

AE353: Design Problem 02

(Control of NASA Robonaut)

T. Bretl

This paper describes the second design problem that you will complete in AE353 (Spring 2020). It asks you to design, implement, and test a controller for NASA Robonaut B (<http://spaceflight.nasa.gov/gallery/images/station/eva/html/jsc2005e11678.html>), and to submit a report that describes both your method of approach and your results.

I. Nomenclature

ζ	=	horizontal position of robot wheel center in meters
θ	=	angle of robot body in radians with respect to vertical
τ	=	torque applied by robot wheel to robot body in Newton-meters
τ_{\max}	=	maximum torque

II. Goal

The code provided in `DesignProblem02.m` simulates the motion of NASA Robonaut B (“the robot”) in MATLAB. This robot has two parts, a wheel and a body. It has one actuator, a motor, that is mounted on the wheel and that can apply a torque on the body. The robot also has sensors that can measure the horizontal position and velocity of the wheel center, as well as the angle and angular velocity of the body with respect to the vertical. The goal is to make the wheel center track some non-constant reference trajectory with little-to-no steady-state error, while keeping the body upright (i.e., off the ground).

III. Model

If the robot were rolling on flat ground, then its motion would be governed by ordinary differential equations with the form

$$\begin{bmatrix} \ddot{\zeta} \\ \ddot{\theta} \end{bmatrix} = f(\zeta, \theta, \dot{\zeta}, \dot{\theta}, \tau) \quad (1)$$

where ζ is the horizontal position of the wheel center, θ is the angle of the body from the vertical (in radians), and $\tau \in [-\tau_{\max}, \tau_{\max}]$ is the torque applied to the body by a motor on the wheel. The function f in (1) depends on a number of parameters (e.g., lengths, masses, moments of inertia). You can access a symbolic description of this function either within your controller code (`Controller.m`) as `parameters.symEOM.f`, or in a separate piece of code that you write for the purpose of control design. In particular, if you call the simulator with the optional parameter `'eomfile'`, then you can save the equations of motion to a file for later analysis. Here is an example:

```
1 % Run the simulator to save the equations of motion
2 DesignProblem02('Controller', 'eomfile', 'eom.mat', 'display', 'false');
3 % Load the equations of motion
4 load('eom.mat');
5 % Parse the equations of motion
6 f = symEOM.f;
7 % Define symbolic variables that appear in the equations of motion
8 syms zeta zetadot theta thetadot tau real
9 % Proceed to linearize or do whatever else you like...
10 % (see: help sym/jacobian, help sym/diff, help sym/subs, etc.)
```

Unfortunately for you, the robot is rolling on ground that is not flat. Indeed, you will have no way of knowing exactly what the shape of the ground is, and so will have to design a controller that somehow compensates for this.

IV. Tasks

A. Requirements

A *requirement* is a property that the system you are designing must have in order to solve your problem (i.e., a thing that needs to get done). A good requirement is quantifiable—it involves a number that must be within a certain range in order to solve your design problem. A good requirement is also both relevant (it *must* be satisfied—it is not optional) and detailed (it can be read and understood by anyone). Here is an example of a requirement that says what needs to get done but that most engineers would consider unacceptable:

The wheel center shall track a reference trajectory.

This requirement is not detailed. One way to improve it would be to say what reference trajectory, and to say what it means to track this trajectory:

The wheel center shall remain close to a constant reference value.

This requirement is not quantifiable. One way to improve it would be to say how close:

The wheel center shall remain within ± 0.1 meters of a constant reference value.

Most engineers would argue that this requirement still needs improvement. How long must the wheel center remain close to the reference value? Is there an upper bound on the time it must take for the wheel center to get close to the reference value, assuming it starts at some other value? Must this requirement be satisfied for any choice of reference value, or just values within a specific range? Must this requirement be satisfied no matter what the initial conditions are, or no matter what the shape of the ground is? Or, are there particular operating conditions within which the requirement applies? These are examples of reasonable questions that might be asked by an engineer reviewing the requirement. Your task is to define *one* requirement—that is quantifiable, relevant, and detailed—that makes clear what the system you are designing must do in order for your goal to be achieved.

B. Verifications

A *verification* is a test that you will perform to make sure that the system you are designing meets a given requirement. A good verification is based on a measurement—it checks that a quantity is in the range specified by the requirement. A good verification also has a set of instructions for how to make the measurement (an experimental protocol) and for how to interpret the results (methods of data analysis and visualization that provide evidence the requirement has been met). Consider the requirement from Section IV.A (which, as we have said, still needs improvement):

The wheel center shall remain within ± 0.1 meters of a constant reference value.

Here is a verification of this requirement that most engineers would consider unacceptable:

The system will be tested in simulation.

This verification is not based on a measurement. Here is a better version that *is* based on a measurement:

The error between the wheel center and the constant reference value will be found in simulation.

This verification does not include a set of instructions for how to make the measurement or for how to interpret the results. Here is a better version that *does* include a set of instructions:

The MATLAB function `DesignProblem02('Controller','datafile','data.mat')` will be used to simulate the system. The data generated by this function will be imported into a MATLAB program for analysis. The error at each time step will be found by taking the difference between the wheel center and the reference value. The maximum absolute value of error over all time steps will be reported—if it is less than 0.1 meters, the requirement is met.

Most engineers would argue that this verification still needs improvement. For example, does the simulation generate the same results every time, or is there variation? You know that there is variation because, for example, the initial values of ζ , $\dot{\zeta}$, θ , and $\dot{\theta}$ are selected at random, as is the shape of the ground. A reasonable engineer, then, would question whether or not the results of only *one* simulation would really show that the requirement is met. Many verifications also provide more than just a single number as evidence—for example, they might produce a figure (e.g., a plot of error as a function of time) or some other type of visualization. Your task is to define *one* verification for your requirement that has a measurement and a set of instructions for how to make the measurement and how to interpret the results.

C. Report

Your report must be a PDF document that was generated using \LaTeX and that conforms to the guidelines for “Preparation of Papers for AIAA Technical Conferences” (<https://go.aerospace.illinois.edu/aiaa-latex-template>). The author must be listed as “Anonymous” (with no affiliations). The report must be exactly four pages. It must have the following sections:

- *Abstract*. Summarize your key results in one short paragraph.
- *Nomenclature*. List all symbols used in your report, with units.
- *Goal*. At minimum, this section will describe the system you have been asked to control, define one requirement and one verification (Sections IV.A and IV.B), and prepare the reader to understand the rest of your report.
- *Model*. At minimum, this section will linearize (1) about some choice of equilibrium point, expressing the result in state-space form.
- *Control Design*. At minimum, this section will determine if the open-loop linear system is controllable, determine if the open-loop linear system is asymptotically stable, consider the application of state feedback, determine if the closed-loop linear system is asymptotically stable, and find the steady-state error in reference tracking (both with and without disturbance) of the closed-loop linear system. This section will likely focus on your efforts to minimize or eliminate steady-state error in the presence of non-flat ground.
- *Results*. At minimum, this section will describe what you did to implement and test your controller in simulation—in particular, you will have followed the instructions that you wrote to verify that your requirement has (or has not) been satisfied.

You are encouraged to go beyond these minimum requirements.

V. Deliverables

A. Submit a first draft by 11:59PM on Monday, March 2, 2020

This draft must be a PDF document that follows the guidelines provided in Section IV.C and that includes, at minimum, a **complete draft** of your “Goal” and “Model” sections. It must be submitted here:

<https://forms.illinois.edu/sec/9831449>

You may submit as many times as you want—only the last submission will be recorded.

B. Submit a second draft by 11:59PM on Monday, March 9, 2020

This draft must be a PDF document that follows the guidelines provided in Section IV.C and that includes, at minimum, a **complete draft** of your “Control Design” and “Results” sections. It must be submitted here:

<https://forms.illinois.edu/sec/6229549>

You may submit as many times as you want—only the last submission will be recorded.

C. Submit a final report by 11:59PM on Friday, March 13, 2020

This report must be a PDF document that follows the guidelines provided in Section IV.C. You must submit three things in addition to the report:

- The \LaTeX source files that you used to produce the document (in a ZIP folder with all figures, references, etc., in addition to the `.tex` file). Please do *not* include MATLAB code with your \LaTeX source files.
- The MATLAB code that you used to produce all of the figures, tables, and other results that are included in the document (in a ZIP folder). Your code must have a MATLAB script called `GenerateResults.m`. It should be possible for any of your peers to reproduce *everything* in your report by executing this script. Please do *not* include \LaTeX source files with your MATLAB code.
- The implementation of your “best” controller as a single file `Controller.m`. If one of your peers runs the simulator with this controller, they should see behavior that is consistent with claims made in the report.

Your report must be submitted here:

<https://forms.illinois.edu/sec/3484132>

You may submit as many times as you want—only the last submission will be recorded.

VI. Evaluation

Your work will be evaluated based on meeting intermediate deadlines (10%), on peer reviews both of your code (20%) and of your report (50%), and on your peer reviews of other reports (20%).

Reviews of your report will place special emphasis on **figures**. Reviewers will be encouraged to ignore issues with style, grammar, or any other aspect of your presentation other than figures, as long as there is no barrier to understanding your work.

Peer review will be “double-blind.” You won’t know who reviewed your report, and your reviewers won’t know whose report they reviewed. To enable a double-blind review process, it is **very important** that your final report be completely anonymous. To repeat, **DO NOT INCLUDE YOUR NAME** or anything else that would identify you in any of the materials you submit—not in your PDF, in your two ZIP folders, or in your MATLAB script. Other details of peer review will be posted to piazza and discussed in class.

VII. On-Time Submission

Draft reports and final reports must be submitted on time or they will receive zero credit. I am so serious about this that I will even give you extra credit if you submit your final report early:

If the last submission of your final report occurs by 11:59PM on Wednesday, March 11, 2020—so, 48 hours before the final deadline—then you will receive 5% extra credit.