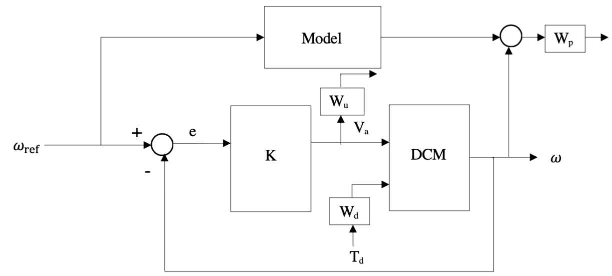

Fig. 5. Block diagram of applying H_∞ control.

B. Robustness Analysis

We draw the Bode plot for the forward transfer function using the PI, LQR, and H_∞ control together with the gain and phase margins in MATLAB as shown in Figure 6, 7, 8. For analysis, we focus on the margins and sensitivity.

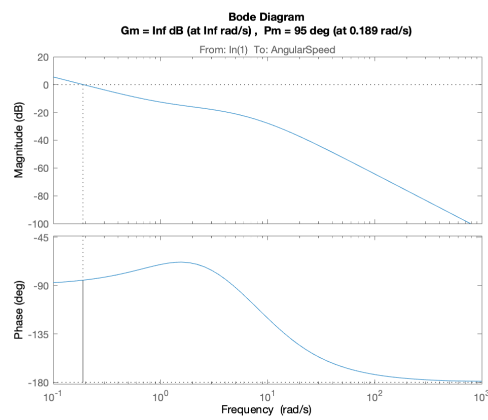

Fig. 6. Bode plot of the system using PI control.

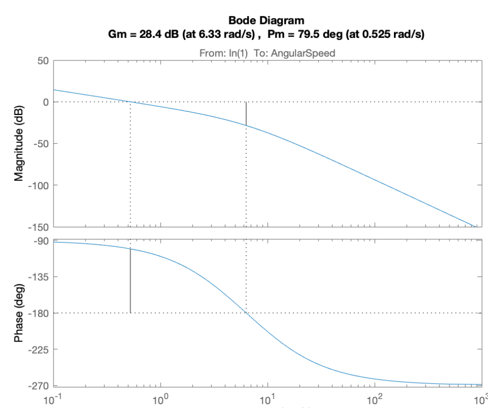

Fig. 7. Bode plot of the system using LQR control.

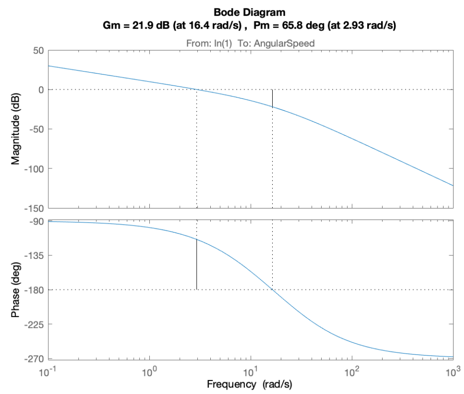

Fig. 8. Bode plot of the system using H_∞ control.

We collect the output of gain margins, phase margins, disk margins, and the worst case disk margins into a table which enables us to do a straightforward comparison. We specifically use worst-case analysis to find out how bad the margins can really get. The *wcdiskmargin* function computes the worst-case gain and phase margins for the modeled uncertainty [2].

TABLE I
COLLECTION OF THE MARGINS

	PI	**LQR**	H_∞
Gain Margin[a] (nominal)	Inf	26.4135	12.4154
Phase Margin[b] (nominal)	95.0340	79.5294	65.7794
Disk Margin (nominal)	1.9198	1.4870	1.1271
Disk Margin (worst case)	1.4677	0.4203	0.1358

[a] Unit for gain margin: magnitude.
[b] Unit for phase margin: degree.

From Table I we can conclude that:

- Nominal case gain margin: $PI > LQR > H_\infty$;
- Nominal case phase margin: $PI > LQR > H_\infty$;
- Nominal case disk margin: $PI > LQR > H_\infty$;
- Worst case disk margin: $PI > LQR > H_\infty$.

Next, to study the disturbance rejection, we further look at the plots about Sensitivity, as shown in Figure 9 and Figure 10. From these samples, we get the worst case and analyze the results in the following parts regarding system bandwidth, frequency domain performance, and time domain performance.

1) Bandwidth of the system:
   - Worst case: $H_\infty > PI > LQR$.
   - Nominal case: $H_\infty > LQR > PI$
2) Frequency domain:
   Worst case:
   - Max gain: $PI < LQR < H_\infty$
   - Low frequency: $H_\infty < LQR < PI$
   - High frequency: Almost the same

   Nominal case:
   - Max gain: Almost the same
   - Low frequency: $H_\infty < LQR < PI$
   - High frequency: Almost the same

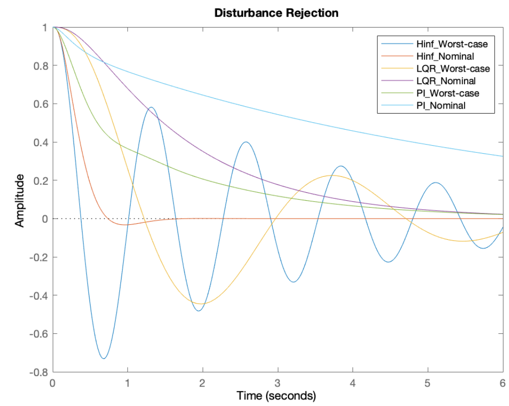

Fig. 9. Nominal and worst-case rejection of a step disturbance.

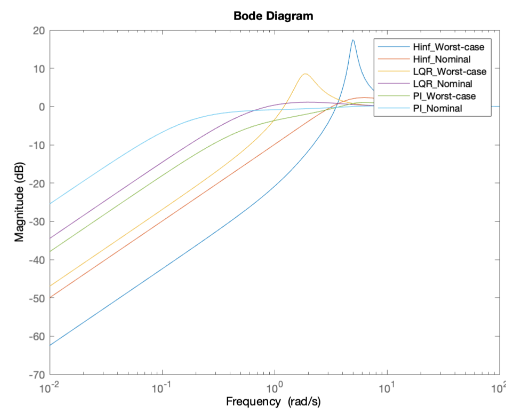

Fig. 10. Magnitude of nominal and worst-case sensitivity.

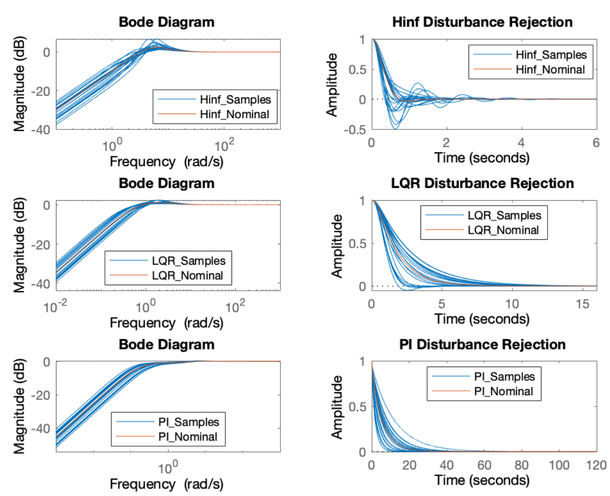

Fig. 11. Samples of Sensitivity.

3) **Time domain**:
   - Worst case settling time: $PI < LQR < H_\infty$
   - Nominal case settling time: $H_\infty < LQR < PI$

We can see from Figure 9 to 11 that generally the H_∞ controller performs better than LQR and PI controllers for the system with uncertainty. The implementation of our designed H_∞ controller demonstrates substantial improvements in bandwidth, worst case maximal gain, and settling time, while eliminating some undesirable effects.

III. Discussion

In this project, we only test limited parameters and build these three controllers based on them, which means the results do not necessarily represent the overall performance of these three kinds of controllers. It is very possible that we can obtain better controllers by changing the weights when designing the LQR or H_∞ controller. When evaluating the robustness, we can also try simulating the motor with more load added.

Moreover, so far we have been only testing around the change of magnitude. It is worthwhile to play with the phase side such as adding nonlinear dynamics like lag/lead to the system and observe the outcome, given sufficient background knowledge about these topics.

In addition, we can alternatively try different kinds of uncertainty such as setting different ranges for the original chosen parameters or turning to another set of parameters to work on. As our controllers are only demonstrating the performance under limited conditions, there are many interesting aspects that we can discuss.

IV. Conclusion

We have successfully synthesized a PI controller, an LQR controller, and an H_∞ controller for our nominal DC motor model and analyzed the system with uncertainty. Their performances are compared in multiple dimensions, and the H_∞ controller stands out with the best robustness in the aspect of disturbance rejection. However, we also notice that its higher order nature brings higher cost, which makes it unable to be the first choice for some small systems. Besides, we have some trade-off choosing different controllers to apply. For instance, PI control, with the largest stability margins among these three, performs the best in disturbance rejection in the worst case, but the worst in disturbance rejection in the nominal case. H_∞ control, with the smallest stability margins, performs the worst in disturbance rejection in the worst case, but the best in disturbance rejection in the nominal case. The LQR controller has the medium performance in almost every aspect among these three, which can be a good choice if we desire a balanced performance.

Acknowledgment

Both authors contributed equally to this project. MATLAB code and published file of this project are attached.

References

[1] "Reference Tracking of DC Motor with Parameter Variations," Accessed on: April. 5, 2020. [Online]. Available: https://www.mathworks.com/help/control/ug/reference-tracking-of-a-dc-motor-with-parameter-variations.html

[2] "Robustness of Servo Controller for DC Motor," Accessed on: April. 5, 2020. [Online]. Available: https://www.mathworks.com/help/robust/examples/robustness-of-servo-controller-for-dc-motor.html

[3] DC motor LQR demo in MATLAB

[4] "P.I Controller," Accessed on: May. 4, 2020. [Online]. https://muse.union.edu/seniorproject-menesese/implementation/

[5] "P, PI and PID control," Accessed on: May. 4, 2020. [Online]. https://www.ao-tera.com.ua/list/us/technology/0/246.html

[6] "H-infinity methods in control theory," Accessed on: May. 4, 2020. [Online]. https://en.wikipedia.org/w/index.php?title=H-infinity_methods_in_control_theory&oldid=887521892