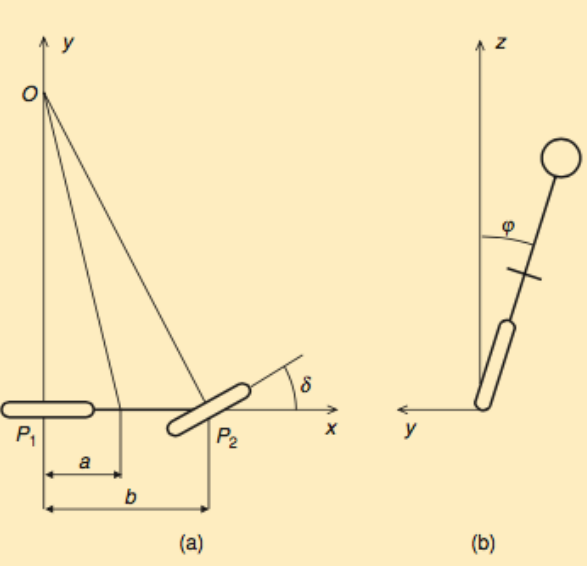
Problem Based Assignment 4

Callum Rohweder

METR7203  
October 2017



The above figure represents the parameters associated with the Whipple bicycle model; where represents the change in steering angle and denotes the change in tilt angle (this is referred to as gamma in the proceeding MATLAB files) . The Whipple model describes the bicycle by the following system of equations:

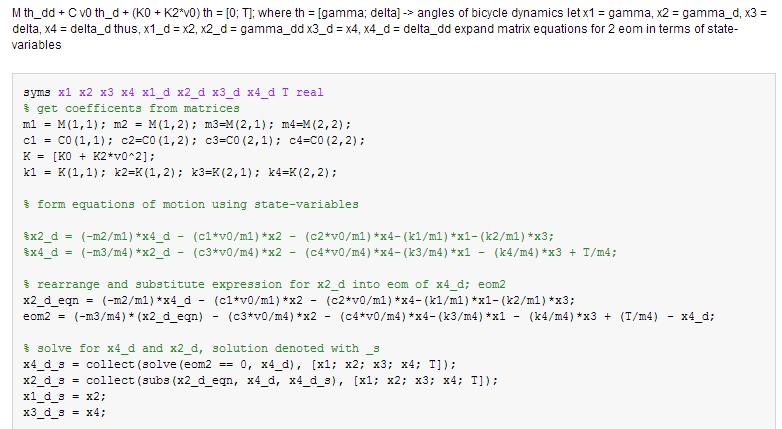
To derive the individual equations of motion of the system, M, the mass matrix is constructed by 2x2 elements denoted , similarly for and the spring-like matrix , such at:

The equations of motion can be evaluated to be:

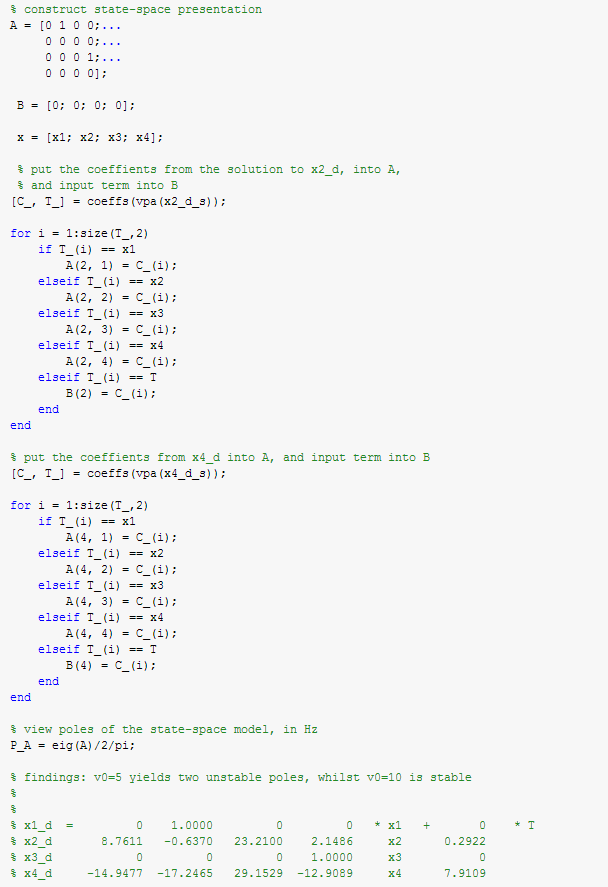
To develop the state-space model, the states will need to be chosen, let:

Thus, the equations of motion described above can be rearranged to be:

# Part A

Using the equations of motion written with state-space variables, the state-space model can be computed using MATLAB. This is done by solving for in the second state-space equation above, and then substituting this into the expression for , to gain an expression without derivative terms. This result is then put back into the last equation for an expression of in terms of state variables. This is presented in the code below, where the elements of the M, C and matrices are assigned to the appropriate element variables, from the equations of motion.

From here, the state-space equation can be constructed by arranging the coefficients of the state-derivate expressions into the state and input matrices. This is done in MATLAB, using the coeffs() function, and assigning values to A and B, as seen below. The eigenvalues of the state-space system for a velocity of 5ms-1 is computed, and the state-space representation highlighted.



This results in the following state-space equation:

Of which has two unstable poles:

And has a full rank controllability matrix, thus the system poles are moveable using state-feedback. This was worked out using the rank(ctrb(A,B)) command, which finds the rank of the controllability matrix of a system defined by A and B; the result was 4.

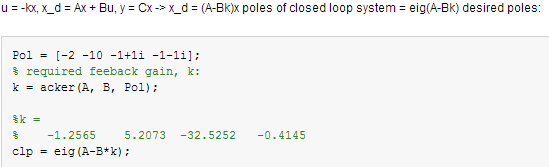
It was found that the system was stable for velocities between 6 and 10 meters per second, however in the above form is unstable.

# Part B

The acker function can be used to place the poles of a state-space system, defined by A and B, at a desired location using state-feedback. The matrix on the feedback loop, K, gains the states back to the input, creating the desired closed loop poles.

The input thus becomes , resulting in a remodel defined by:

Where the the eigenvalues of will be positioned at the desired locations; -2, -10, -11i. This can be seen in the code below.



This gives:

Thus it correctly places the poles at the desired location.

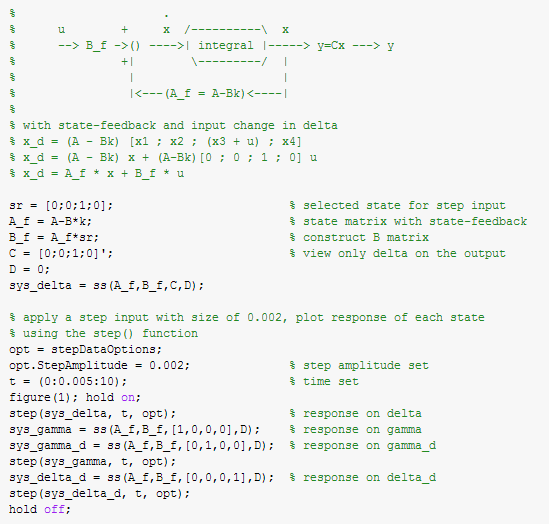
# Part 3

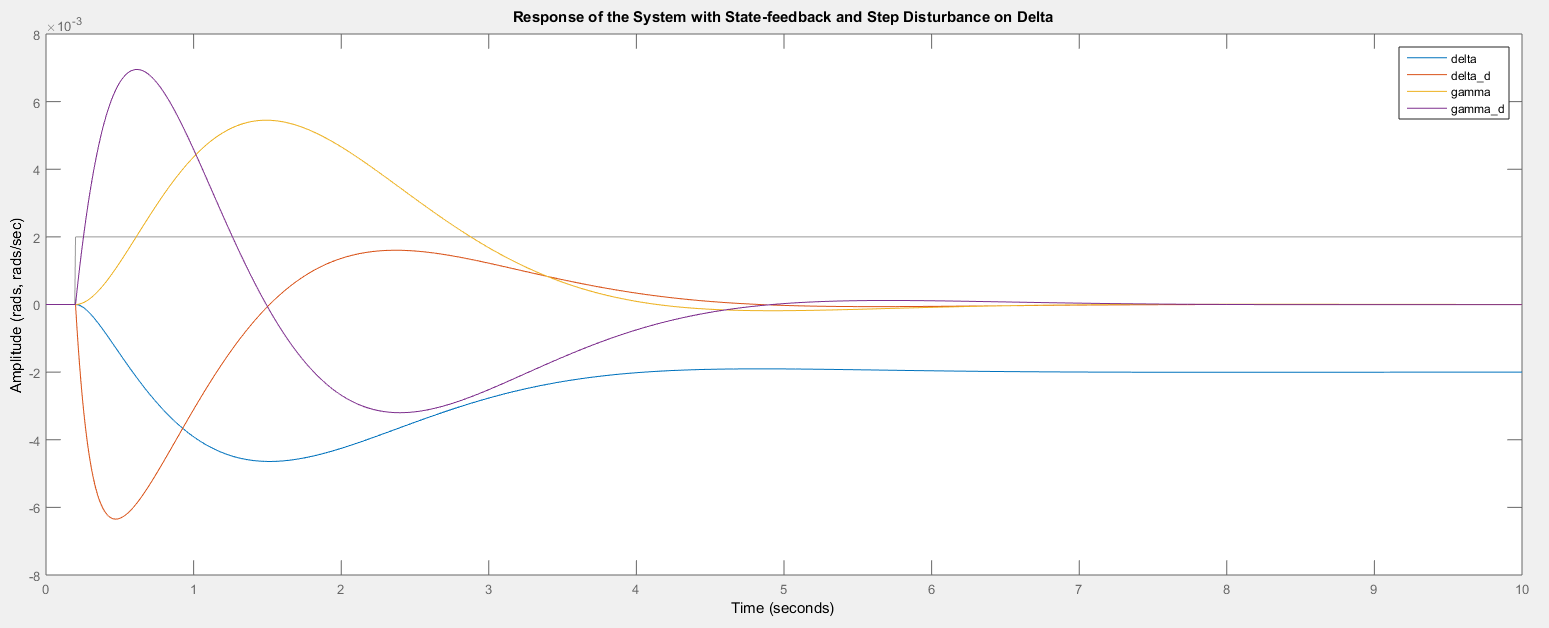
## Disturbance input of delta

Consider the situation where, before state-feedback, a step change in delta was caused by an input, u. The state-space model would be:

To view delta on the output, the C matrix is:

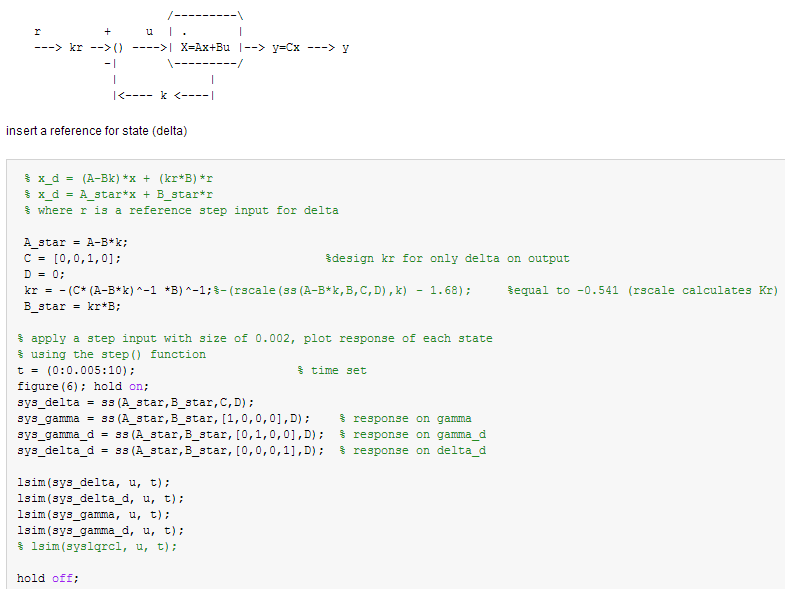
The result of a step input of 0.002 on this state can be modelled in MATLAB, in the following way:





It can be seen that all of the state’s change as a result of this disturbance, where the steering angle remains at -0.002 and the other’s return to their initial conditions. The response of delta is negative due to the input matrix B\_f containing a component related to -B\*K, of magnitude greater than A, thus resulting in a sign change when evaluated. This can be noticed when computing B\_f, giving [0; 20.7544; 0; -37.333], where when state 4 is integrated it results in delta being multiplied by (-37.33 + some feedback component).  
In a physical sense, a disturbance in position of the steering angle will result in a coupling with the tilt angle, which is not evident in the above graph. Thus in order to simulate a change in steering angle, this will need to be done through developing a model with an input torque that creates a steady-state step change in angle. This procedure follows:

## State reference input



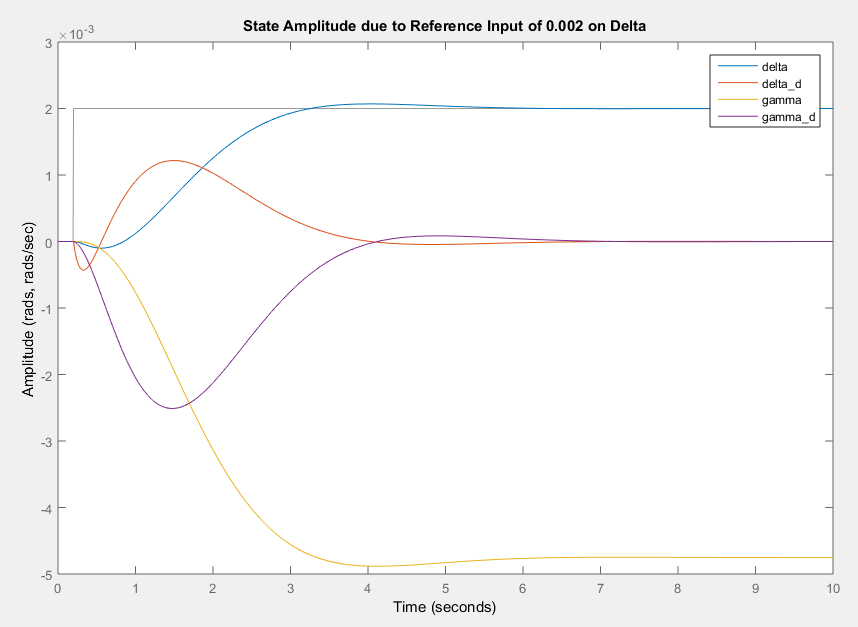
For the system presented above, it can be seen that there is a reference input for the state, r, and gain kr, which are inputted into the system along with the negative state-feedback. Similarly the state-space equations of this model can be written as follows:

During steady-state, there is no change in angles (gamma and delta), thus the angle’s velocities and accelerations are zero, resulting in . Thus, the value of kr to develop a steady state reference r on delta (the output is only delta, the other states are 0) can be computed as follows:

similarly, at the output:

As the output is only made up of the state of delta, we desire y = r:

It can be seen that the code above computes this, and the response of the states to a step input of 0.002, with kr designed to produce this response on delta:

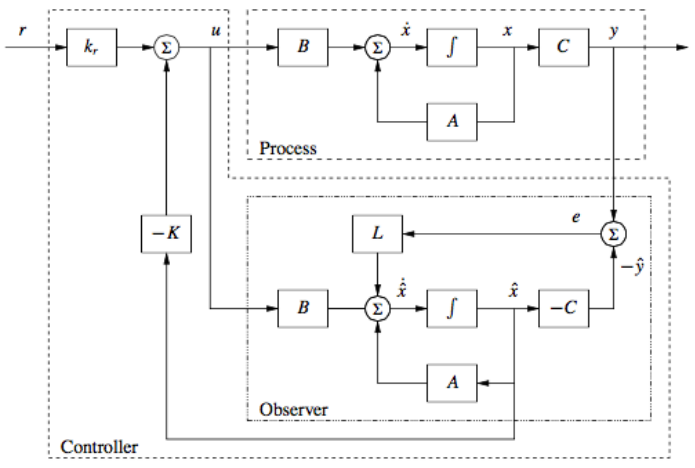


The coupling between the steering and tilt angles are evident in this circumstance.

# Part D

Below is a block-diagram representation of an observer, realistic process or plant, and the associated controller. Output feedback with a gain matrix, L, is used to place the poles of the system and correct for error between the plant and observer model. Without being able to directly measure the states of the plant, state-feedback comes from the observer state’s, ; designed to mimic that of the plant. Similarly to the previous section, the response of this system can be due to an input r and gain kr. The poles of the observer are typically designed to be 4 or 5 times the size of the plant, in order to correct for unexpected changes in the plant’s output. The state-space model can be written as:

This can be rearranged into state-space form where the state variables are the four of those from the plant, and the 4 ‘equivalent’ from the observer, denoted and  , respectively.



The state-space representation of this system can described by:

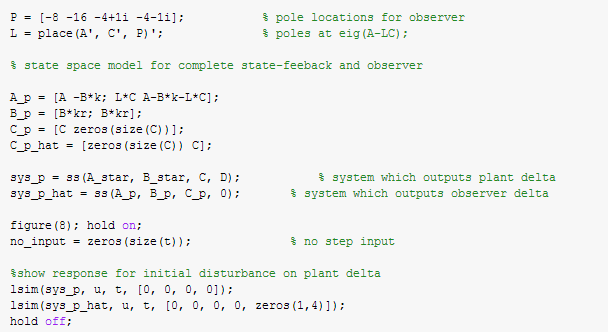
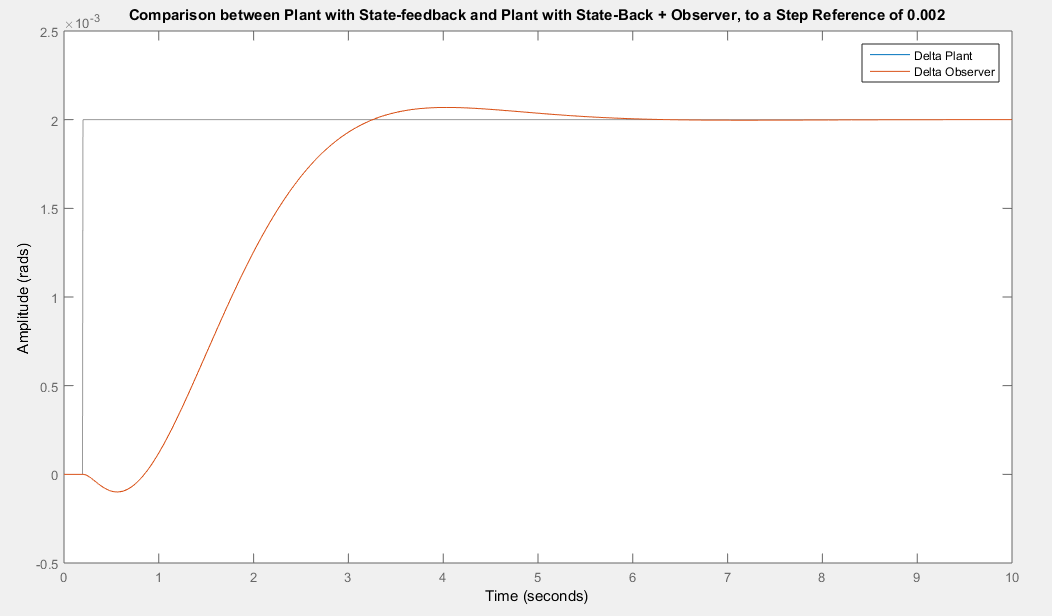
The eigenvalues of this system can be calculated in a similar manner to those in part B, however it was not shown, they are calculated by finding the determinant of (sI – A\*); where I is the identity matrix.

Let row2 = row1 – row2:

let column2 = column 1 + column 2:

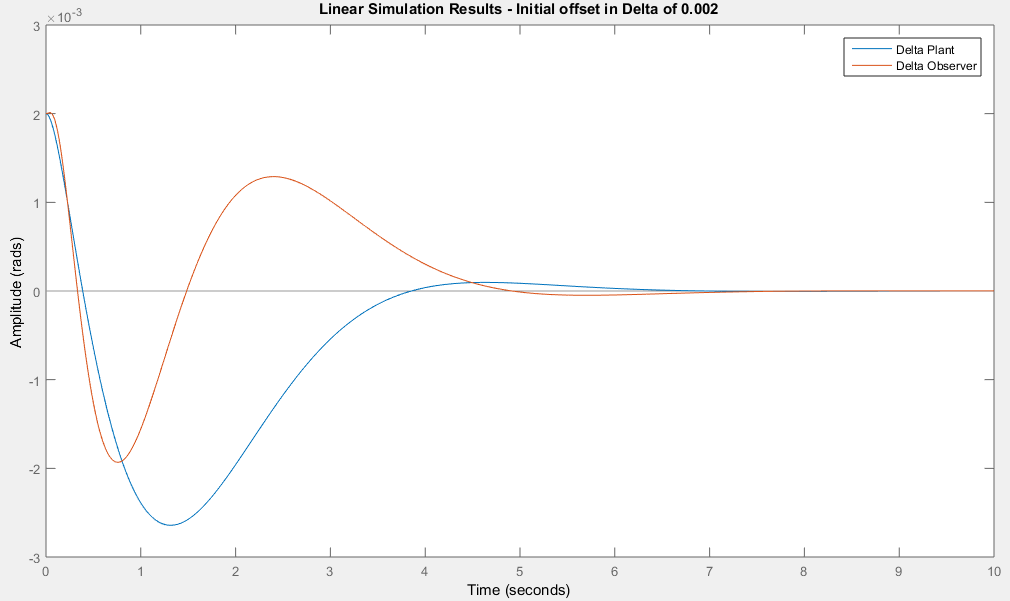
It can be seen above that the eigenvalues of the system are a combination of poles of the observer and poles of the state-feedback; where the state-feedback poles have already been chosen. Therefore, the eigenvalues of the observer can be chosen in a similar manner, using MATLAB’s acker() or place() function, such that:

The desired pole locations are P = [-8, -16, -4+1i, -4-1i]. The code below calculates the state-space model, the required L matrix for the aforementioned poles, and then plots the response of the model from part C and this new model to a step input reference; defined in the previous task.



It can be seen that the observer’s delta matches the plant’s exactly, when a step input is applied.

Interestingly, if the plant had a state originally offset from those of the observer, the responses don’t exactly match. With an initial state of 0.002, it can be seen that the observers unforced response is faster than the plant’s:  

# Part E

To understand the effect of the observer poles, in observing a plant with state-feedback poles denoted in the matrix Pol, the observer poles were changed to different multiples to that of the state-feedback poles. The responses of the different observers were plotted against the plant’s response, for both delta and phi, illustrated below. It can be seen that the observer with poles at 10\*Pol and 20\*Pol had identical responses, following the plant in shape but lower in amplitude. When the multiple of Pol was too low, below 2, the response became increasingly unstable. This is due to the observer compensating for the error between the plant and observer state’s too slowly, leading to an increase in error, and increase in amplitude, but a too large of a rise time to compensate.  
Having high frequency poles allows the observer to respond to changes in the plant’s state without noticeable deviation. High frequency poles can be sensitive to noise, and cause saturation of actuators, so there may be design considerations associated with how fast the poles can be. From the responses presented below, it is noticeable that poles 10 times those of the plant is sufficient for providing a similar shaped response. From research, most sources agree that 5 times faster is sufficient, however some too sight 10; again, this could be a design consideration.

The plots were generated by replicating the previous part, part D, various times for different scalings of Pol. Other variations of poles were simulated and the outcome the same, the response is constantly the same for frequencies higher than 10 times the plant poles, with the poles having a damping ratio between .707 and 1.

