MATLAB PROBLEM 3

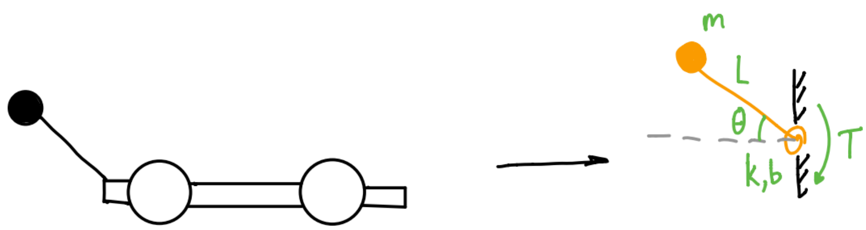
A quick note before you start – you will answer all questions for this problem within this Word document. When finished, please save the document as a .pdf, which you will upload directly to Gradescope. **For this and future MATLAB problems, we will also ask you to upload your code to BruinLearn.** This can be done through the same BruinLearn "assignment" from which you downloaded the problem set files. You should turn in all 3 of the .m files that you use in this problem.

In the field of autonomous robotics, there has recently been a trend toward "biomimetic design," or studying and replicating evolved characteristics of nature's most agile movers (click the links that follow to check out some sweet videos). Wall-climbing robots use gecko-like micro-spines to scale sheer surfaces with ease. Insect-like robots fly to a vertical surface, perch, climb, and then take off again. Cheetah-like robots, built with actuators that mimic biological muscle and tendon, offer unprecedented maneuverability and dexterity in a small package. Biomimetic design is motivated by a recognition that nature has millions of years of experience producing biological machines that adeptly navigate their environment, and that we would do well to draw from nature's expertise as we seek to do the same.

In this problem, we're going to focus on an adaptation that is wide-spread in the animal kingdom, and almost completely absent in the majority of today's robots: the use of a dynamic tail! Some fun studies out of UC Berkeley have explored how lizards use their tails for mid-air adjustments while jumping, and applied the same ideas to a wheeled robot. We are going to consider a simplified version of this fascinating problem, and explore how changing the system's dynamics affects overall behavior.

3a: Build a mathematical model and analytically solve the resulting differential equation.

The simplified robotic tail system can be modeled according to the diagram below.



Here, we have abstracted away the wheeled robot and represented it as a rigid wall, with a tail connected to it via a hinge joint. For our purposes, we've also chosen the robotic tail to be a *passive* dynamic system, meaning that its behavior is governed by 0^{th}, 1^{st}, and 2^{nd} order elements (spring, damper, and inertia, respectively). The tail has been reduced to a point mass m at a distance L from the center of the tail's rotation, such that the tail's inertia is $J = mL^2$. The rotational spring-damper has spring constant k and damping constant b. The spring produces zero torque at $\theta = 0$. All perturbations to the robot are represented by an external torque T acting on the system. For this problem, we can ignore the effects of gravity on the mass.

Your first task is to find the differential equation that models this system when there is no external torque acting on the tail ($T = 0$). For this first pass, let's say $m = 0.4\,kg$, $L = 0.5\,meters$, $b = 0.3\frac{Nms}{rad}$, and $k = 0.2\frac{Nm}{rad}$. Initial conditions are $\theta(0) = -1\,rad$ and $\dot{\theta}(0) = 0\,\frac{rad}{s}$. **Find a mathematical model that characterizes this system, and classify the equation as completely as possible. Once you have your model, solve the initial value problem analytically, using the methods we learned in class. Do all of this by hand, then take a picture and insert it into this document. Make sure that your work is readable! Hint:** it may make your math easier to multiply the equation through by a factor of 10 before solving.

MATLAB Problem 3:

$\tau_s = -k\theta \quad \tau_d = -b\omega \quad T = J\alpha = mL^2\alpha$

$\Sigma T = J\alpha = T - b\omega - k\theta$

$mL^2 \frac{d^2\theta}{dt^2} + b\frac{d\theta}{dt} + k\theta = T$

$m = 0.4\,kg,\ L = 0.5\,m,\ b = 0.3\,\frac{Nm\cdot s}{rad} \quad k = 0.2\,\frac{Nm}{rad}$

$(0.4kg)(0.5m)^2 \frac{d^2\theta}{dt} + (0.3\,\frac{Nm\cdot s}{rad})\frac{d\theta}{dt} + (0.2\,\frac{Nm}{rad})\,\theta = T$

$0.1\frac{d^2\theta}{dt} + 0.3\frac{d\theta}{dt} + 0.2\,\theta = T$

$(0.1\frac{d^2\theta}{dt^2} + 0.3\frac{d\theta}{dt} + 0.2\theta = 0)10$

$\frac{d^2\theta}{dt^2} + 3\frac{d\theta}{dt} + 2\theta = 0$

2nd Order Linear Homogeneous Constant Coefficient ODE

$\theta = e^{rt} \quad \frac{d\theta}{dt} = re^{rt} \quad \frac{d^2\theta}{dt^2} = r^2e^{rt}$

$(r^2+3r+2)e^{rt} = 0$

$(r+2)(r+1)e^{rt} = 0$

$r = -1, -2$

$\theta(t) = c_1e^{-t} + c_2e^{-2t}$

$\dot{\theta}(t) = -c_1e^{-t} - 2c_2e^{-2t}$

$\theta(0) = -1\ rad \quad \dot{\theta}(0) = 0\ rad/s$

$-1 = c_1e^{-(0)} + c_2e^{-2(0)}$

$-1 = c_1 + c_2$

$0 = -c_1e^{-(0)} - 2c_2e^{-2(0)}$

$0 = -c_1 - 2c_2$

$c_1 = -2c_2$

$-1 = -2c_2 + c_2$

$-1 = -c_2$

$c_2 = 1$

$c_1 = -1 - c_2$

$c_1 = -2$

$\theta(t) = -2e^{-t} + e^{-2t}$

Based on the roots of the characteristic equation, what can we say about this system's response?

The system will damp and there is no oscillation involved.

What will this system do as $t \to \infty$?

Theta will shrink to zero and the tail will become perpendicular to the rigid wall.

Now that you've solved the homogeneous system model, let's see what happens when we add an external torque $T = 30te^{-10t}$. **Hint**: don't forget that you may have already multiplied your equation by 10, and would need to adjust this accordingly. **Solve the resulting non-homogenous equation analytically, using the same initial conditions as the homogeneous equation. Do this by hand, then take a picture and insert it into this document. Feel free to convert to decimal approximations, where convenient.**

$T = 30te^{-10t}$

$0.1\frac{d^2\theta}{dt} + 0.3\frac{d\theta}{dt} + 0.2\,\theta = 30te^{-10t}$

$L[\theta] = 30te^{-10t}$

$\theta_p(t) = (A_0 + A_1 t)(Be^{-10t})$

$\theta_p(t) = A_0Be^{-10t} + A_1Bte^{-10t}$

$\theta_p(t) = Ae^{-10t} + Bte^{-10t}$

$\theta_p'(t) = -10Ae^{-10t} + Be^{-10t} - 10Bte^{-10t}$

$\theta_p''(t) = 100Ae^{-10t} - 10Be^{-10t} - 10Be^{-10t} + 100Bte^{-10t}$

$\theta''p(t) = 100Ae^{-10t} - 20Be^{-10t} + 100Bte^{-10t}$

$0.1(100Ae^{-10t} - 20Be^{-10t} + 100Bte^{-10t}) + 0.3(-10Ae^{-10t} + Be^{-10t} - 10Bte^{-10t}) + 0.2(Ae^{-10t} + Bte^{-10t}) = 30te^{-10t}$

$10Ae^{-10t} - 2Be^{-10t} + 10Bte^{-10t} - 3Ae^{-10t} + 0.3Be^{-10t} - 3Bte^{-10t} + 0.2Ae^{-10t} + 0.2Bte^{-10t} = 30te^{-10t}$

$7.2Ae^{-10t} - 1.7Be^{-10t} + 7.2Bte^{-10t} = 30te^{-10t}$

$7.2A - 1.7B = 0$

$7.2B = 30$

$B = 25/6$

$7.2A - 1.7(25/6) = 0$

$A = \frac{425}{432}$

$\theta_p(t) = \frac{425}{432}e^{-10t} + \frac{25}{6}te^{-10t}$

$\theta(t) = \theta_c(t) + \theta_p(t)$

$\theta(t) = -2e^{-t} + e^{-2t} + \frac{425}{432}e^{-10t} + \frac{25}{6}te^{-10t}$

What will this system do as $t \to \infty$? What does this tell us about the system's response to this particular external torque?

As t reaches infinity, theta goes to zero. The system will damp and there is no continuous oscillation. I graphed the solution and saw that there is a bump going past zero, and then eventually dropping back to zero.

3b: Use ODE45 to solve the homogeneous and non-homogeneous cases.

Download Homework4_MatlabProbs.m and roboTailODEfun.m. As always, without changing any of the file names for these files, put both into a single folder. Navigate to that folder in Matlab's "current folder" pane.

Open both files in MATLAB. In the section of Homework4_MatlabProbs.m entitled "Problem 3b", your task is to use ODE45 to solve your second-order differential equation describing the tail's behavior. To do this, we're going to make use of a technique that we haven't learned in class quite yet, which involves representing our 2nd order differential equation as a system of 1st order equations. For instance, consider the equation:

$$y''(t) + p(t)y'(t) + q(t)y(t) = g(t)$$

If we let $x_1 = y(t)$ and $x_2 = y'(t)$, then we readily see that $x_1' = x_2$ and $x_2' = y''(t)$. As such, we can rewrite our 2nd order equation as the following system of two 1st order equations:

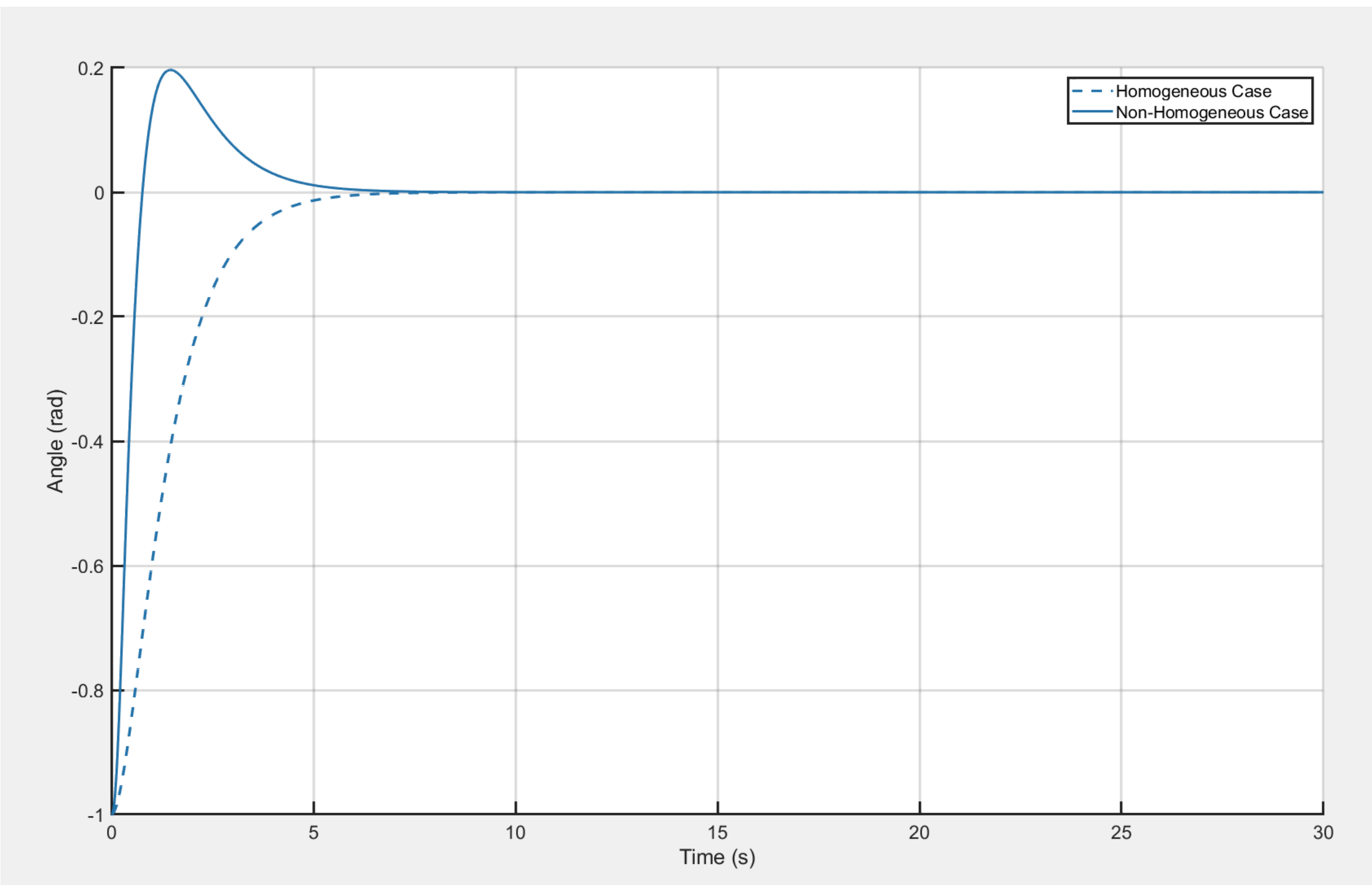
$$x_1' = x_2$$

$$x_2' = g(t) - p(t)x_2 - q(t)x_1$$

It turns out that this is an extremely helpful formulation, because ODE45 is built to handle systems of 1st order equations. Check out this example from MATLAB documentation before you continue: https://www.mathworks.com/help/matlab/math/solve-nonstiff-odes.html

This is not only useful for numerical methods; there are some great tools for analysis of linear 2nd and higher order ODEs that require reduction of the equation to a system of 1st order ODEs. We'll learn more about this later in the course.

For now, fill in the missing code in Problem 3b of Homework4_MatlabProbs.m and roboTailODEfun.m to find numerical solutions to the differential equation, for both the homogeneous and non-homogeneous cases of external torque: $T = 0$, and $T = 30te^{-10t}$. You'll want to solve these one-at-a-time, and store the results in two different sets of "y" and "t" variables. Fill in code to plot the resultant $\theta(t)$ trajectories for the two cases in Figure 1. Note that your "y" output will now be an nx2 array, where the first column contains the variable x_1 ($\theta(t)$ in your case) and the second column contains the variable x_2 ($\dot{\theta}(t)$ in your case). This means that you need to plot only the first column for each case. Be sure to follow the formatting instructions as described in the comments in the code. **Save the resultant figure as a .png or .jpg (don't just screenshot it), and insert the image into this document.**



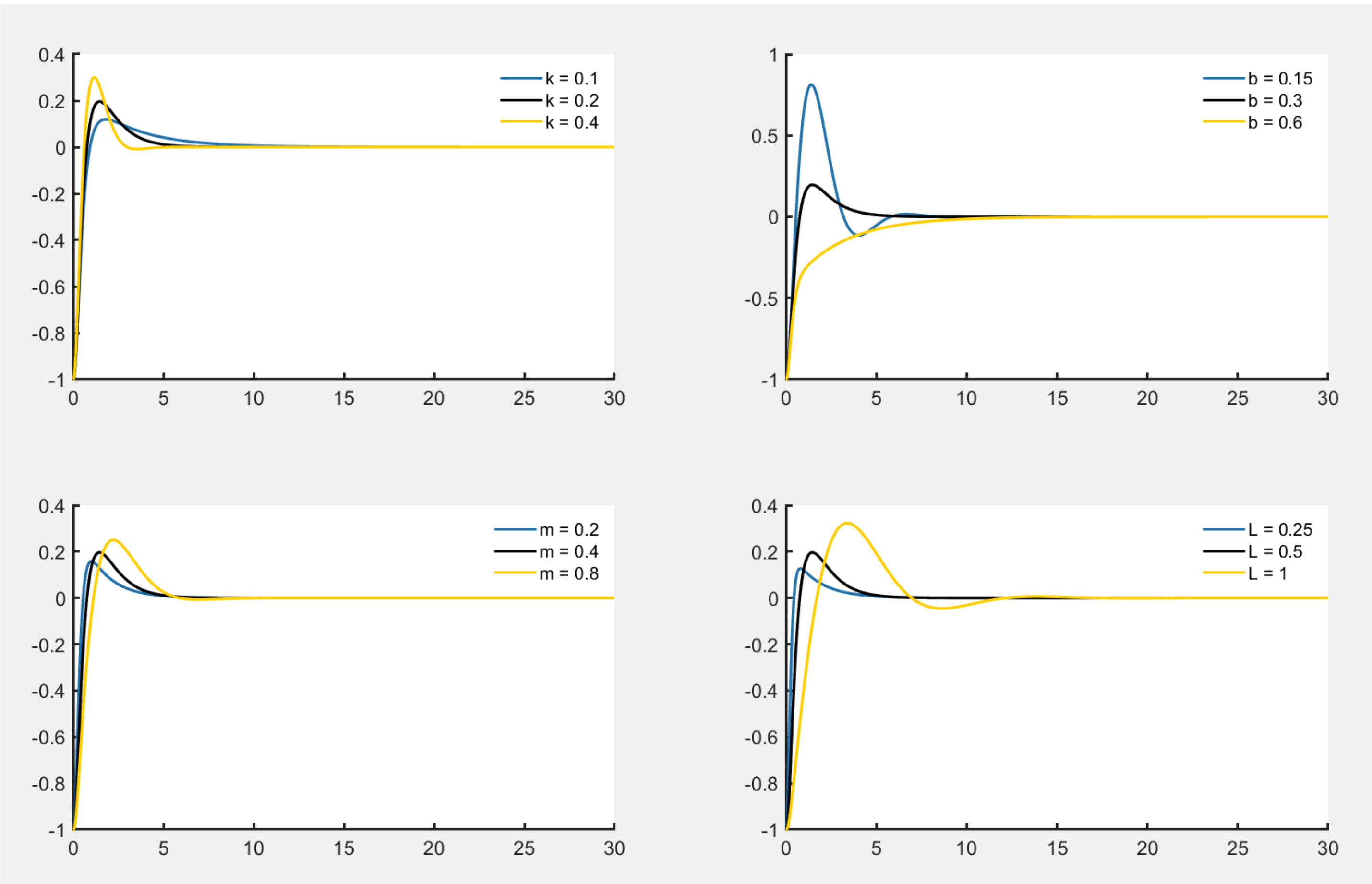
What do the two curves in this plot represent? Do these seem consistent with your analytical solutions? Hint: you might even consider using this plot to check that your analytical solutions are correct…

The two curves represent the solution to the initial value problem of the homogeneous and nonhomogeneous ordinary differential equation. They show the angle theta as a function of time. These plots do seem consistent with my analytical solutions because both graphs move towards zero as t increases.

3c. Explore how model parameters affect system behavior.

Next, we're going to explore the relative impact of changing each parameter on the overall system behavior. To do this, we'll use ODE45 to find a numerical solution to the non-homogeneous equation ($T = 30te^{-10t}$), and change the parameters one at a time while keeping everything else constant. For each parameter, we want to simulate system behavior at the original parameter value, at half the original value, and at twice the original value. We then want to plot these three solutions on top of each other, so we can see how they compare. We'll do this for each of the four key parameters: k, b, m, and L. Putting each of these in a subplot will make it easy to see them all at the same time. To get you started, I've done the first one for you.

Complete the code to add the remaining three subplots for b, m, and L, exactly as described above, and drop the resultant figure here.



Which of these parameters seems to have the greatest impact on system behavior? Which has the second-largest effect?

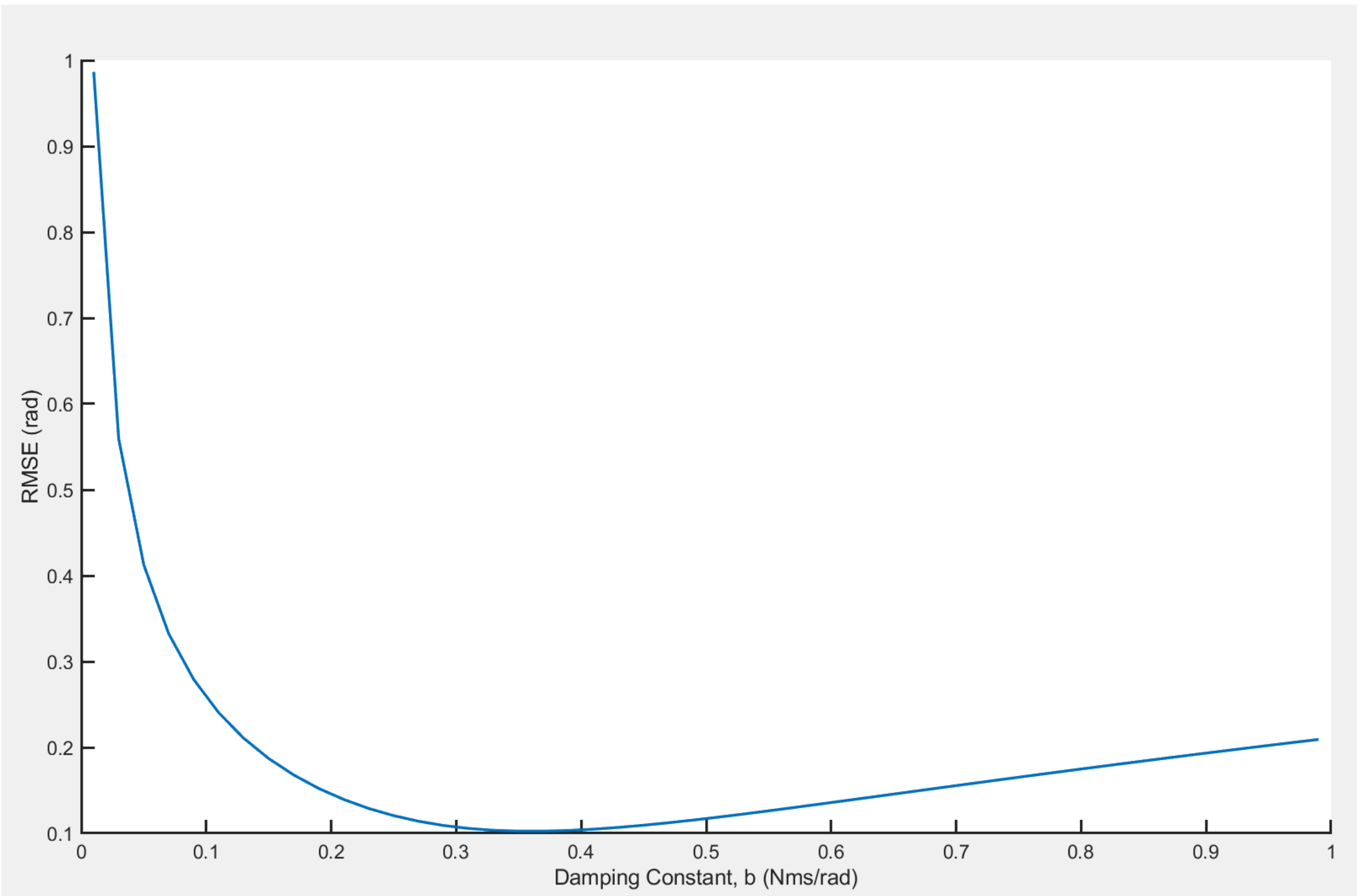
The damping constant, b, seems to have the greatest impact on system behavior. The distance the point mass is from the center of the tail's rotation, L, seems to have the second-largest effect.

Let's play with the damping a bit, and see if we can use it to minimize the impact of the external torque on our tail's stability. In other words, we want to get close to $\theta = 0$ as quickly as possible, and stay as close as possible without oscillating. One way to quantify this is our old friend RMSE. We can compare our results to 0 by simply taking the RMS of our resultant theta vector.

Why might RMSE be an effective way to capture our objective for $\theta(t)$ as described in the previous sentence?

We want the theta to be as close to zero as possible, and finding the minimum RMSE can achieve this.

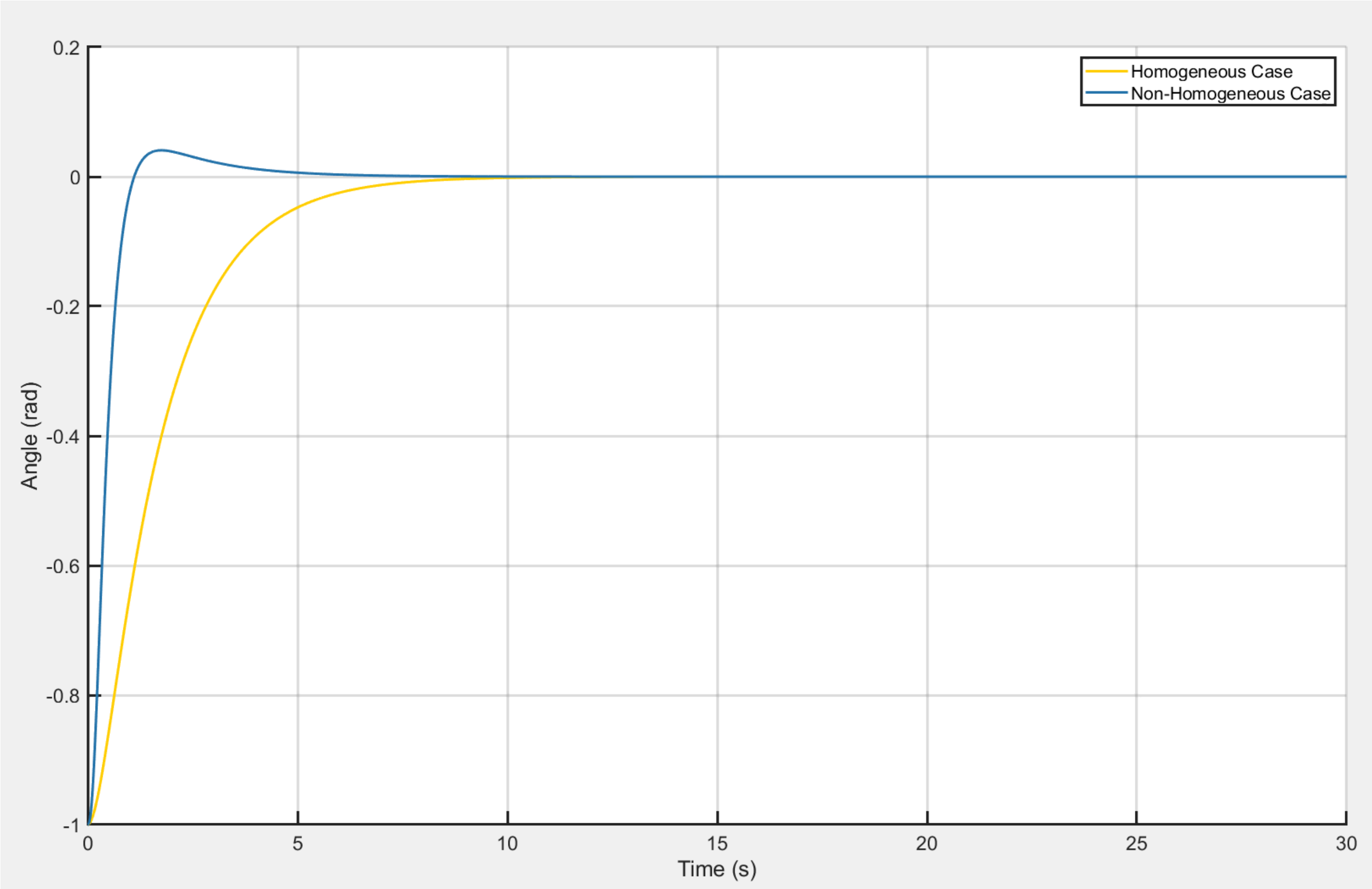
To minimize RMSE, we'll need to run an optimization. As in MATLAB Problem 1, we'll use a brute force method. Write code to sweep through the values of b contained in the array bVals in Homework4_MatlabProbs.m, while keeping all other parameters at their original values ($m = 0.4\, kg$, $L = 0.5\, meters$, and $k = 0.2 \frac{Nm}{rad}$). For each iteration, solve the differential equation numerically, and calculate RMSE of the resultant $\theta(t)$. Make sure you interpolate your theta vector to the testTime vector before finding the RMSE (see MATLAB problem 1c). Store these RMSE values, and make a plot of b versus RMSE. **Insert the resultant figure here.**



Based on this plot, does there appear to be an optimum value of b?

There does appear to be an optimum value of b.

It's time to see what your optimization produced. Write code to find the value of b that produces the minimum RMSE. Then, re-run your numerical simulation using that value of b, and plot the results on top of the non-homogeneous result from 3b. **Add your plot here.**



What value of b produced the minimum RMSE?

b=0.3700 Nms/rad

What was the minimum RMSE? Make sure to include units.

0.1025 radians

How does this compare to the RMSE produced using the original value of b from 3a?

This RMSE value is lower than the RMSE produced using the original value of b from 3a by about 0.003 radians.