AE353 Project 2: Differential Robot Simulation for State Feedback Controller Using Linear Quadratic Regulator

Austin A. Zickur
University of Illinois Urbana-Champaign, Urbana, Illinois, 61820

For Project 2, Linear Quadratic Regulator is utilized with linearization at an equilibrium point in finding a control gain-matrix, K, that will be used in a controller for the original, nonlinearized equations of motion. The robot's movement is dictated by the controller, which can be adjusted by the "weighted" matrix Q.

I. Nomenclature

e_l	=	lateral error (meter)
e_h	=	heading error (rad)
v	=	forward speed (m/s)
w	=	turning rate (rad/s)
$theta$	=	pitch angle (rad)
$thetadot$	=	pitch rate (rad/s)
tau_L	=	left wheel torque (N(m))
tau_R	=	right wheel torque (N(m))
A	=	state coefficient matrix
B	=	input coefficient matrix
x	=	state variable matrix
u	=	input variable matrix
Q	=	weighted state coefficient matrix
R	=	weighted input coefficient matrix
P	=	optimal cost matrix
K	=	optimal control gain-matrix
$Amatrix$	=	matrix A for linearized model
$Bmatrix$	=	matrix B for linearized model
$nonlinA$	=	matrix A for nonlinearized model
$nonlinB$	=	matrix B for nonlinearized model
$eigen$	=	eigenvalues of system (A-BK) for optimal control
f	=	vector-valued function for nonlinearized model
e_le	=	equilibrium leading error values
e_{he}	=	equilibrium heading error values
v_e	=	equilibrium forward speed values
w_e	=	equilibrium turning rate values
$theta_e$	=	equilibrium pitch angle values
$thetadot_e$	=	equilibrium pitch change values
tau_Le	=	equilibrium left wheel torque values
tau_Re	=	equilibrium right wheel torque values

II. Introduction

This design project involves linearizing a system involving equations of motion in Ordinary Differential form by taking an equilibrium velocity value for the differential robot, and considering initial velocity of the robot (2 m/s). When linearizing, we find an A and B matrix which can be utilized to find a K matrix using Linear Quadratic Regulator that assures linear relationship between input and state (u=-Kx). Then, using the determined value for K in (nonlinA-nonlinB K), we can test the controller's functionality on the nonlinearized model in moving the robot around the track at equilibrium velocity while stable enough to not move over the center line by .1m. The design requires completion of the track in less than 30 seconds, error values less than .1m, left and right wheel torque values less than 5N-m, and ability to sustain all requirements for any finite amount of time to be successful. These requirements prioritize reliability, speed, steadiness, and take account of physical constraints (such as maximum 5N-m actuators).Furthermore, the system's effectiveness in achieving the requirements can be increased by altering the weights of the controller in the Q matrix, each value representing a different state variable (e_l, e_h, v, w, theta, thetadot) and graphing models of torque, error, and speed with run times chosen related to requirements.

III. Theory

By finding the A and B matrices from the equations of motion, f matrix, we can find the linearized A and B matrix using the chosen equilibrium point. The nonlinearized A and B matrices can be found by partially differentiating f with respect to e_l, e_h, v, w, thetadot, theta (A) and tau_L, tau_R (B). The gain matrix, K, can then be found by LQR (K=Rinverse Btranspose P) with calculated P from A, B, Q, and R. Controllability is tested using the linearized A and B matrices and asymptotic stability is tested by the eigenvalues of (A-BK). This ensures the controller's functions where x is the difference between each state variable and the corresponding equilibrium state value and u is linearly related to x by a factor of K.

Vector-Valued Equations of Motion (f)

```
Matrix([[v*sin(e_h)], [w], [-(2400*tau_L + 2400*tau_R + 2808*(thetadot**2 + w**2)*sin(theta) + 13*(250*tau_L + 250*tau_R - 195*
w**2*sin(2*theta) - 8829*sin(theta))*cos(theta))/(11700*cos(theta)**2 - 12168)], [32*(-875*tau_L + 875*tau_R - 1443*thetadot*w*
sin(2*theta) - 2925*v*w*sin(theta))/(13*(3120*sin(theta)**2 + 2051))], [(42250*tau_L + 42250*tau_R - 32955*w**2*sin(2*theta) +
300*(100*tau_L + 100*tau_R + 117*(thetadot**2 + w**2)*sin(theta))*cos(theta) - 1492101*sin(theta))/(1404*(25*cos(theta)**2 - 2
6))], [thetadot]])
```

IV. Design

The design followed will be broken into subcategories including Equations and Matrices, Validity Check, and Digging Deeper.

A. Equations and Matrices

Firstly, vector-valued functions were found by setting up a matrix that utilizes the equations of motion with added function thetadot. This matrix, f, was partially differentiated twice with respect to e_l, e_h, v, w, thetadot, theta and tau_L, tau_R for the nonlinearized A and B matrices, respectively. Then, equilibrium values for each variable in the equations of motion are used in the nonlinearized A and B matrices found to obtain the linearized A and B matrices. This was done by lambdifying the two partially differentiated f matrices and running all equilibrium values previously defined into the now defined functions yielding the A and B matrices at the equilibrium point. Using LQR to obtain an accurate controller requires the use of the two weighted matrices Q and R, so they were created under the constraint of Q»R so the penalized state returns to 0 quickly. This ensures a steady, consistent, accurate controller for moving the robot. Furthermore, the P matrix was found using the scipy.linalg.solve-continuous-are command with the Amatrix, Bmatrix, Q, and R matrices. The gain matrix K was then found using LQR by K = Rinverse Btranspose P.

B. Validity Check

The first validity check was for the nonlinearized A and B matrices, symbolically, to ensure the differentiated variables are correct. nonlinA must be partially differentiated f such that nonlinA has form [[0,v*cos(e-h),sin(e-h),0,0,0],[0,0,0,1,0,0],. . . [0,0,0,0,1,0]] and nonlinB has form [[0,0],[0,0],. . . [0,0]]. Overall, nonlinA was checked for 6x6 size and nonlinB was checked for 6x2 size. The next validity check was to ensure that the value of f at the equilibrium point is [[0],[0],[0],[0],[0],[0]]. After the matrices were proven valid, controllability and asymptotic stability was validated for the system. The controllability matrix was checked for full rank so invertibility is possible, meaning the system using Amatrix and Bmatrix is controllable. The eigenvalues of the system, (A-BK), were validated as consisting

Coefficient Matrices (A,B)

```
[[ 0.00000e+00  4.00000e+00  0.00000e+00  0.00000e+0
   0.00000e+00]
 [ 0.00000e+00  0.00000e+00  0.00000e+00  1.00000e+0
   0.00000e+00]
 [ 0.00000e+00  0.00000e+00  0.00000e+00  0.00000e+0
  -2.45250e+02]
 [ 0.00000e+00  0.00000e+00  0.00000e+00  0.00000e+0
   0.00000e+00]
 [ 0.00000e+00  0.00000e+00  0.00000e+00 -0.00000e+0
   1.06275e+03]
 [ 0.00000e+00  0.00000e+00  0.00000e+00  0.00000e+0
   0.00000e+00]]
[[  0.           0.        ]
 [  0.           0.        ]
 [ 12.07264957  12.07264957]
 [ -1.05014439   1.05014439]
 [-51.46011396 -51.46011396]
 [  0.           0.        ]]
```

Fig. 1 figure 1

Controllability Matrix and Rank (W)

```
6
[[ 0.00000000e+00  0.00000000e+00  0.00000000e+00  0.00000000e+00
  -4.20057758e+00  4.20057758e+00  0.00000000e+00  0.00000000e+00
   0.00000000e+00  0.00000000e+00  0.00000000e+00  0.00000000e+00]
 [ 0.00000000e+00  0.00000000e+00 -1.05014439e+00  1.05014439e+00
   0.00000000e+00  0.00000000e+00  0.00000000e+00  0.00000000e+00
   0.00000000e+00  0.00000000e+00  0.00000000e+00  0.00000000e+00]
 [ 1.20726496e+01  1.20726496e+01  0.00000000e+00  0.00000000e+00
   1.26205929e+04  1.26205929e+04  0.00000000e+00  0.00000000e+00
   1.34125352e+07  1.34125352e+07  0.00000000e+00  0.00000000e+00]
 [-1.05014439e+00  1.05014439e+00  0.00000000e+00  0.00000000e+00
   0.00000000e+00  0.00000000e+00  0.00000000e+00  0.00000000e+00
   0.00000000e+00  0.00000000e+00  0.00000000e+00  0.00000000e+00]
 [-5.14601140e+01 -5.14601140e+01  0.00000000e+00  0.00000000e+00
  -5.46892361e+04 -5.46892361e+04  0.00000000e+00  0.00000000e+00
  -5.81209857e+07 -5.81209857e+07  0.00000000e+00  0.00000000e+00]
 [ 0.00000000e+00  0.00000000e+00 -5.14601140e+01 -5.14601140e+01
   0.00000000e+00  0.00000000e+00 -5.46892361e+04 -5.46892361e+04
   0.00000000e+00  0.00000000e+00 -5.81209857e+07 -5.81209857e+07]]
```

Fig. 2 figure2

of all negative real parts, meaning the system is asymptotically stable.

C. Digging Deeper

To examine how the change in values of the weighted matrix change the results of the controller for the nonlinearized system, Q and R were changed such that Q»R. Lateral error's corresponding value of Q (digit 1) seemed to make the robot slower and steadier when weight is increased while the heading error (digit 2) had a similar effect. Lowering the weight on both of these resulted in less steadiness of the robot. The linear velocity's corresponding value of Q (digit 3) seemed to make the robot fall forward while the angular velocity (digit 4) increased speed and pulled to the outside with increased weight. The thetadot's corresponding value of Q (digit 5) seemed to keep the robot centered and stable while theta (digit 6) made it slower and steadier with increased weight. It is important to note that decreased weight for thetadot results in the robot falling forward.

Gain Matrix and Eigenvalue Analysis (K)

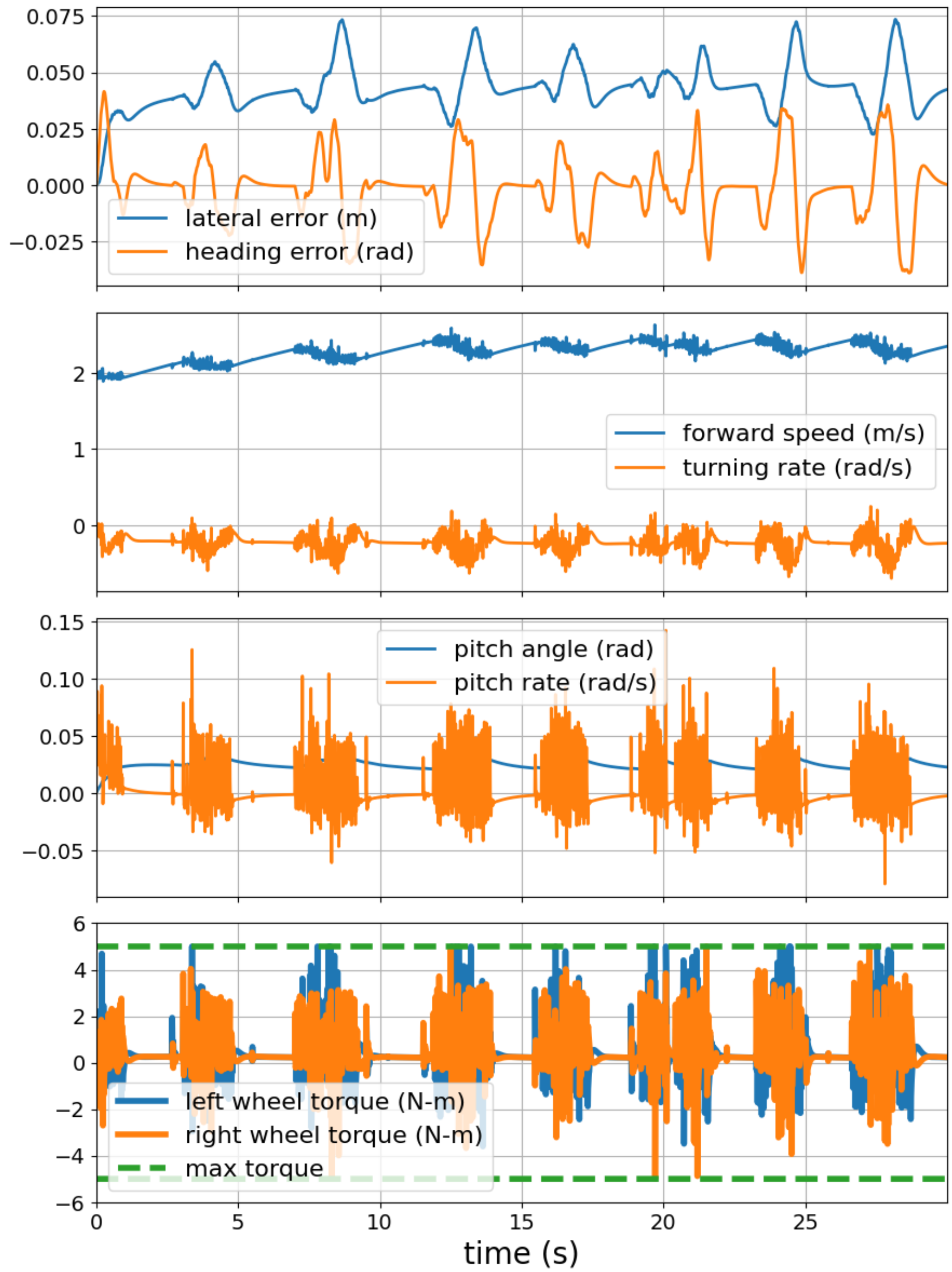
```
[[-50.         -60.27834645  -0.5          -9.07744763 -50.12287212
  -27.96239067]
 [ 50.          60.27834645  -0.5           9.07744763 -50.12287212
  -27.96239067]]
[-4.00909987e+00+4.68524341j -4.00909987e+00-4.68524341j
 -1.10470617e+01+0.j         -5.14623207e+03+0.j
 -1.76352802e-01+0.09814024j -1.76352802e-01-0.09814024j]
```

Weighted Matrices (Q,R)

```
[[10000     0     0     0     0     0]
 [    0    10     0     0     0     0]
 [    0     0     1     0     0     0]
 [    0     0     0   100     0     0]
 [    0     0     0     0 10000     0]
 [    0     0     0     0     0     1]]
[[2 0]
 [0 2]]
```

V. Results

Considering the judgements on Q, the corresponding values were chosen with correspondence with each other. For example, digits 1 and 5 involve slow, steady, center, stable, so those two values were chosen as 10,000 to maximize balance and steadiness. Digits 2 and 4 involve slow, steady, speed, pull, so those two values were chosen as 10 and 100, respectively. Digits 3 and 4 involve jerk and fluctuation, so those two values were chosen as 1 and 100, respectively. These considerations cross-reference all effects of Q on the chassis for smooth speed with precise movement. The plots show that the robot completes a full lap in less than 30 seconds with errors staying in the bounds of (-.03,.075). This is extremely smooth and optimally controlled movement. The forward speed gradually increases, although in the time frame of 30 seconds, it reaches a maximum of approximately 2.5 m/s. The left and right wheel torque stay within the 5 N-m maximum allowed by the constraints of the actuators.



VI. Conclusion

Since the results validate the design requirements, this design succeeds in functioning as an lqr controller for the chassis. The robot can continue on its path to completing laps for any finite amount of time and the lateral and heading errors are significantly smaller than the requirements stated. The torques do not exceed 5 N-m and the chassis moves at a constant, smooth pace. Using lqr allowed the different combinations of weights needed to optimize this system to validate the design requirements in the introduction.