Some Notes from the Book: Optimal Control and Estimation by Robert F. Stengel

John L. Weatherwax*

October 8, 2004

Introduction

Here you'll find some notes that I wrote up as I worked through this excellent book. I've worked hard to make these notes as good as I can, but I have no illusions that they are perfect. If you feel that that there is a better way to accomplish or explain an exercise or derivation presented in these notes; or that one or more of the explanations is unclear, incomplete, or misleading, please tell me. If you find an error of any kind – technical, grammatical, typographical, whatever – please tell me that, too. I'll gladly add to the acknowledgments in later printings the name of the first person to bring each problem to my attention.

Acknowledgments

Special thanks to (the most recent comments are listed first): Andrew J. Taylor and Adam Chapman for finding bugs and typos in this material.

All comments (no matter how small) are much appreciated. In fact, if you find these notes useful I would appreciate a contribution in the form of a solution to a problem that is not yet worked in these notes. Sort of a "take a penny, leave a penny" type of approach. Remember: pay it forward.

*wax@alum.mit.edu

The Mathematics of Control and Estimation

Notes on the text

Notes on the Newton-Raphson method

When J is a scalar function and u is a vector and we consider

$$\left.\frac{\partial J}{\partial u}\right|_{u=\xi},$$

this is a vector function of the point where we evaluate it i.e. ξ. If we Taylor expand this function about $\xi = u_0$ we have

$$\left.\frac{\partial J}{\partial u}\right|_{u=\xi} = \left.\frac{\partial J}{\partial u}\right|_{u=u_0} + (\xi - u_0)^T \left.\frac{\partial^2 J}{\partial u^2}\right|_{u=u_0}$$

If we evaluate this expression at $\xi = u^*$ the location of a minimum, where by the necessary condition for a minimum we have $\left.\frac{\partial J}{\partial u}\right|_{u=u^*} = 0$ and the left-hand-side vanishes and we get

$$0 = \left.\frac{\partial J}{\partial u}\right|_{u=u_0} + (u^* - u_0)^T \left.\frac{\partial^2 J}{\partial u^2}\right|_{u=u_0}.$$

Solving for u^* and we get

$$u^* = u_0 - \left(\frac{\partial^2 J}{\partial u^2}\right)^{-1} \left(\frac{\partial J}{\partial u}\right)^T,$$

where both $\frac{\partial^2 J}{\partial u^2}$ and $\frac{\partial J}{\partial u}$ are evaluated at $u = u_0$.

Notes on Lagrange Multipliers

Because of the vector constraint $\mathbf{f}(\mathbf{u}') = 0$ is of dimension n (the vector $\mathbf{f}$ has n components) the vector $\mathbf{u}'$ must have at least n components or else it is over-specified by the constraint $\mathbf{f}(\mathbf{u}') = 0$. Lets therefore assume that $\mathbf{u}'$ is of dimension $n+m$. Using the constraint we could in principal solve for n of the components u_i's in the vector $\mathbf{u}'$ terms of the remaining m other variables. To emphasis this we write the vector $\mathbf{u}'$ as $\mathbf{u}' = (\mathbf{x}, \mathbf{u})$ where the components in the n vector $\mathbf{x}$ are viewed as functions of the remaining m elements in $\mathbf{u}$, implicitly defined by $f(\mathbf{x}, \mathbf{u}) = 0$. Then form the augmented objective function J_A defined as

$$J_A(\mathbf{x}, \mathbf{u}) = J(\mathbf{x}, \mathbf{u}) + \boldsymbol{\lambda}^T \mathbf{f}(\mathbf{x}, \mathbf{u}). \tag{1}$$

Here the vector $\boldsymbol{\lambda}$ is of dimension equal to the number of constraints in $\mathbf{f} = 0$ or n. Then ΔJ_A is given by

$$\Delta J_A = \left.\frac{\partial J_A}{\partial \mathbf{x}}\right|_{(\mathbf{x},\mathbf{u})=(\mathbf{x}^*,\mathbf{u}^*)} \Delta\mathbf{x} + \left.\frac{\partial J_A}{\partial \mathbf{u}}\right|_{(\mathbf{x},\mathbf{u})=(\mathbf{x}^*,\mathbf{u}^*)} \Delta\mathbf{u}.$$

But from the form of J_A given in Equation 1 we have these derivatives given by

$$\frac{\partial J_A}{\partial \mathbf{x}} = \frac{\partial J}{\partial \mathbf{x}} + \boldsymbol{\lambda}^T \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \tag{2}$$

$$\frac{\partial J_A}{\partial \mathbf{u}} = \frac{\partial J}{\partial \mathbf{u}} + \boldsymbol{\lambda}^T \frac{\partial \mathbf{f}}{\partial \mathbf{u}} \,. \tag{3}$$

If we pick $\boldsymbol{\lambda}^*$ such that $\frac{\partial J_A}{\partial \mathbf{x}} = 0$ in Equation 2 or

$$\boldsymbol{\lambda}^* = -\left[\left(\frac{\partial J}{\partial \mathbf{x}} \right) \left(\frac{\partial \mathbf{f}}{\partial \mathbf{x}} \right)^{-1} \right]^T = -\left[\left(\frac{\partial \mathbf{f}}{\partial \mathbf{x}} \right)^{-1} \right]^T \left(\frac{\partial J}{\partial \mathbf{x}} \right) . \tag{4}$$

Then ΔJ_A is given by (we can drop the $\Delta \mathbf{x}$ term) with the above value for $\boldsymbol{\lambda}^*$

$$\Delta J_A = \left. \frac{\partial J_A}{\partial \mathbf{u}} \right|_{(\mathbf{x},\mathbf{u})=(\mathbf{x}^*,\mathbf{u}^*)} \Delta \mathbf{u} = \left(\frac{\partial J}{\partial \mathbf{u}} + \boldsymbol{\lambda}^{*T} \frac{\partial \mathbf{f}}{\partial \mathbf{u}} \right) \Delta \mathbf{u} \,.$$

For $\Delta J_A = 0$ for all $\Delta \mathbf{u}$ requires that

$$\frac{\partial J}{\partial \mathbf{u}} + \boldsymbol{\lambda}^{*T} \frac{\partial \mathbf{f}}{\partial \mathbf{u}} = 0 \,, \tag{5}$$

or when we put in $\boldsymbol{\lambda}^*$ from before we get

$$\frac{\partial J}{\partial \mathbf{u}} - \frac{\partial J}{\partial \mathbf{x}} \left(\frac{\partial \mathbf{f}}{\partial \mathbf{x}} \right)^{-1} \frac{\partial \mathbf{f}}{\partial \mathbf{u}} = 0 \,. \tag{6}$$

Note that Equation 6 is a system of m equations that we need to solve for $\mathbf{u}$ and $\mathbf{x}$ to find the constrained extrema. We get a complete system of $m+n$ equations by augmenting the n constraint equations $\mathbf{f}(\mathbf{x},\mathbf{u}) = 0$ to this system. If the *combined* system

$$\frac{\partial J}{\partial \mathbf{u}} - \frac{\partial J}{\partial \mathbf{x}} \left(\frac{\partial \mathbf{f}}{\partial \mathbf{x}} \right)^{-1} \frac{\partial \mathbf{f}}{\partial \mathbf{u}} = 0 \quad \text{and} \tag{7}$$

$$\mathbf{f}(\mathbf{x},\mathbf{u}) = 0 \,, \tag{8}$$

is sufficiently complicated then it may have to be solved using Newton iterations like as discussed on Page 2.

Notes on Example 2.1-5 (decent into the valley part 2)

Consider the example where $J = u_1^2 - 2u_1u_2 + 3u_2^2 - 40$. The let $u = u_1$ and $x = u_2$ to get

$$J(x,u) = u^2 - 2ux + 3x^2 - 40 \,.$$

The equality constraint is $x - u - 2 = 0$ which is also the definition of the constraint $f(x,u) = 0$. Then to find the minimum of this constrained problem we need to solve the system given by Equations 7 and 8. One way to do that is to follow the derivation above by first computing the Lagrange multiplier as in Equation 4 we have

$$\lambda^* = -\left(\frac{\partial f}{\partial x} \right)^{-1} \left(\frac{\partial J}{\partial x} \right)^T .$$

Thus we need to compute

$$\frac{\partial f}{\partial x} = 1 \quad \text{and} \quad \frac{\partial J}{\partial x} = -2u + 6x\,.$$

So $\lambda^* = -(-2u + 6x) = 2u - 6x$. To put this into Equation 5 we need

$$\frac{\partial J}{\partial u} = 2u - 2x \quad \text{and} \quad \frac{\partial f}{\partial u} = -1\,.$$

Then Equation 5 becomes

$$(2u - 2x) + (2u - 6x)(-1) = 0\,,$$

or $x = 0$. Then with this value for x the constraint $f(x,u) = 0$ gives $u^* = -2$. Then we find the optimal value for J using these two numbers given by

$$J^* = J(x^*, u^*) = 4 - 2(-2)(0) + 0 - 40 = -36\,,$$

as we found from the method of substitution.

Notes on sufficient condition for a constrained minimum

For a constrained problem a positive definite Hessian matrix for J_A is sufficient to conclude that we have a minimum of our constrained problem. We can write a block matrix expression for this Hessian as

$$\Delta^2 J_A = \frac{1}{2}\begin{bmatrix} \Delta\mathbf{x}^T & \Delta\mathbf{u}^T \end{bmatrix}\begin{bmatrix} J_{A_{xx}} & J_{A_{xu}} \\ J_{A_{ux}} & J_{A_{uu}} \end{bmatrix}\begin{bmatrix} \Delta\mathbf{x} \\ \Delta\mathbf{u} \end{bmatrix}.$$

If $\Delta^2 J_A \equiv 0$ then we require that a higher order *even* derivative be positive definite. By the constraint $\mathbf{f}(\mathbf{x},\mathbf{u}) = 0$ changes in $\mathbf{x}$ and $\mathbf{u}$ must be related by $\Delta\mathbf{f} = 0$ or

$$\mathbf{f_x}\Delta\mathbf{x} + \mathbf{f_u}\Delta\mathbf{u} = 0\,.$$

Thus $\Delta\mathbf{x} = -\mathbf{f_x}^{-1}\mathbf{f_u}\Delta\mathbf{u}$. Thus the left most matrix above becomes

$$\begin{bmatrix} \Delta\mathbf{x}^T & \Delta\mathbf{u}^T \end{bmatrix} = \begin{bmatrix} -\Delta\mathbf{u}^T\mathbf{f_u}^T(\mathbf{f_x}^{-1})^T & \Delta\mathbf{u}^T \end{bmatrix} = \Delta\mathbf{u}^T\begin{bmatrix} -(\mathbf{f_x}^{-1}\mathbf{f_u})^T & I \end{bmatrix}.$$

Putting this into the expression for $\Delta^2 J_A$ we get that

$$\begin{aligned}
\Delta^2 J_A &= \frac{1}{2}\Delta\mathbf{u}^T\begin{bmatrix} -(\mathbf{f_x}^{-1}\mathbf{f_u}) & I \end{bmatrix}\begin{bmatrix} J_{A_{xx}} & J_{A_{xu}} \\ J_{A_{ux}} & J_{A_{uu}} \end{bmatrix}\begin{bmatrix} -(\mathbf{f_x}^{-1}\mathbf{f_u}) \\ I \end{bmatrix}\Delta\mathbf{u} \\
&= \frac{1}{2}\Delta\mathbf{u}^T\left[(\mathbf{f_x}^{-1}\mathbf{f_u})^T J_{A_{xx}}(\mathbf{f_x}^{-1}\mathbf{f_u}) - (\mathbf{f_x}^{-1}\mathbf{f_u})^T J_{A_{xu}} - J_{A_{ux}}(\mathbf{f_x}^{-1}\mathbf{f_u}) + J_{A_{uu}}\right]\Delta\mathbf{u} \qquad (9) \\
&= \frac{1}{2}\Delta\mathbf{u}^T J'_{A_{uu}}\Delta\mathbf{u}\,,
\end{aligned}$$

where we have defined $J'_{A_{uu}}$ as the four term expression above it. Then $\Delta^2 J_A$ will be positive definite if $J'_{A_{uu}}$ is. Thus one can compute $J'_{A_{uu}}$ and observe if this is positive definite to determine if we have found a constrained, local, minimum of J.

Notes on Pivotal Condensation for Evaluating Determinants

The method of Laplace expansion is what is typically taught in a course on linear algebra for evaluating determinants. Another method for evaluating determinants which the book claims may requires fewer multiplications and is more easily implemented is called **Pivotal Condensation**. In this method, to evaluate the determinant of A, a nonzero element of A, say a_{ij}, is selected. Then many 2×2 determinants of submatrices with one element taken to be the element a_{ij} two elements taken from the ith row and jth column of A and the fourth element taken from A with the ith row and jth column removed. These determinants as elements give a matrix of size $n-1 \times n-1$ made up of these smaller determinants. This matrix is multiplied by $\left(\frac{1}{a_{ij}}\right)^{n-2}$ to give a final $n-1 \times n-1$ matrix. This process is repeated until a 2×2 matrix is obtained for which we know how to take the determinant. An example will help clarify this procedure. Consider evaluating the determinant of the 3×3 matrix A

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} .$$

If we assume that $a_{11} \neq 0$ and perform pivotal condensation about this element. Then $i = 1$ and $j = 1$. Now we consider the matrix A' that remains when we remove the first row and first column or

$$A' = \begin{bmatrix} a_{22} & a_{23} \\ a_{32} & a_{33} \end{bmatrix} .$$

Then for *each* element in the matrix A' we consider a matrix of 2×2 determinants that are created using the element a_{11} as the upper left element, and and an element from the above matrix as its lower right element. The cross diagonal elements are obtained by selecting the corresponding elements from row $i = 1$ and column $j = 1$ that are in the same row and column of A and then dividing by $\frac{1}{a_{11}^{3-2}} = \frac{1}{a_{11}}$ for example we would have

$$\begin{aligned} |A| &= \frac{1}{a_{11}} \begin{vmatrix} \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} & \begin{vmatrix} a_{11} & a_{13} \\ a_{21} & a_{23} \end{vmatrix} \\ \begin{vmatrix} a_{11} & a_{12} \\ a_{31} & a_{32} \end{vmatrix} & \begin{vmatrix} a_{11} & a_{13} \\ a_{31} & a_{33} \end{vmatrix} \end{vmatrix} = \frac{1}{a_{11}} \begin{vmatrix} a_{11}a_{22} - a_{12}a_{21} & a_{11}a_{23} - a_{13}a_{21} \\ a_{11}a_{32} - a_{12}a_{31} & a_{11}a_{33} - a_{13}a_{31} \end{vmatrix} \\ &= \frac{1}{a_{11}}[(a_{11}a_{22} - a_{12}a_{21})(a_{11}a_{33} - a_{13}a_{31}) - (a_{11}a_{23} - a_{13}a_{21})(a_{11}a_{32} - a_{12}a_{31})] \\ &= \frac{1}{a_{11}} \left[a_{11}^2 a_{22}a_{33} - a_{11}a_{22}a_{13}a_{31} - a_{12}a_{21}a_{11}a_{33} + a_{12}a_{21}a_{13}a_{31} \right. \\ &\quad \left. - a_{11}^2 a_{23}a_{32} + a_{11}a_{12}a_{23}a_{31} + a_{11}a_{13}a_{21}a_{32} - a_{13}a_{21}a_{12}a_{31}\right] \\ &= a_{11}a_{22}a_{33} + a_{12}a_{23}a_{31} + a_{13}a_{21}a_{32} - a_{11}a_{23}a_{32} - a_{12}a_{21}a_{33} - a_{22}a_{13}a_{31} , \end{aligned}$$

which is the save value when computed via other means.

Notes on properties of positive/negative definite matrices

Given a matrix Q of size $n \times n$, we define the **principal minors** of Q to be determinants of smaller square matrices obtained from the matrix Q. The smaller submatrices are selected

from Q by selecting a set of indices from 1 to n representing the rows (and columns) we want to downsample from. Thus if you view the indices selected as the indices of rows from the original matrix Q to extract into the submatrix, then the columns we select for this submatrix must *equal* the indices of the rows we select. As an example, if the matrix Q is 6×6 we could construct one of the principal minors from the first, third, and sixth rows. If we denote the elements of Q denoted as q_{ij} then this would be the value of

$$\begin{vmatrix} q_{11} & q_{13} & q_{16} \\ q_{31} & q_{33} & q_{36} \\ q_{61} & q_{63} & q_{66} \end{vmatrix} .$$

Then the theorem of interest about how principal minors relate to Q is that if *all* principal minors of a matrix Q are *positive* then we can conclude that Q is positive definite. In addition, if all principal minors are either positive or *zero* then the matrix Q is positive semidefinite.

We next consider a slight modification of the above definition by considering the **leading principal minors** which are the specific principal minors that we get by considering the first k rows and columns. Since Q is of dimension $n \times n$ we will then have n leading principal minors. We can denote these leading principal minors as Δ_k. In the example above where Q was assumed to be a 6×6 matrix the 6 leading principal minors are the following expressions

$$\Delta_1 = |q_{11}| , \quad \Delta_2 = \begin{vmatrix} q_{11} & q_{12} \\ q_{21} & q_{22} \end{vmatrix} , \quad \Delta_3 = \begin{vmatrix} q_{11} & q_{12} & q_{13} \\ q_{21} & q_{22} & q_{23} \\ q_{31} & q_{32} & q_{33} \end{vmatrix}$$

$$\Delta_4 = \begin{vmatrix} q_{11} & q_{12} & q_{13} & q_{14} \\ q_{21} & q_{22} & q_{23} & q_{24} \\ q_{31} & q_{32} & q_{33} & q_{34} \\ q_{41} & q_{42} & q_{43} & q_{44} \end{vmatrix} , \quad \Delta_5 = \begin{vmatrix} q_{11} & q_{12} & q_{13} & q_{14} & q_{15} \\ q_{21} & q_{22} & q_{23} & q_{24} & q_{25} \\ q_{31} & q_{32} & q_{33} & q_{34} & q_{35} \\ q_{41} & q_{42} & q_{43} & q_{44} & q_{45} \\ q_{51} & q_{52} & q_{53} & q_{54} & q_{55} \end{vmatrix} ,$$

and Δ_6 is the the determinant of Q. Then with these definitions we have the fact that if the leading principal minors are all positive then Q is positive definite. If in fact the leading principal minors *alternate* in sign as we consider more of them as

$$\Delta_1 < 0 , \quad \Delta_2 > 0 , \quad \Delta_3 < 0 , \quad \cdots ,$$

then Q is *negative definite*. The take away from this section is that one way to test of positive/negative definiteness of a matrices we might be working with is to consider the signs of the leading principal minors.

Notes on derivation of the pseudoinverses (left and right)

To begin this section we assume that A is of dimension $r \times n$. Then the product AA^T is of dimension $r \times r$ and the product $A^T A$ is of dimension $n \times n$. Because of the dimensions of A the largest the value that the of A can be is $\min(r, n)$. It is possible that the rank A actually be smaller than $\min(r, n)$, for example if it was a matrix with a single column

repeated multiple times. We will assume for further discussion that the rank of A equals this minimum. Then there are two cases to consider when attempting to "solve" the expression

$$y = Ax\,.$$

for x. These two cases depend on which is larger r or n.

- **Overdetermined Systems:** The most common situation is the one that arises when we have many measurements (elements of y) each of which is a linear mapping (via A) of a few parameters (elements of x). Thus the dimension of y is larger than the dimension of x or we have $r > n$. Then since the rank of A is $\min(r, n) = n$ the matrix $A^T A$ which is of size $n \times n$ is nonsingular. Thus we will define the **Left Pseudoinverse** for overdetermined system is given by

$$A^{\text{LPI}} = (A^T A)^{-1} A^T\,. \tag{10}$$

- **Underdetermined Systems:** This is the opposite case as above and we have that $r < n$ and $y = Ax$ is said to be underdetermined. When this happens there are *multiple* solutions to the system $Ax = y$. In this case AA^T is of dimension $r \times r$ and because of the rank condition on A the product matrix is non-singular. We now derive the right pseudoinverse by starting with the fact that AA^T is invertible or the expression

$$(AA^T)(AA^T)^{-1} = I\,.$$

  Using $y = Ax$ we can multiply the left-hand-side of this expression by y and the right-hand-side of it by Ax to get

$$(AA^T)(AA^T)^{-1} y = Ax\,.$$

  Then "canceling" A on both sides of this equation gives the expression for x in terms of y or

$$x = A^T (AA^T)^{-1} y\,.$$

  Thus we define the **Right Pseudoinverse** for underdetermined system as

$$A^{\text{RPI}} = A^T (AA^T)^{-1}\,. \tag{11}$$

  We note a property of this expression. The right pseudoinverse selects one of the multiple possible solutions in that it is the solution x with the smallest value of $||x||$ over all x that satisfy $Ax = y$.

Given the type of system we are presented with (in terms of the relative size of r and n) we then "solve" for x (at least in principle) using

$$x = A^{\text{XPI}} y\,.$$

where X is either L or R and computed using Equation 10 or Equation 11 depending on the problem considered.

Notes on resistor network

In the book at the end of this section it is mentioned that the overdetermined system produces solutions x that are more robust to errors in measurements than simply using a smaller but invertible system (like for instance using only the first two measurements). To test this idea lets introduce a 10% error in the measurement of i_2 i.e. we take $i_2' = 0.9(2.667) = 2.4$ Amps. The true voltages do not change i.e. $V_1 = 100$ V and $V_2 = 200$ V. Then using the left pseudoinverse gives

$$\begin{bmatrix} V_1 \\ V_2 \end{bmatrix} = A^{\text{LPI}} \begin{bmatrix} 0.5 \\ 2.4 \\ 2 \end{bmatrix} = \begin{bmatrix} 118.611 \\ 162.20 \end{bmatrix}.$$

While using just the first two current measurements gives

$$\begin{bmatrix} V_1 \\ V_2 \end{bmatrix} = \begin{bmatrix} -3000 & 600 \\ 2000 & -300 \end{bmatrix} \begin{bmatrix} 0.5 \\ 2.4 \end{bmatrix} = \begin{bmatrix} -60 \\ 280 \end{bmatrix}.$$

Given the correct value of $\begin{bmatrix} 100 \\ 200 \end{bmatrix}$ we see that the first method does a much better job at estimating its value than the second method does. This experiment is performed in the Matlab script `chap_2_notes_on_resistor_network.m`.

Notes on minimal norm flow (the right pseudoinverse)

The book derives the minimal dimensional flow rate expression $\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 8.33 \\ 8.33 \\ 8.33 \end{bmatrix}$, which correspond to percentage opening for values with maximal openings of $10, 20, 30$ given by

$$\frac{8.33}{10} = 0.833\,, \quad \frac{8.33}{20} = 0.4167\,, \quad \frac{8.33}{30} = 0.277\,.$$

Now maybe we want to solve for a minimum norm solution in *percentage* flow terms rather than dimensional flow. We can solve this problem by redefining the definition of the matrix A. Then our problem in the percentage flow terms is to find $\begin{bmatrix} x_1 & x_2 & x_3 \end{bmatrix}$ such that

$$25 = \begin{bmatrix} 10 & 20 & 30 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}.$$

The minimum norm solution to the above system is given by the right pseudoinverse as

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = A^T(AA^T)^{-1}(25) = \begin{bmatrix} 10 \\ 20 \\ 30 \end{bmatrix} \frac{1}{10^2 + 20^2 + 30^2}(25) = \begin{bmatrix} 0.178 \\ 0.358 \\ .535 \end{bmatrix},$$

the same answer one gets in Example 2.2-5. Another solution method would be to open each valve by the same p percent. This would require

$$p(10) + p(20) + p(30) = 25\,,$$

or $p = 0.41$.

Notes on matrix identities

In this section we consider block matrix inverses. That means we start with a block matrix A of dimensions $(m+n)\times(m+n)$ written as

$$A = \begin{bmatrix} A_1 & A_2 \\ A_3 & A_4 \end{bmatrix},$$

and then consider its inverse B partitioned in the same way

$$B = \begin{bmatrix} B_1 & B_2 \\ B_3 & B_4 \end{bmatrix},$$

Our goal is to find the block matrix elements of B in terms of the block matrix elements of A. We can focus on the right column of the expression $AB = I$ to derive equations for B_1 and B_3. For example, if we then form the matrix block product $AB = I$ and equate the $(2,1)$ block we get an expression for B_3, in terms of B_1. Equating the $(1,1)$ block then gives an equation for B_1, in terms of the submatrices of A. We can use that solution to derive an expression for B_3 in terms of the submatrices of A. This gives the expressions

$$\begin{aligned} B_1 &= (A_1 - A_2A_4^{-1}A_3)^{-1} && (12)\\ B_3 &= -A_4^{-1}A_3(A_1 - A_2A_4^{-1}A_3)^{-1}\,. && (13)\end{aligned}$$

The same sort of procedure on the second column of the expression $AB = I$ gives relationships for B_2 and B_4 in terms of the components of A. In this case the equating the $(1,2)$ block element gives for B_2

$$B_2 = -A_1^{-1}A_2B_4\,.$$

Then equating the $(2,2)$ element gives

$$A_3B_2 + A_4B_4 = I_n\,.$$

or grouping to solve for B_4 we find

$$(-A_3A_1^{-1}A_2 + A_4)B_4 = I \quad \text{so} \quad B_4 = (A_4 - A_3A_1^{-1}A_2)^{-1}\,. \tag{14}$$

Using this in what we just derived for B_2 give

$$B_2 = -A_1^{-1}A_2(A_4 - A_3A_1^{-1}A_2)^{-1}\,.$$

If A is symmetric then its block matrices must satisfy $A_1^T = A_1$, $A_4^T = A_4$, and $A_3 = A_2^T$, and its inverse matrix B must *also* be symmetric. This means that the block element of B at $(1,2)$ (or B_2) must equal the transpose of the block element of B at $(2,1)$ (or B_3). Since we know expressions for B_2 and B_3 in terms of block elements of A this means that from

$$-B_2 = -B_3^T\,,$$

we get

$$\begin{aligned} A_1^{-1}A_2(A_4 - A_3A_1^{-1}A_2)^{-1} &= (A_4^{-1}A_3(A_1 - A_2A_4^{-1}A_3))^T = (A_1^T - A_3^TA_4^{-T}A_2^T)^{-1}A_3^TA_4^{-T} \\ &= (A_1 - A_2A_4^{-1}A_2^T)^{-1}A_2A_4^{-1}\,. \end{aligned} \tag{15}$$

Similar relationships can be derived by considering the product $BA = I$. If we form that block product and then consider the equation given by $(2,2)$ component we have

$$B_3 A_2 + B_4 A_4 = I_n \,,$$

or solving for B_4 we get

$$B_4 = (I_n - B_3 A_2) A_4^{-1} = A_4^{-1} - B_3 A_2 A_4^{-1} \,. \tag{16}$$

Using B_4 from Equation 14 and B_3 from Equation 13 we derive

$$\begin{aligned}(A_4 - A_3 A_1^{-1} A_2)^{-1} &= A_4^{-1} + A_4^{-1} A_3 (A_1 - A_2 A_4^{-1} A_3)^{-1} A_2 A_4^{-1} \\ &= A_4^{-1} - A_4^{-1} A_3 (A_2 A_4^{-1} A_3 - A_1)^{-1} A_2 A_4^{-1} \,.\end{aligned}$$

Note that we can get an alternative expression by negating the matrix A_1. The resulting expression will be called the **Matrix Inversion Lemma**:

$$(A_4 \pm A_3 A_1^{-1} A_2)^{-1} = A_4^{-1} - A_4^{-1} A_3 (A_2 A_4^{-1} A_3 \pm A_1)^{-1} A_2 A_4^{-1} \,. \tag{17}$$

If A is symmetric $A_3 = A_2^T$ and we get the **Symmetric Matrix Inversion Lemma**:

$$(A_4 \pm A_2^T A_1^{-1} A_2)^{-1} = A_4^{-1} - A_4^{-1} A_2^T (A_2 A_4^{-1} A_2^T \pm A_1)^{-1} A_2 A_4^{-1} \,. \tag{18}$$

Notes on Numerical Integration of Linear Equations

In this section we will discuss solution methods for the linear system for the perturbation function $\Delta x(t)$ that in general looks like

$$\Delta \dot{x}(t) = F(t) \Delta x(t) + G(t) \Delta u(t) + L(t) \Delta w(t) \,. \tag{19}$$

The total solution to Equation 19 is specified in terms of the solution to an unforced (homogeneous) part plus the solution to the forced or inhomogeneous part. We start with the solution to the homogeneous part

$$\Delta \dot{x}(t) = F(t) \Delta x(t) \,.$$

One way to solve this is to construct the so called fundamental solution matrix, $U(t)$, from unit initial conditions in each of the state variables. What this means is that we solve $\Delta \dot{x}(t) = F(t) \Delta x(t)$, with n "unit" initial conditions given by

$$\begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ \vdots \\ 0 \end{bmatrix}, \cdots, \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix} .$$

Note each of these i initial conditions $1 \leq i \leq n$ is a zero vector with a single one in the ith spot. The solutions at time t to each of these problems is denoted by the column vectors

$$\Delta x_1(t) \,, \Delta x_2(t) \,, \cdots \,, \Delta x_n(t) \,.$$

If we put all n of these solutions into the columns of a matrix $U(t)$ then by the fact that $\Delta\dot{x}(t) = F(t)\Delta x(t)$, is a linear problem by superposition the solution at any time t to that problem with an arbitrary initial condition $\Delta x(t_0)$ is given by

$$[\Delta x_1(t)\,, \Delta x_2(t)\,, \cdots, \Delta x_n(t)]\,\Delta x(t_0) \equiv U(t)\Delta x(t_0)\,.$$

Thus the solution $\Delta x(t)$ is given by

$$\Delta x(t) = U(t)\Delta x(t_0)\,. \tag{20}$$

The matrix $U(t)$ is known as the *fundamental solution matrix*. An example of these fundamental solutions $\Delta x_i(t)$ is given in figure 2.3-2 in the book. There we see that at $t = t_0 = 0$ the first function $\Delta x_1(t)$ takes the value 1 in its first component and is 0 in all other components, the function $\Delta x_2(t)$ takes the value 1 in its second component and is 0 in all other components, etc. Viewing Equation 20 this fundamental solution matrix U can be though of as linear mapping of the initial conditions $\Delta x(t_0)$ at the time t_0 to the solution $\Delta x(t)$ at an arbitrary time. In the same way the mapping U^{-1} can then be viewed as a mapping of the solution from the time t back to the initial time t_0. That is

$$\begin{aligned}\Delta x(t_1) &= U(t_1)\Delta x(t_0)\\ \Delta x(t_0) &= U(t_1)^{-1}\Delta x(t_1)\,.\end{aligned}$$

We can generate a mapping of the solution $\Delta x(t)$ between two arbitrary times say t_1 and t_2 by combining these two relationships

$$\Delta x(t_2) = U(t_2)\Delta x(t_0) = U(t_2)U(t_1)^{-1}\Delta x(t_1) \equiv \Phi(t_2, t_1)\Delta x(t_1)\,,$$

where we have defined

$$\Phi(t_2, t_1) = U(t_2)U(t_1)^{-1}\,, \tag{21}$$

and Φ is known as the *state-transition matrix*. We now introduce some properties of the state transition matrix. If we take t_2 to be arbitrary say $t_2 = t$ from the above we have

$$\Delta x(t) = \Phi(t, t_1)\Delta x(t_1) \quad \text{so} \quad \Delta\dot{x}(t) = \dot{\Phi}(t, t_1)\Delta x(t_1)\,,$$

but from the differential equation $\Delta\dot{x}(t) = F(t)\Delta x(t)$ and we have

$$\dot{\Phi}(t, t_1)\Delta x(t_1) = F(t)\Delta x(t) = F(t)\Phi(t, t_1)\Delta x(t_1)\,.$$

Thus we get the important identity

$$\frac{d\Phi(t, t_1)}{dt} = F(t)\Phi(t, t_1)\,. \tag{22}$$

Thus the state-transition matrix $\Phi(t, t_1)$, when viewed as a function of its first argument, satisfies the given differential equation. In the special case where $F(t)$ is not actually a function of time we can make some simplifications. In that case $\Phi(t, t_0)$ takes on a simple form

$$\Phi(t, t_0) = e^{F(t-t_0)}\,, \tag{23}$$

so that $\Phi(t, t_0)$ only depends on difference in time $t - t_0$. If this time difference is Δt then $\Phi(\Delta t) = e^{F\Delta t}$. Thus we see that

$$\Phi(2\Delta t) = e^{F2\Delta t} = (e^{F\Delta t})(e^{F\Delta t}) = (e^{F\Delta t})^2 = \Phi(\Delta t)^2\,.$$

In the same way in general we find we have

$$\Phi(n\Delta t) = \Phi(\Delta t)^n \,. \tag{24}$$

After this discussion we want to now discuss solving the forced system using our knowledge of the fundamental matrix $U(t)$. Recall the derivative of a matrix inverse identity

$$\frac{d}{dt}U^{-1} = -U^{-1}\dot{U}U^{-1} \,, \tag{25}$$

we can solve this for $\dot{U}$ to get

$$\dot{U} = -U\left(\frac{d}{dt}U^{-1}\right)U \,.$$

In addition, by the definition of U it must solve the differential equation $\dot{U} = FU$ and we get

$$-U\left(\frac{d}{dt}U^{-1}\right)U = FU \,,$$

or canceling factors

$$\frac{d}{dt}U^{-1} = -U^{-1}F \,. \tag{26}$$

When we premultiply the linear dynamic equation of interest Equation 19 or

$$\Delta\dot{x} = F(t)\Delta x + G(t)\Delta u(t) + L(t)\Delta w(t) \,,$$

by U^{-1} we get

$$U^{-1}\Delta\dot{x} = U^{-1}F(t)\Delta x + U^{-1}G\Delta u + U^{-1}L\Delta w \,.$$

If we add to this to $\left(\frac{d}{dt}U^{-1}\right)\Delta x = -U^{-1}F\Delta x$ we get

$$U^{-1}\Delta\dot{x} + \frac{d}{dt}U^{-1}\ \Delta x = U^{-1}G\Delta u + U^{-1}L\Delta w \,,$$

or using the product rule on the left-hand-side

$$\frac{d}{dt}(U^{-1}\Delta x) = U^{-1}[G\Delta u + L\Delta w] \,. \tag{27}$$

When we integrate this from t_0 to t we find

$$U^{-1}(t)\Delta x(t) - U^{-1}(t_0) = \int_{t_0}^{t} U^{-1}(\tau)[G(\tau)\Delta u(\tau) + L(\tau)\Delta w(\tau)]d\tau \,.$$

Premultiply both sides by $U(t)$ and recall the definition of Φ via Equation 21 we get the *full* solution for $\Delta x(t)$ given by

$$\Delta x(t) = \Phi(t, t_0)\Delta x(t_0) + \int_{t_0}^{t} \Phi(t, \tau)[G(\tau)\Delta u(\tau) + L(\tau)\Delta w(\tau)]d\tau \,. \tag{28}$$

If we take $G(\cdot)$ to be the identity matrix and Δu to be the Dirac delta function $\delta(t_0 - \tau)$ in the first coordinate then the forcing term in the solution for $\Delta x(t)$ above becomes

$$\int_{t_0}^{t} \Phi(t,\tau) \begin{bmatrix} \delta(t_0-\tau) \\ 0 \\ \vdots \\ 0 \end{bmatrix} d\tau = \int_{t_0}^{t} (\text{first column of } \Phi(t,\tau))\delta(t_0-\tau)d\tau$$

$$= \text{first column of } \Phi(t,t_0) = \Delta x_1(t)\,,$$

where $\Delta x_1(t)$ is the first fundamental solution column.

We can use Equation 28 as a recursive algorithm to compute $\Delta x(t_k)$ from $\Delta x(t_{k-1})$. Do do this we take $t \to t_k$ and $t_0 \to t_{k-1}$ as

$$\Delta x(t_k) = \Phi(t_k, t_{k-1})\Delta x(t_{k-1}) + \int_{t_{k-1}}^{t_k} \Phi(t_k,\tau)[G(\tau)\Delta u(\tau) + L(\tau)\Delta w(\tau)]d\tau\,.$$

If, in addition, all matrices are independent of time and we take $\Delta u(\tau)$ and $\Delta w(\tau)$ constant at their pointwise values at the left end of the τ interval: $t_{k-1} \le \tau \le t_k$, i.e. equal to $\Delta u(t_{k-1})$ and $\Delta w(t_{k-1})$ the above simplifies further. Since when F is constant we have $\Phi(t_k, t_{k-1}) = e^{F(t_k-\tau)} = e^{F(t_k-\tau)}$ and get with $\Delta t = t_k - t_{k-1}$ that

$$\Delta x(t_k) = \Phi(\Delta t)\Delta x(t_{k-1}) + \int_{t_{k-1}}^{t_k} e^{F(t_k-\tau)}d\tau\,[G\Delta u(t_{k-1}) + L\Delta w(t_{k-1})]\,. \tag{29}$$

Note that we can factor the expression $G\Delta u(t_{k-1}) + L\Delta w(t_{k-1})$ out of the integral since it is a constant vector. Let write the exponential expression we are integrating as

$$e^{F(t_k - t_{k-1} + t_{k-1} - \tau)} = e^{F\Delta t}e^{F(t_{k-1}-\tau)} = \Phi(\Delta t)e^{F(t_{k-1}-\tau)}\,.$$

Then let $v = -(t_{k-1} - \tau) = \tau - t_{k-1}$ so that $dv = d\tau$ and our integral $\int_{t_{k-1}}^{t_k} e^{F(t_k-\tau)}d\tau$ becomes

$$\Phi(\Delta t)\int_0^{\Delta t} e^{-Fv}dv\,. \tag{30}$$

If F were a scalar (and not a matrix0 then $\int e^{-Fv}dv = -\frac{e^{-Fv}}{F}$. As F is a matrix this integral has to be written as

$$-e^{-Fv}F^{-1} \quad \text{or} \quad -F^{-1}e^{-Fv}\,.$$

These two expressions are the same since the matrices e^{-Fv} and F^{-1} commute. Considering the first of these two expressions we have

$$\int_0^{\Delta t} e^{-Fv}dv = -\,e^{-Fv}F^{-1}\Big|_0^{\Delta t} = (-e^{-F\Delta t} + I)F^{-1} = (I - \Phi^{-1}(\Delta t))F^{-1}\,.$$

Using this in Equation 29 we have $\Delta x(t_k)$ given by

$$\Phi(\Delta t)\Delta x(t_{k-1}) + \Phi(\Delta t)(I - \Phi^{-1}(\Delta t))F^{-1}G\Delta u(t_{k-1}) + \Phi(\Delta t)(I - \Phi^{-1}(\Delta t))F^{-1}L\Delta w(t_{k-1})\,.$$

If we define $\Gamma(\Delta t)$ and $\Lambda(\Delta t)$ as

$$\Gamma(\Delta t) = \Phi(\Delta t)(I - \Phi^{-1}(\Delta t))F^{-1}G \tag{31}$$
$$\Lambda(\Delta t) = \Phi(\Delta t)(I - \Phi^{-1}(\Delta t))F^{-1}L\,, \tag{32}$$

we can write $\Delta x(t_k)$ using these as

$$\Delta x(t_k) = \Phi(\Delta t)\Delta x(t_{k-1}) + \Gamma(\Delta t)\Delta u(t_{k-1}) + \Lambda(\Delta t)\Delta w(t_{k-1})\,. \tag{33}$$

At this point we note that we don't have to demand that F be invertible to evaluate $\Delta x(t_k)$. We can also evaluate the integral in Equation 30 using the Taylor expansion of e^{-Fv} or

$$e^{-Fv} = I - Fv + \frac{1}{2}F^2v^2 - \frac{1}{3!}F^3v^3 + \cdots = \sum_{k=0}^{\infty}(-1)^k\frac{v^k}{k!}F^k\,. \tag{34}$$

Thus using this series we have that the integral factor in Equation 30 becomes

$$\begin{aligned}\int_0^{\Delta t} e^{-Fv}dv &= \int_0^{\Delta t}\sum_{k=0}^{\infty}(-1)^k\frac{v^k}{k!}F^k dv = \sum_{k=0}^{\infty}(-1)^k\frac{\Delta t^{k+1}}{(k+1)!}F^k \\ &= \left[I - \frac{1}{2}\Delta tF + \frac{1}{3!}\Delta t^2F^2 - \cdots\right]\Delta t\,.\end{aligned}$$

With this Equation 29 becomes

$$\begin{aligned}\Delta x(t_k) &= \Phi(\Delta t)\Delta x(t_{k-1}) \\ &\quad + \Phi(\Delta t)\left[I - \frac{1}{2}\Delta tF + \frac{1}{3!}\Delta t^2F^2 - \cdots\right][G\Delta w(t_{k-1}) + L\Delta w(t_{k-1})]\Delta t \\ &= \Phi(\Delta t)\Delta x(t_{k-1}) + \Gamma(\Delta t)\Delta w(t_{k-1}) + \Lambda(\Delta t)\Delta w(t_{k-1})\,.\end{aligned}$$

This expression defines $\Gamma(\Delta t)$ and $\Lambda(\Delta t)$ when F is not invertible. In particular we have

$$\Gamma(\Delta t) \equiv \Phi(\Delta t)\left[I - \frac{1}{2}\Delta tF + \frac{1}{3!}\Delta t^2F^2 - \cdots\right]G\Delta t \tag{35}$$
$$\Lambda(\Delta t) \equiv \Phi(\Delta t)\left[I - \frac{1}{2}\Delta tF + \frac{1}{3!}\Delta t^2F^2 - \cdots\right]L\Delta t\,. \tag{36}$$

Notes on Spectral Density Functions of Random Process

In this section of these notes we discuss the spectral density functions of two very important signals. The first is the *white noise process* and the second is a *Markov process*.

For **white noise process** we have the autocovariance function given by

$$\phi_{xx}(\tau) = \phi_{xx}(0)\delta(\tau)\,, \tag{37}$$

Using the definition of the power spectral density $\Phi_{xx}(\omega)$ as the Fourier transform of the autocorrelation function $\phi_{xx}(\tau)$ we find

$$\Phi_{xx}(\omega) = \phi_{xx}(0)\int_{-\infty}^{\infty}\delta(\tau)e^{-j\omega\tau}d\tau = \phi_{xx}(0)\,, \tag{38}$$

showing that the power spectral density function for white noise, $\Phi_{xx}(\omega)$, is a constant function. Computing the the autocovariance function, by taking the inverse Fourier transform of the power spectral density, and then evaluating the autocovariance function at a lag of zero ($\tau = 0$) we get the variance of the process. As an equation this is

$$\sigma_x^2 = \frac{1}{\pi}\int_0^\infty \Phi_{xx}(\omega)d\omega\,. \tag{39}$$

We now notice a difficulty with this relationship for a white noise process. In the above equation the left-hand-side is a constant, σ_x^2, while using Equation 38 the right-hand-side can not be a convergent integral if $\Phi_{xx}(\omega)$ is a nonzero constant. As a *fix* for this problem we argue that for any realizable physical system there must be an *upper most* frequency at which the *most* oscillatory disturbance propagates. In other wards a physical system cannot be made to respond to arbitrary large frequencies. Thus we turn the above procedure where we went from $\phi_{xx}(\tau)$ to $\Pi_{xx}(\omega)$, into one where we *specify* $\Phi_{xx}(\omega)$ (based on arguments above) and from that derive the autocovariance function $\phi_{xx}(\tau)$.

To do this if we truncate $\Phi_{xx}(\omega)$ at some upper bound say, ω_B, then we have

$$\Phi_{xx}(\omega) = \begin{cases} \Phi & |\omega| \le \omega_B \\ 0 & |\omega| > \omega_B \end{cases}\,, \tag{40}$$

Then with this power spectral density function Equation 39 then becomes

$$\sigma_x^2 = \frac{1}{\pi}\Phi\omega_B \quad \text{or} \quad \Phi = \pi\frac{\sigma_x^2}{\omega_B}\,. \tag{41}$$

Given a functional form for the power spectral density $\Phi_{xx}(\omega)$ we can take its inverse Fourier transform to determine $\phi_{xx}(\tau)$ as

$$\begin{aligned}\phi_{xx}(\tau) &= \frac{1}{2\pi}\int_{-\infty}^{\infty}\Phi_{xx}(\omega)e^{j\omega\tau}d\omega = \frac{\Phi}{2\pi}\int_{-\omega_B}^{\omega_B} e^{+j\omega\tau}d\omega \\ &= \frac{\Phi}{2\pi}\left(\frac{e^{j\omega\tau}}{j\tau}\right|_{-\omega_B}^{\omega_B} = \frac{\Phi}{2\pi j\tau}(e^{j\tau\omega_B} - e^{-j\tau\omega_B}) \\ &= \frac{\Phi}{\pi\tau}\sin(\tau\omega_B)\,. \end{aligned} \tag{42}$$

The other process we want to study is the **Markov process**. From the book a Markov process has an autocovariance function given by

$$\phi_{xx}(\tau) = \sigma_w^2 e^{f|\tau|}\,, \tag{43}$$

here $f < 0$. Then with this autocovariance function we find the power spectral density

function for a Markov process given by

$$\begin{aligned}\Phi_{xx}(\omega) &= \int_{-\infty}^{\infty} \phi_{xx}(\tau) e^{-j\omega\tau} d\tau = \sigma_w^2 \int_{-\infty}^{\infty} e^{f|\tau|} e^{-jw\tau} d\tau \\ &= \sigma_w^2 \left[\int_{-\infty}^{0} e^{-f\tau} e^{-j\omega\tau} d\tau + \int_0^{\infty} e^{f\tau} e^{-j\omega\tau} d\tau \right] = \sigma_w^2 \left[\int_0^{\infty} e^{f\tau} e^{j\omega\tau} d\tau + \int_0^{\infty} e^{f\tau} e^{-j\omega\tau} d\tau \right] \\ &= \sigma_w^2 \left[\int_0^{\infty} e^{(f+j\omega)\tau} d\tau + \int_0^{\infty} e^{(f-j\omega)\tau} d\tau \right] = \sigma_w^2 \left[\frac{e^{(f+j\omega)\tau}}{f+j\omega} \Big|_0^{\infty} + \frac{e^{(f-j\omega)\tau}}{f-j\omega} \Big|_0^{\infty} \right] \\ &= \sigma_w^2 \left[-\frac{1}{f+j\omega} - \frac{1}{f-j\omega} \right] = -\sigma_w^2 \left[\frac{f-j\omega+f+j\omega}{f^2+\omega^2} \right] \\ &= -\frac{2f}{f^2+\omega^2} \sigma_w^2 \,. \end{aligned} \tag{44}$$

Because we have defined $f < 0$ the above expression must have the negative sign otherwise $\Phi_{xx}(\omega)$ will not be positive.

Notes on the quasistatic equilibrium example

When we consider the *equilibrium* point of the system

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} a & b \\ 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} c \\ 0 \end{bmatrix} u \,.$$

For fixed $u = u^*$ and when $b \neq 0$ we have a constant equilibrium solution for $\begin{bmatrix} x_1^* & x_2^* \end{bmatrix}^T$. If $b = 0$ then for a fixed control $u = u^*$ the system is

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} a & 0 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} c \\ 0 \end{bmatrix} u^* \,.$$

The steady-state solution for the first equation is given by

$$0 = a x_1^* + c u^* \,,$$

or $x_1^* = -\frac{c}{a} u^*$ and the steady-state solution to the second equation is given by $\dot{x}_2^* = x_1^*$. Thus solving for $x_2^*(t)$ by integrating we find

$$x_2^*(t) = x_2(0) + \int_0^t x_1^*(\tau) d\tau = x_2(0) - \frac{c}{a} u^* t \,. \tag{45}$$

Since the solution for $x_2^*(t)$ is not a constant but depends on t we call this a quasistatic equilibrium solution.

Notes on the cost function J when $\tilde{u}(t) = -C\tilde{x}$

If we introduce an objective function J that depends on the perturbed state $\tilde{x}$ and the perturbed control $\tilde{u}$ as

$$J = \frac{1}{2} \int_0^{\infty} [\tilde{x}^T(t) Q \tilde{x}(t) + \tilde{u}^T(t) R \tilde{u}(t)] dt \,,$$

then when our control $\tilde{u}$ is proportional to our perturbed state $\tilde{x}$ as

$$\tilde{u} = -C\tilde{x}\,,$$

in that we want to "push" the state back to the equilibrium value, then we see that J becomes

$$J = \frac{1}{2}\int_0^\infty [\tilde{x}^T(t)Q\tilde{x}(t) + \tilde{x}^T(t)C^T RC\tilde{x}(t)]dt = \frac{1}{2}\int_0^\infty [\tilde{x}^T(t)(Q + C^T RC)\tilde{x}(t)]dt\,,$$

or the integral of a quadratic form with a matrix

$$Q + C^T RC\,.$$

Notes on Reachability, Controllability, and Stability

In this section of these notes we discuss reachability and controllability conditions. The book gives a very high level overview of these concepts and what follows are simply some notes that I made to myself as I read these sections.

If we consider linearizing our state $x(t)$ solution around a base trajectory or state $x_0(t)$ as $x(t) = x_0(t) + \Delta x(t)$ where $\Delta x(t)$ satisfies the linearized dynamical system

$$\frac{d\Delta x}{dt} = F(t)\Delta x(t) + G(t)\Delta u(t)\,.$$

Then the solution for $\Delta x(t)$ given the initial condition $\Delta x(t_0)$ is

$$\Delta x(t) = \Phi(t, t_0)\Delta x(t_0) + \int_{t_0}^{t_f} \Phi(t, \tau)G(\tau)\Delta u(\tau)d\tau\,, \tag{46}$$

and out total solution $x(t)$ is then

$$x(t) = x_0(t) + \Phi(t, t_0)\Delta x(t_0) + \int_{t_0}^{t} \Phi(t, \tau)G(\tau)\Delta u(\tau)d\tau\,, \tag{47}$$

Note that this point the system matrices F and G can be time dependent. Then for *local* controllability we must have a control that enables us to make $x(t_f) = 0$. As a condition that must hold for this to be true we introduce the *controllability Grammian matrix*, $\mathbb{M}$, as

$$\mathbb{M}(t_f, t_0) = \int_{t_0}^{t_f} \Phi(t_f, \tau)G(\tau)G(\tau)^T\Phi^T(t_f, \tau)d\tau\,. \tag{48}$$

Lets assume that $\mathbb{M}$ is invertible and consider a control, $\Delta u(t)$, given by

$$\Delta u(t) = G^T(t)\Phi^T(t_f, t)\mathbb{M}^{-1}(t_f, t_0)[-x_0(t_f) - \Phi(t_f, t_0)\Delta x(t_0)]\,. \tag{49}$$

Lets define Γ to be the vector $\Gamma \equiv -x_0(t_f) - \Phi(t_f, t_0)\Delta x(t_0)$, then in Equation 47 we find

$$\begin{aligned} x(t) &= x_0(t) + \Phi(t, t_0)\Delta x(t_0) + \int_{t_0}^{t} \Phi(t, \tau)G(\tau)G^T(\tau)\Phi(t_f, \tau)^T\mathbb{M}^{-1}(t_f, t_0)\Gamma d\tau \\ &= x_0(t) + \Phi(t, t_0)\Delta x(t_0) + \left[\int_{t_0}^{t} \Phi(t, \tau)G(\tau)G^T(\tau)\Phi(t_f, \tau)^T d\tau\right]\mathbb{M}^{-1}(t_f, t_0)\Gamma\,. \end{aligned}$$

Under this control, when we let $t = t_f$, since the integral term in brackets above becomes $\mathbb{M}(t_f, t_0)$ we see that $x(t_f)$ is given by

$$x(t_f) = x_0(t_f) + \Phi(t_f, t_0)\Delta x(t_0) + \Gamma = 0\,.$$

In summary then, to be able to guarantee a control that takes us from x_0 to $x(t_f) = 0$ we must have the controllability Grammian matrix given by Equation 48 nonsingular. In that case the control given by Equation 49 will do the desired job.

In the above the system matrices F and G can be time dependent resulting in a $\Phi(t_f, t_0)$ that in general can be arbitrary. If in fact our system is linear and *time-invariant* so that F and G don't depend on t then Φ takes the special form

$$\Phi(t_f, t_0) = e^{F(t_f - t_0)}\,.$$

With this in Equation 46 and taking $\Delta x(t_f) = 0$ (assuming our system is controllable) and taking $t_0 = 0$ we have

$$x(t_f) = 0 = e^{F(t_f - 0)}x(0) + \int_0^{t_f} e^{F(t-\tau)}Gu(\tau)d\tau = e^{Ft_f}\left[x(0) + \int_0^{t_f} e^{-F\tau}Gu(\tau)d\tau\right]\,.$$

Use the Cayley-Hamilton theorem to write $e^{-F\tau}$ as

$$e^{-F\tau} = r_0 I_n + r_1 F + r_2 F^2 + \cdots + r_{n-1}F^{n-1}\,,$$

where the r_i are functions that depend on the eigenvalues of F and the value of τ. Using this and the above we get

$$\begin{aligned} -x(0) &= G\int_0^{t_f} r_0(\tau)u(\tau)d\tau + FG\int_0^{t_f} r_1(\tau)u(\tau)d\tau + \cdots + F^{n-1}G\int_0^{t_f} r_{n-1}(\tau)u(\tau)d\tau \\ &= \begin{bmatrix} G & FG & F^2G & \cdots & F^{n-1}G \end{bmatrix} \begin{bmatrix} \int_0^{t_f} r_0(\tau)u(\tau)d\tau \\ \int_0^{t_f} r_1(\tau)u(\tau)d\tau \\ \int_0^{t_f} r_2(\tau)u(\tau)d\tau \\ \vdots \\ \int_0^{t_f} r_{n-1}(\tau)u(\tau)d\tau \end{bmatrix}\,. \end{aligned}$$

We will call the stacked matrix of G's and F's the controllability matrix

$$\mathbb{C} \equiv \begin{bmatrix} G & FG & F^2G & \cdots & F^{n-1}G \end{bmatrix}\,. \tag{50}$$

In theory at least one can pick a function $u(\tau)$ such that we can drive all components of x to zero if $\mathbb{C}$ *is of rank* n.

We now briefly talk about the concept of *observability*, which has to do with how well we can reconstruct x based on the measurements, y, we take. We consider *local* observability in that we will Taylor expand about an equilibrium solution where the perturbed solution (here denoted x) will have a dynamic equation and a measurement equation given by

$$\begin{aligned} x(\tau) &= \Phi(\tau, t_0)x(t_0) \\ y(\tau) &= H(\tau)\Phi(\tau, t_0)x(t_0)\,, \end{aligned} \tag{51}$$

Note that we are considering the case where the system matrices are time varying. Then the condition for observability can be obtained by multiplying both sides of Equation 51 by $\Phi^T(\tau, t_0)H^T(\tau)$ and integrating from t_0 to t_f to get

$$\int_{t_0}^{t_f} \Phi^T(\tau, t_0)H^T(\tau)y(\tau)d\tau = \left[\int_{t_0}^{t_f} \Phi^T(\tau, t_0)H^T(\tau)H(\tau)\Phi(\tau, t_0)d\tau\right] x(t_0)\,.$$

Thus if we define the **observability Grammian**, $\mathbb{N}$, as

$$\mathbb{N} \equiv \int_{t_0}^{t_f} \Phi^T(\tau, t_0)H^T(\tau)H(\tau)\Phi(\tau, t_0)d\tau\,. \tag{52}$$

Then our system is observable if $\mathbb{N}$ is nonsingular. This is the condition needed for linear time-*varying* systems.

If we have a time-*invariant* system then we can apply the fact that controllability and observability are dual problems. To use this result we recognize that the *controllability* of the dual problem is the observability condition of the primal problem. The dual to the dynamic equation is

$$\dot{x}(t) = -F^T x(t) + H^T u(t)\,. \tag{53}$$

Here since we are in the time-invariant case F and G are independent of time. Then writing the controllability matrix $\mathbb{C}$ in terms of this dual problem we have

$$\mathbb{O} = \left[\begin{array}{ccccc} H^T & F^T H^T & (F^T)^T H^T & \cdots & (F^T)^{n-1}H^T \end{array}\right]\,. \tag{54}$$

If this matrix $\mathbb{O}$ is of rank n then our time-invariant system is observable.

As a final result comment on the dual problem, we recall that the **modal matrix** D is the matrix that diagonalizes the system matrix F as $\Lambda = D^{-1}FD$. From this we can derive the modal matrix for the dual system Equation 53 with the following manipulations. We first take the transpose of both sides

$$\Lambda = D^{-1}FD \quad \text{so} \quad \Lambda^T = D^T F^T D^{-T} \quad \text{so} \quad -\Lambda = (D^{-T})^{-1}(-F^T)(D^{-T})\,,$$

showing that the matrix that diagonalizes $-F^T$ is D^{-T}. Since now we know the modal matrix for the dual problem we can enforce complete modal controllability for the dual problem by requiring that there are *no zero rows* in the product of the inverse of the dual problem's modal matrix D^{-T}, and the dual problem's control coupling matrix H^T or

$$(D^{-T})^{-1}H^T = D^T H^T\,.$$

Notes on discrete time systems

If we look for the steady-state solution x^* to our time-invariant discrete time system

$$x_{k+1} = \Phi x_k + \Gamma u_k + \Lambda w_k \quad \text{for} \quad k \geq 0\,, \tag{55}$$

when the inputs u and w are *fixed* at u^* and w^* respectively then we must have x^* given by

$$x^* = (I_n - \Phi)^{-1}(\Gamma u^* + \Lambda w^*) \,. \tag{56}$$

If the discrete system is the result of discretizing a continuous system then Γ and Λ are given by Equations 35 and 36 respectively. In that case we find that x^* becomes

$$x^* = -F^{-1}Gu^* - F^{-1}Lw^* = -F^{-1}(Gu^* + Lw^*) \,. \tag{57}$$

Notes on Example 2.6-1: Transfer functions for aircraft pitching motion

For this example α is the pitch angle (angle of attack) and q is the pitching rate (the time derivative of the angle of attack). For the given two dimensional time domain system given in the book by taking the Laplace transform of the matrix system and dropping the transfer function of the noise (α_w) term we get

$$s \begin{bmatrix} q(s) \\ \alpha(s) \end{bmatrix} - \begin{bmatrix} q(0) \\ \alpha(0) \end{bmatrix} = \begin{bmatrix} M_q & M_\alpha \\ 1 & -\frac{L_\alpha}{V} \end{bmatrix} \begin{bmatrix} q(s) \\ \alpha(s) \end{bmatrix} + \begin{bmatrix} M_{\delta E} & M_{\delta F} \\ -\frac{L_{\delta E}}{V} & -\frac{L_{\delta F}}{V} \end{bmatrix} \begin{bmatrix} \delta E(s) \\ \delta F(s) \end{bmatrix} .$$

Dropping the initial conditions $\begin{bmatrix} q(0) \\ \alpha(0) \end{bmatrix}$ to study the steady-state or long term forcing response only we get

$$\begin{bmatrix} s - M_q & -M_\alpha \\ -1 & s + \frac{L_\alpha}{V} \end{bmatrix} \begin{bmatrix} q(s) \\ \alpha(s) \end{bmatrix} = \begin{bmatrix} M_{\delta E} & M_{\delta F} \\ -\frac{L_{\delta E}}{V} & -\frac{L_{\delta F}}{V} \end{bmatrix} \begin{bmatrix} \delta E(s) \\ \delta F(s) \end{bmatrix} .$$

Thus when we solve for $\begin{bmatrix} q(s) \\ \alpha(s) \end{bmatrix}$ we get

$$\begin{bmatrix} q(s) \\ \alpha(s) \end{bmatrix} = \begin{bmatrix} s - M_q & -M_\alpha \\ -1 & s + \frac{L_\alpha}{V} \end{bmatrix}^{-1} \left\{ \begin{bmatrix} M_{\delta E} \\ -\frac{L_{\delta E}}{V} \end{bmatrix} \delta E(s) + \begin{bmatrix} M_{\delta F} \\ -\frac{L_{\delta F}}{V} \end{bmatrix} \delta F(s) \right\} .$$

Considering the input-output relationship that follows from an input of $\delta E(s)$ followed by an output of $\alpha(s)$. If we multiply the above by $\begin{bmatrix} 0 & 1 \end{bmatrix}$ to extract out the component $\alpha(s)$ we get

$$\frac{\alpha(s)}{\delta E(s)} = \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} s - M_q & -M_\alpha \\ -1 & s + \frac{L_\alpha}{V} \end{bmatrix}^{-1} \begin{bmatrix} M_{\delta E} \\ -\frac{L_{\delta E}}{V} \end{bmatrix} .$$

The inverse of the given $sI_n - F$ coefficient matrix using matrix adjoint theory is given by

$$\begin{bmatrix} s - M_q & -M_\alpha \\ -1 & s + \frac{L_\alpha}{V} \end{bmatrix}^{-1} = \frac{1}{s^2 - \left(M_q - \frac{L_\alpha}{V}\right) s - M_\alpha} \begin{bmatrix} s + \frac{L_\alpha}{V} & M_\alpha \\ 1 & s - M_q \end{bmatrix} ,$$

so we get

$$\begin{aligned} \frac{\alpha(s)}{\delta E(s)} &= \begin{bmatrix} 0 & 1 \end{bmatrix} \left(\frac{1}{s^2 - \left(M_q - \frac{L_\alpha}{V}\right) s - M_\alpha} \right) \begin{bmatrix} s + \frac{L_\alpha}{V} & M_\alpha \\ 1 & s - M_q \end{bmatrix} \begin{bmatrix} M_{\delta E} \\ -\frac{L_{\delta E}}{V} \end{bmatrix} \\ &= \frac{M_{\delta E} - (s - M_q)\frac{L_{\delta E}}{V}}{s^2 - \left(M_q - \frac{L_\alpha}{V}\right) s - M_\alpha} = -\frac{L_{\delta E}}{V} \left(\frac{s - M_q - \frac{M_{\delta E} V}{L_{\delta E}}}{s^2 - \left(M_q - \frac{L_\alpha}{V}\right) s - M_\alpha} \right) \end{aligned}$$

Lets define ω_n and ξ using $\omega_n^2 = -M_\alpha$ and $2\xi\omega_n = \frac{L_\alpha}{V} - M_q$ so that

$$\frac{\alpha(s)}{\delta E(s)} = -\frac{L_{\delta E}}{V}\left(\frac{s - (M_q + \frac{M_{\delta E}V}{L_{\delta E}})}{s^2 + 2\xi\omega_n s + \omega_n^2}\right) .$$

From the definition of ω_n and ξ we have in terms of more primitive variables that

$$\omega_n = \sqrt{-M_\alpha}$$
$$\xi = \frac{1}{2\omega_n}\left(\frac{L_\alpha}{V} - M_q\right) = \frac{1}{2\sqrt{-M_\alpha}}\left(\frac{L_\alpha}{V} - M_q\right) .$$

If we assume that $M_{(\cdot)} < 0$ we see that ω_n is a real number. Next to consider the poles of this transfer function compute the roots of the quadratic polynomial $s^2 + 2\xi\omega_n s + \omega_n^2 = 0$, which are given by

$$s = \frac{-2\xi\omega_n \pm \sqrt{4\xi^2\omega_n^2 - 4\omega_n^2}}{2} = -\xi\omega_n \pm \omega_n\sqrt{\xi^2 - 1}\,.$$

If we assume that $\frac{L_\alpha}{V} - M_q < 2\sqrt{-M_\alpha}$ then from the definition of ξ we have that

$$\xi = \frac{1}{2\sqrt{-M_\alpha}}\left(\frac{L_\alpha}{V} - M_q\right) < 1\,,$$

showing that the poles of the denominator are complex since $\xi^2 - 1 < 0$. The one root at the location

$$M_q + \frac{VM_{\delta E}}{L_{\delta E}} < 0\,,$$

is less than zero when we take $L_{(\cdot)} > 0$.

Notes on the root locus

The transfer function algebra can be obtained by walking clockwise around the presented circuit. First the signals $y_c(s)$ and $-y(s)$ combine before entering a linear time invariant system with transfer function $Y(s)$, where the output is again $y(s)$, thus we have

$$(y_c(s) - y(s))Y(s) = y(s)\,.$$

Solving for "output over input" or $\frac{y(s)}{y_c(s)}$ we find

$$\frac{y(s)}{y_c(s)} = \frac{Y(s)}{1 + Y(s)}\,. \tag{58}$$

Problem Solutions

Section 2.1 Problem 1 (stationary points)

For these problems we plot the function $J_i(u)$ in Figure 1. With these plots we can verify the classification of each stationary point.

Part (a): For the given $J(u)$ we find $J'(u) = 15 + 10u = 0$ so $u = -\frac{3}{2}$, and $J''(u) = 10 > 0$, thus the point $u = -\frac{3}{2}$ is a minimum.

Part (b): For this expression for $J(u)$ we find extreme points given by

$$J'(u) = 4 - 12u + 30u^2 = 0\,,$$

or after we divide by 2 and us the quadratic formula

$$u = \frac{6 \pm \sqrt{36 - 120}}{30} = \frac{3 \pm \sqrt{-21}}{15}\,,$$

which are both complex numbers. Thus $J'(u)$ has no real roots and $J'(u)$ is one sign. We see that $J'(0) = 4 \geq 0$ so $J(u)$ has no maximum or minimum on $-\infty < u < +\infty$.

Part (c): First expand $J(u)$ as

$$J(u) = (u^2 + u - 2)(u - 3) = u^3 - 2u^2 - 5u + 6\,,$$

Thus $J'(u)$ is given by

$$J'(u) = 3u^2 - 4u - 5 = 0\,.$$

Thus we get for u when we solve this

$$u = \frac{4 \pm \sqrt{16 - 4(3)(-5)}}{2(3)} = \frac{4 \pm \sqrt{76}}{6} = -0.786\,,\ 2.119\,.$$

We have $J''(u) = 6u - 4$. Thus

$$\begin{aligned} J''(u) > 0 \quad &\text{if} \quad u > \frac{2}{3} \\ J''(u) < 0 \quad &\text{if} \quad u < \frac{2}{3}\,. \end{aligned}$$

Thus $u_1 = -0.786$ is a maximum and $u_2 = 2.119$ is a minimum.

Part (d): For this J expression we have

$$J'(u) = e^u - e^{-u} = 0\,,$$

or $e^u = e^{-u}$ so $u = -u$ or $u = 0$ is the only solution. Then the second derivative is given by

$$J''(u) = e^u + e^{-u} > 0\,,$$

for all u. Thus $u = 0$ is a minimum.

Section 2.1 Problem 2 (more stationary points)

Part (a): For $J(u)$ given by

$$J(u) = (u_1^2 + 3u_1 - 4)(u_2^2 - u_2 + 6)\,,$$

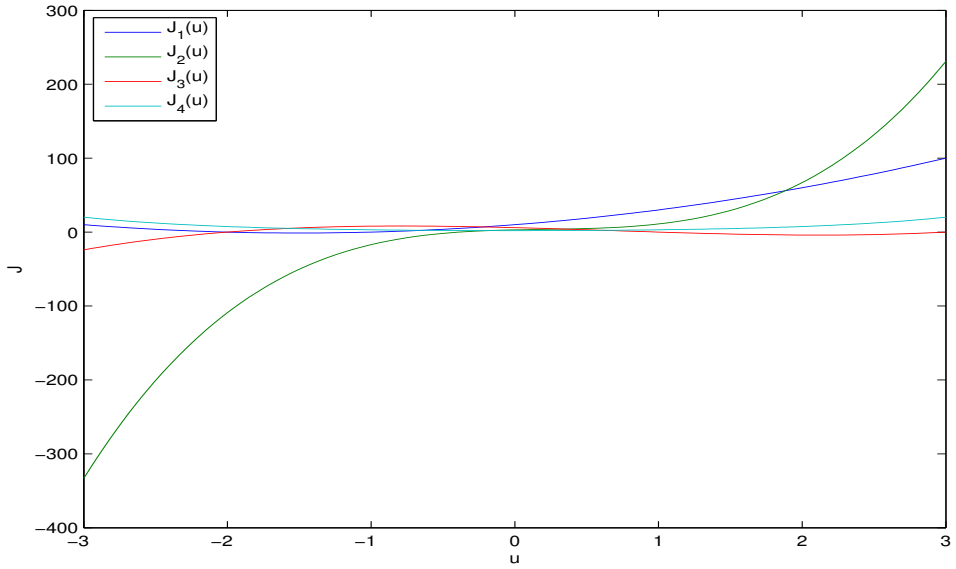

Figure 1: Plots of the functions $J_i(u)$ as a function of u.

we find setting its first derivative to zero the following equations

$$\frac{\partial J}{\partial u_1} = (2u_1 + 3)(u_2^2 - u_2 + 6) = 0 \quad \text{and}$$
$$\frac{\partial J}{\partial u_2} = (u_1^2 + 3u_1 - 4)(2u_2 - 1) = 0\,.$$

If $u_1 = -\frac{3}{2}$ and $u_2 = \frac{1}{2}$ then these equations are satisfied. These two solutions were found by setting the linear factors in the above equations equal to zero. If we seek another solution to $\frac{\partial J}{\partial u_1} = 0$ say by picking $u_2^2 - u_2 + 6 = 0$ so that have that

$$u_2 = \frac{1 \pm \sqrt{1 - 4(1)(6)}}{2} = \frac{1 \pm \sqrt{-23}}{2}\,,$$

which is imaginary showing that the factor $u_2^2 - u_2 + 6$ is never zero. We conclude that $(u_1, u_2) = \left(-\frac{3}{2}, \frac{1}{2}\right)$ is the only stationary point. Lets compute the Hessian of J at the point $\left(-\frac{3}{2}, \frac{1}{2}\right)$. To do this we need the second derivatives

$$\frac{\partial^2 J}{\partial {u_1}^2} = 2(u_2^2 - u_2 + 6)$$
$$\frac{\partial^2 J}{\partial u_2 \partial u_1} = (2u_1 + 3)(2u_2 - 1)$$
$$\frac{\partial^2 J}{\partial {u_2}^2} = 2(u_1^2 + 3u_1 - 4)\,.$$

Thus the Hessian is then

$$H(u_1, u_2) = \begin{bmatrix} 2(u_2^2 - u_2 + 6) & (2u_1 + 3)(2u_2 - 1) \\ (2u_1 + 3)(2u_2 - 1) & 2(u_1^2 + 3u_1 - 4) \end{bmatrix}.$$

When we evaluate this at the point $(u_1, u_2) = \left(-\frac{3}{2}, \frac{1}{2}\right)$ we get

$$H = \begin{bmatrix} 23/2 & 0 \\ 0 & -25/2 \end{bmatrix} .$$

This matrix has eigenvalues of two different signs and thus the point $(u_1, u_2) = \left(-\frac{3}{2}, \frac{1}{2}\right)$ is a saddle point.

Part (b): Since we have only one equality constraint $u_1 - 2u_2 = 0$, we notationally separate the two unknowns u_1 and u_2 into the two parts x and u as $x = u_1$ and $u = u_2$. In terms of x and u we next form the augmented function J_A

$$J_A(x, u) = (x^2 + 3x - 4)(u^2 - u + 6) + \lambda(x - 2u) ,$$

and the constraint function $f(x, u) = x - 2u = 0$. We then need derivatives

$$\begin{aligned} \frac{\partial J}{\partial x} &= (2x + 3)(u^2 - u + 6) \\ \frac{\partial f}{\partial x} &= 1 \\ \frac{\partial J}{\partial u} &= (x^2 + 3x - 4)(2u - 1) \\ \frac{\partial f}{\partial u} &= -2 . \end{aligned}$$

Then Equation 7 and Equation 8 are given by

$$\begin{aligned} (x^2 + 3x - 4)(2u - 1) - (2x + 3)(u^2 - u + 6)(-2) &= 0 \\ x - 2u &= 0 . \end{aligned}$$

If we put $x = 2u$ into the first equation we get

$$16u^3 + 6u^2 + 28u + 40 = 0 .$$

Solving this and looking for the real root gives $u = -1.03$ so $x = -2.06$.

Section 2.1 Problem 3 (optimal values under a constraint)

We need to construct the derived Lagrangian equation

$$\frac{\partial J}{\partial \mathbf{u}} - \frac{\partial J}{\partial \mathbf{x}} \left(\frac{\partial \mathbf{f}}{\partial \mathbf{x}} \right)^{-1} \frac{\partial \mathbf{f}}{\partial \mathbf{u}} = 0 . \tag{59}$$

Since J in this problem is given by the expression

$$J = \frac{1}{2}(2x_1^2 + 4x_2^2) + u^2 = x_1^2 + 2x_2^2 + u^2 ,$$

With $\mathbf{u} = u$ and $\mathbf{x} = \begin{bmatrix} x_1 & x_2 \end{bmatrix}$ we have

$$\frac{\partial J}{\partial \mathbf{u}} = 2u$$
$$\frac{\partial J}{\partial \mathbf{x}} = \begin{bmatrix} 2x_1 & 4x_2 \end{bmatrix}.$$

With $\mathbf{f} = \begin{bmatrix} x_2 \\ -x_1 - x_2 + u \end{bmatrix}$ we find the derivatives given by

$$\frac{\partial \mathbf{f}}{\partial \mathbf{u}} = \frac{\partial \mathbf{f}}{\partial u} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \quad \text{and} \quad \frac{\partial \mathbf{f}}{\partial \mathbf{x}} = \begin{bmatrix} 0 & 1 \\ -1 & -1 \end{bmatrix}.$$

Then Equation 59 becomes

$$2u - \begin{bmatrix} 2x_1 & 4x_2 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ -1 & 1 \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = 0\,.$$

Since $\begin{bmatrix} 0 & 1 \\ -1 & 1 \end{bmatrix}^{-1} = \begin{bmatrix} 1 & -1 \\ 1 & 0 \end{bmatrix}$ this is given by

$$2u - \begin{bmatrix} 2x_1 & 4x_2 \end{bmatrix} \begin{bmatrix} -1 \\ 0 \end{bmatrix} = 0\,,$$

or $2u + 2x_1 = 0$. This equation with constraint $\mathbf{f} = 0$ gives the following linear system for x_1, x_2, and u

$$\begin{bmatrix} 0 & 1 & 0 \\ -1 & -1 & 1 \\ 2 & 0 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ u \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}.$$

The first equation gives $x_1 = 0$ and then the third equation gives $u = 0$ and then the second equation gives $x_2 = 0$. This solution certainly satisfies the constraint $\mathbf{f} = 0$ and gives a cost of $J = 0$. We might have been able to directly guess at this solution without going through all of this work.

Section 2.1 Problem 4 (optimal values under a constraint continued)

For the J given here

$$\begin{aligned} J &= \frac{1}{2} \begin{bmatrix} x_1 & x_2 \end{bmatrix} \begin{bmatrix} 2x_1 + 2x_2 \\ 2x_1 + 4x_2 \end{bmatrix} + u^2 \\ &= \frac{1}{2}(2x_1^2 + 2x_1x_2 + 2x_1x_2 + 4x_2^2) + u^2 \\ &= x_1^2 + 2x_1x_2 + 2x_2^2 + u^2\,. \end{aligned}$$

Now as in the previous case we have $\mathbf{u} = u$ with $\mathbf{x} = \begin{bmatrix} x_1 & x_2 \end{bmatrix}$ and we have

$$\frac{\partial J}{\partial \mathbf{u}} = 2u$$
$$\frac{\partial J}{\partial \mathbf{x}} = \begin{bmatrix} 2x_1 + 2x_2 & 2x_1 + 4x_2 \end{bmatrix}.$$

Then Equation 59 becomes

$$2u - \begin{bmatrix} 2x_1 & 4x_2 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ -1 & 1 \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = 0\,.$$

Since $\begin{bmatrix} 0 & 1 \\ -1 & 1 \end{bmatrix}^{-1} = \begin{bmatrix} 1 & -1 \\ 1 & 0 \end{bmatrix}$ we get

$$2u - \begin{bmatrix} 2x_1 + 2x_2 & 2x_1 + 4x_2 \end{bmatrix} \begin{bmatrix} -1 \\ 0 \end{bmatrix} = 0\,,$$

or $2u + 2x_1 + 2x_2 = 0$. This equation with constraint $\mathbf{f} = 0$ gives the following linear system for x_1, x_2, and u

$$\begin{bmatrix} 0 & 1 & 0 \\ -1 & -1 & 1 \\ 2 & 2 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ u \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}\,.$$

The first equation gives $x_1 = 0$ and the second and third equations then becomes

$$\begin{bmatrix} -1 & 1 \\ 2 & 2 \end{bmatrix} \begin{bmatrix} x_2 \\ u \end{bmatrix} = 0\,.$$

Again the solution to this system is $u = 0$ and $x_2 = 0$. This solution could perhaps be seen from the form for J in that the matrix in the quadratic form $\begin{bmatrix} 2 & 2 \\ 2 & 4 \end{bmatrix}$ is positive definite thus we will have a minimum in (x_1, x_2) of $(0, 0)$.

Section 2.1 Problem 5 (derivatives)

The gradient vector in this book is defined as a *row* vector, so for the objective function J given by $J(x_1, x_2, u) = (a + bu^2)x_2^2$ we have with $\mathbf{u}^T = \begin{bmatrix} x_1 & x_2 & u \end{bmatrix}$ that

$$\frac{\partial J}{\partial \mathbf{u}} = \begin{bmatrix} \frac{\partial J}{\partial x_1} & \frac{\partial J}{\partial x_2} & \frac{\partial J}{\partial u} \end{bmatrix} = \begin{bmatrix} 0 & 2(a + bu^2)x_2 & 2bx_2^2 \end{bmatrix}\,.$$

The the Hessian of J is defined in general as

$$\frac{\partial^2 J}{\partial \mathbf{u}^2} = J_{\mathbf{uu}} = \begin{bmatrix} \frac{\partial^2 J}{\partial {u_1}^2} & \frac{\partial^2 J}{\partial u_1 \partial u_2} & \cdots & \frac{\partial^2 J}{\partial u_1 \partial u_m} \\ \vdots & & & \vdots \\ \frac{\partial^2 J}{\partial u_m \partial u_1} & \frac{\partial^2 J}{\partial u_m \partial u_2} & \cdots & \frac{\partial^2 J}{\partial {u_m}^2} \end{bmatrix}\,.$$

For this problem since there are *three* input variables x_1, x_2, and u this is

$$\frac{\partial^2 J}{\partial \mathbf{u}^2} = \begin{bmatrix} \frac{\partial^2 J}{\partial {x_1}^2} & \frac{\partial^2 J}{\partial x_1 \partial x_2} & \frac{\partial^2 J}{\partial x_1 \partial u} \\ \frac{\partial^2 J}{\partial x_2 \partial x_1} & \frac{\partial^2 J}{\partial {x_2}^2} & \frac{\partial^2 J}{\partial x_2 \partial u} \\ \frac{\partial^2 J}{\partial u \partial x_1} & \frac{\partial^2 J}{\partial u \partial x_2} & \frac{\partial^2 J}{\partial u^2} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 2(a + bu^2) & 4bux_2 \\ 0 & 4bux_2 & 2bx_2^2 \end{bmatrix}\,.$$

The Jacobian matrix associated with $\mathbf{f}$ has (i, j)th elements given by $\frac{\partial f_i}{\partial u_j}$ so we find it equal

$$f_{X,U} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \frac{\partial f_1}{\partial u} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \frac{\partial f_2}{\partial u} \end{bmatrix} = \begin{bmatrix} -1 & 2c(a + bu^2)x_2 & 2bucx_2^2 \\ 0 & du + \frac{e}{x_2^2} & dx_2 \end{bmatrix}\,.$$

Section 2.1 Problem 7 (minimization with inequality constraints)

For this problem we want to minimize $J = x^2 + 5u$ subject to $f(x,u) = x - 3u - 3 \leq 0$. One way we *might* be able to do this problem is to first attempt to perform the minimization unconditionally. That is to ignore the constraint, find the minimum over an unconstrained set of variables, and then see if the found minimum satisfies the given constraint. If it does we are done. To look for unconstrained minimization we consider

$$\frac{\partial J}{\partial x} = 2x = 0$$
$$\frac{\partial J}{\partial u} = 5 = 0\,.$$

Since the last equation is impossible to satisfy, we are not able to find an unconstrained solution. Since this did not work we now discuss a more formal way to solve constrained optimization problems.

Notes on optimization with inequality constraints

In this section of these notes we document at a very high level (without much motivation or background) how to solve constrained optimization problems. These notes can then be referenced, as needed, when working with specific optimization problems. The general optimization problem with inequality constraints is given by

$$\begin{aligned} &\text{minimize} \quad J(\theta) \\ &\text{subject to} \quad f_i(\theta) \geq 0 \quad \text{for} \quad i = 1, 2, \cdots, m\,. \end{aligned}$$

To solve this problem we first form the *Lagrangian*, $\mathcal{L}$, defined by

$$\mathcal{L}(\theta; \lambda) \equiv J(\theta) - \sum_{i=1}^{m} \lambda_i f_i(\theta)\,. \tag{60}$$

The variables λ_i in the above expression are called Lagrange multipliers. Using this definition, a set of *necessary* conditions for a local minimizer θ^* to exist is the following:

1. $\frac{\partial}{\partial \theta}\mathcal{L}(\theta^*; \lambda) = 0$.
2. $\lambda_i \geq 0$ for $i = 1, 2, \ldots, m$.
3. $\lambda_i f_i(\theta^*) = 0$ for $i = 1, 2, \ldots, m$.

These three conditions are called the *Karush-Kuhn-Tucker* or KKT conditions. The third conditions are called the *complementary slackness conditions*. A given complementary slackness condition say $\lambda_i f_i(\theta^*) = 0$ mean that when this product is zero and $\lambda_i \neq 0$ we have the original nonlinear constraint $f_i(\theta^*) \geq 0$ *active* i.e. at the optimal point θ^* it is the hard constraint $f_i(\theta^*) = 0$. Given these conditions we next ask how to use them to actually *find* the

optimal point θ^*. One approach, that might work for small problems, is to explicitly specify which nonlinear constraints we want to have active that is assume $f_i(\theta^*) = 0$, from some set of i. We can than solve the remaining equations for the respective Lagrange multipliers. To verify that we indeed have a solution we would then need to check that the values computed for these Lagrange multipliers were non-negative. This can be hard to do in general when there are many constraints, since there are many possible sets $f_i(\theta^*) = 0$ to consider. An alternative approach is to express the problem in its *Wolfe Dual Form*. This later form expresses the fact that in the situation where the objective function $J(\theta)$ is convex while the constraint functions $f_i(\theta)$ are concave then the above programming problem is equivalent to a simpler convex *maximization* programming problem

$$\begin{array}{cc} \text{maximize}_{\lambda \geq 0} & \mathcal{L}(\theta; \lambda) \\ \text{subject to} & \dfrac{\partial}{\partial \theta}\mathcal{L}(\theta; \lambda) = 0 \\ \text{and} & \lambda \geq 0\,. \end{array}$$

The benefit of this later formulation is that the relatively complicated nonlinear inequality constraints of the original problem, $f_i(\theta) \geq 0$, are replaced with the simpler equality constraint $\frac{\partial}{\partial \theta}\mathcal{L}(\theta; \lambda) = 0$ and a maximization over $\lambda \geq 0$. This later problem (if needed) can be solved with more standard convex programming codes.

To use the above notes we first convert the given inequality into an inequality of the form $g(x, u) \geq 0$ as

$$g(x, u) = -f(x, u) = -x + 3u + 3 \geq 0\,.$$

Then we form the Lagrangian

$$\mathcal{L}(x, u; \lambda) \equiv J(x, u) - \lambda(-x + 3u + 3)\,,$$

or

$$\mathcal{L}(x, u; \lambda) = x^2 + 5u^2 - \lambda(-x + 3u + 3)\,.$$

The necessary conditions are then that

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial x} &= 2x + \lambda = 0 \\ \frac{\partial \mathcal{L}}{\partial u} &= 10u - \lambda(3) = 0\,. \end{aligned}$$

Using the above to solve for x and u in terms of λ and then putting those expressions into the complementary slackness condition gives

$$\lambda\left(\frac{\lambda}{2} + \frac{9\lambda}{10} + 3\right) = 0\,.$$

This means that $\lambda = 0$ or $\lambda = -\frac{15}{7}$. If $\lambda = 0$ then $x = u = 0$ while if $\lambda = -\frac{15}{7}$ we have $x = \frac{15}{14}$ and $u = -\frac{9}{14}$. The point $(x, u) = (0, 0)$ clearly gives the smaller value for the objective function and is the desired minimum.

Section 2.1 Problem 8 (the minimum of $x(t)$)

We find the extreme points of $x(t)$ via the equation

$$\frac{dx}{dt} = \xi\omega_n e^{-\xi\omega_n t}\cos(\omega_n t + \phi) + \omega_n e^{-\xi\omega_n t}\sin(\omega_n t + \phi) = 0\,,$$

or

$$\xi\cos(\omega_n t + \phi) + \sin(\omega_n t + \phi) = 0\,,$$

or

$$\tan(\omega_n t + \phi) = -\frac{1}{\xi}\,,$$

To find the solutions to this equation we can plot the function $\tan(\omega_n t + \phi)$ as a function of t and the constant function $-\frac{1}{\xi}$ on the same graph. Their intersection is the solution. When $\omega_n = 1$ and $\phi = 0$ we plot $\tan(t)$ and we need to solve for $t = \tan^{-1}\left(-\frac{1}{\xi}\right)$.

Section 2.2 Problem 1 (determinants)

Part (a): Using Laplace expansion or cofactor expansion we expand about the first row to get

$$|F| = a\begin{vmatrix} d & 0 & 0 \\ 0 & e & 0 \\ 0 & 0 & f \end{vmatrix} - c\begin{vmatrix} b & 0 & 0 \\ 0 & e & 0 \\ 0 & 0 & f \end{vmatrix} = adef - cbef = ef(ad - bc)\,.$$

Note that we can consider this matrix as a block diagonal matrix with three blocks on the diagonal given by

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix}\,, \quad [e]\,, \quad \text{and} \quad [f]\,.$$

Then the determinant of F can be computed via the product of the determinants of the block elements. This gives

$$|F| = \begin{vmatrix} a & b \\ c & d \end{vmatrix}|e||f| = ef(ad - bc)\,,$$

the same was what we have above.

To use pivotal condensation to evaluate this determinant we can expand about the $(1,1)$ element (assumed nonzero) to get

$$|F| = \frac{1}{a^{4-2}}\begin{vmatrix} \begin{vmatrix} a & b \\ c & d \end{vmatrix} & \begin{vmatrix} a & 0 \\ c & 0 \end{vmatrix} \\ \begin{vmatrix} a & b \\ 0 & 0 \end{vmatrix} & \begin{vmatrix} a & 0 \\ 0 & e \end{vmatrix} \end{vmatrix} \begin{vmatrix} \begin{vmatrix} a & 0 \\ c & 0 \end{vmatrix} & \begin{vmatrix} a & 0 \\ 0 & 0 \end{vmatrix} \\ \begin{vmatrix} a & 0 \\ a & 0 \end{vmatrix} & \begin{vmatrix} a & 0 \\ 0 & f \end{vmatrix} \end{vmatrix} = \frac{1}{a^2}\begin{vmatrix} ad - bc & 0 & 0 \\ 0 & ae & 0 \\ 0 & 0 & af \end{vmatrix}\,.$$

To evaluate this determinant we can again use the method of pivotal condensation about the $(1,1)$ element assuming that $ad-bc \neq 0$. We find

$$|F| = \frac{1}{ad-bc}\left(\frac{1}{a^2}\right)\begin{vmatrix} \begin{vmatrix} ad-bc & 0 \\ 0 & ae \end{vmatrix} & \begin{vmatrix} ad-bc & 0 \\ 0 & 0 \end{vmatrix} \\ \begin{vmatrix} ad-bc & 0 \\ 0 & 0 \end{vmatrix} & \begin{vmatrix} ad-bc & 0 \\ 0 & af \end{vmatrix} \end{vmatrix}$$
$$= \frac{1}{a^2(ad-bc)}\begin{vmatrix} ae(ad-bc) & 0 \\ 0 & af(ad-bc) \end{vmatrix} = \frac{a^2ef(ad-bc)^2}{a^2(ad-bc)} = ef(ad-bc)\,,$$

the same result as before.

Part (b): The determinant of an upper/lower triangular matrix is the product of the diagonal elements so

$$|F| = 1 \cdot 3 \cdot 6 \cdot 10 = 180\,.$$

Performing a Laplace expansion about the first row for each matrix gives

$$1\begin{vmatrix} 3 & 0 & 0 \\ 5 & 6 & 0 \\ 8 & 9 & 10 \end{vmatrix} = 3\begin{vmatrix} 6 & 0 \\ 9 & 10 \end{vmatrix} = 3 \cdot 6 \cdot 10 = 180\,,$$

the same result. To use pivotal condensation to evaluate this determinant we can expand about the $(1,1)$ element to get

$$|F| = \frac{1}{1^{4-2}}\begin{vmatrix} \begin{vmatrix} 1 & 0 \\ 2 & 3 \end{vmatrix} & \begin{vmatrix} 1 & 0 \\ 2 & 0 \end{vmatrix} & \begin{vmatrix} 1 & 0 \\ 2 & 0 \end{vmatrix} \\ \begin{vmatrix} 1 & 0 \\ 4 & 5 \end{vmatrix} & \begin{vmatrix} 1 & 0 \\ 4 & 6 \end{vmatrix} & \begin{vmatrix} 1 & 0 \\ 4 & 0 \end{vmatrix} \\ \begin{vmatrix} 1 & 0 \\ 7 & 8 \end{vmatrix} & \begin{vmatrix} 1 & 0 \\ 7 & 9 \end{vmatrix} & \begin{vmatrix} 1 & 0 \\ 7 & 10 \end{vmatrix} \end{vmatrix} = \begin{vmatrix} 3 & 0 & 0 \\ 5 & 6 & 0 \\ 8 & 9 & 10 \end{vmatrix}.$$

Again expanding about the $(1,1)$ element gives

$$|F| = \frac{1}{3^{3-2}}\begin{vmatrix} \begin{vmatrix} 3 & 0 \\ 5 & 6 \end{vmatrix} & \begin{vmatrix} 3 & 0 \\ 5 & 0 \end{vmatrix} \\ \begin{vmatrix} 3 & 0 \\ 8 & 9 \end{vmatrix} & \begin{vmatrix} 3 & 0 \\ 8 & 10 \end{vmatrix} \end{vmatrix} = \frac{1}{3}\begin{vmatrix} 18 & 0 \\ 27 & 30 \end{vmatrix} = \frac{1}{3}18 \cdot 30 = 180\,,$$

the same value as before.

Part (c): Performing a Laplace expansion about the first row of the matrix gives

$$
\begin{aligned}
|F| &= 1\begin{vmatrix} 1 & -3 & 4 \\ -2 & 0 & -5 \\ 3 & 5 & 0 \end{vmatrix} + 2\begin{vmatrix} 1 & 0 & 4 \\ -2 & 3 & -5 \\ 3 & -4 & 0 \end{vmatrix} + 3\begin{vmatrix} 1 & 0 & -3 \\ -2 & 3 & 0 \\ 3 & -4 & 5 \end{vmatrix} \\
&= 1\begin{vmatrix} 0 & -5 \\ 5 & 0 \end{vmatrix} + 3\begin{vmatrix} -2 & -5 \\ 3 & 0 \end{vmatrix} + 4\begin{vmatrix} -2 & 0 \\ 3 & 5 \end{vmatrix} \\
&+ 2\begin{vmatrix} 3 & -5 \\ -4 & 0 \end{vmatrix} + 2\cdot 4\begin{vmatrix} -2 & 3 \\ 3 & -4 \end{vmatrix} \\
&+ 3\begin{vmatrix} 3 & 0 \\ -4 & 5 \end{vmatrix} - 3(3)\begin{vmatrix} -2 & 3 \\ 3 & -4 \end{vmatrix} \\
&= 25 + 3(15) + 4(-10) + 2(-20) + 8(8-9) + 3(15) - 9(8-9) = 36\,.
\end{aligned}
$$

To use pivotal condensation to evaluate this determinant we can expand about the $(2,1)$ element (since it is nonzero) to get

$$
|F| = \frac{1}{1^{4-2}}\begin{vmatrix} \begin{vmatrix} 0 & -1 \\ 1 & 0 \end{vmatrix} & \begin{vmatrix} 0 & 2 \\ 1 & -3 \end{vmatrix} & \begin{vmatrix} 0 & -3 \\ 1 & 4 \end{vmatrix} \\ \begin{vmatrix} 1 & 0 \\ -2 & 3 \end{vmatrix} & \begin{vmatrix} 1 & -3 \\ -2 & 0 \end{vmatrix} & \begin{vmatrix} 1 & 4 \\ -2 & -5 \end{vmatrix} \\ \begin{vmatrix} 1 & 0 \\ 3 & -4 \end{vmatrix} & \begin{vmatrix} 1 & -3 \\ 3 & 5 \end{vmatrix} & \begin{vmatrix} 1 & 4 \\ 3 & 0 \end{vmatrix} \end{vmatrix} = \begin{vmatrix} 1 & -2 & 3 \\ 3 & -6 & 3 \\ -4 & 14 & -12 \end{vmatrix}.
$$

Again using the $(1,1)$ element we have

$$
|F| = \frac{1}{1^{3-2}}\begin{vmatrix} \begin{vmatrix} 1 & -2 \\ 3 & -6 \end{vmatrix} & \begin{vmatrix} 1 & 3 \\ 3 & 3 \end{vmatrix} \\ \begin{vmatrix} 1 & -2 \\ -4 & 14 \end{vmatrix} & \begin{vmatrix} 1 & 3 \\ -4 & -12 \end{vmatrix} \end{vmatrix} = \begin{vmatrix} 0 & -6 \\ 6 & 0 \end{vmatrix} = 36\,,
$$

The same as before. These numbers are verified in the `R` code `chap_2_sec_2_2_prob_1.R`.

Section 2.2 Problem 3 (definiteness of some matrices)

Part (a): The leading principle minors for this matrix are

$$
\begin{aligned}
\Delta_1 &= 1 \\
\Delta_2 &= \begin{vmatrix} 1 & 0 \\ 0 & 2 \end{vmatrix} = 2 \\
\Delta_3 &= \begin{vmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 3 \end{vmatrix} = 6\,.
\end{aligned}
$$

Since all leading principal minors are positive this matrix is positive definite. In addition, since this is a diagonal matrix the eigenvalues are the elements of the diagonal. Since all eigenvalues are positive we can conclude that this is a positive definite matrix.

Part (b): Since this is a diagonal matrix it has eigenvalues given by 1, -2 and 0, since there are both positive and negative eigenvalues this matrix is of indeterminate type.

Part (c): Q has eigenvalues 1, 2, and 0 this is a positive semidefinite matrix.

Part (d): We compute the leading principal minors of this matrix to find

$$\Delta_1 = 1 \quad \Delta_2 = 1 - 1 = 0 \,.$$

We cannot tell what the type of the matrix Q is with this test. Lets compute the eigenvalues of Q we have

$$(\lambda - 1)^2 - 1 = 0 \quad \rightarrow \quad \lambda \in \{0, 2\} \,.$$

Thus Q is positive semidefinite, since it has one positive eigenvalue and one *zero* eigenvalue.

Section 2.2 Problem 4 (the pseudoinverses)

Part (a): This matrix H is of rank 1 and thus and both $H^T H$ and $H H^T$ are singular so the pseudoinverse does not exist for this matrix.

Part (b): For this matrix H the system $y = Hx$ would represent an overdetermined system. Because the rank of H is two, the product $H H^T$ which is of dimension 3×3 must be singular. The matrix $H^T H$ which is of dimension 2×2 is nonsingular. Because of this we can compute a left pseudoinverse using Equation 10.

Part (c): This matrix is of size 3×4 and has rank 3. The system $y = Hx$ represents an underdetermined system. The product $H^T H$ is of dimension 4×4 and must be singular. The product $H H^T$ is of dimension 3×3 and will be nonsingular. Because of this using Equation 11 we can compute a right pseudoinverse.

Section 2.2 Problem 5 (a linear transformation)

From the given expression for how $\mathbf{y}$ is computed from $\mathbf{x}$ we have in matrix notation that

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} 3 & 1 & 0 \\ 0 & 1 & 3 \\ 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} .$$

Thus we find (using the differential equation for $\mathbf{x}$) that

$$\begin{aligned}\frac{d}{dt}\begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} &= \begin{bmatrix} 3 & 1 & 0 \\ 0 & 1 & 3 \\ 1 & 0 & 1 \end{bmatrix} \frac{d}{dt}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \\ &= \begin{bmatrix} 3 & 1 & 0 \\ 0 & 1 & 3 \\ 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} -1 & 0 & 0 \\ 0 & -10 & 0 \\ 0 & 0 & -100 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} 3 & 1 & 0 \\ 0 & 1 & 3 \\ 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \\ &= \begin{bmatrix} -3 & -10 & 0 \\ 0 & 10 & -300 \\ -1 & 0 & -100 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} 3 & 1 \\ 3 & 1 \\ 2 & 0 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}.\end{aligned}$$

From the mapping of $\mathbf{x}$ to $\mathbf{y}$ we have

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 3 & 1 & 0 \\ 0 & 1 & 3 \\ 1 & 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} 1/6 & -1/6 & 1/2 \\ 1/2 & 1/2 & -3/2 \\ -1/6 & 1/6 & 1/2 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix}.$$

Thus doing the matrix multiplication we get

$$\frac{d}{dt}\begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \frac{1}{2}\begin{bmatrix} -11 & -9 & 27 \\ 90 & -110 & -270 \\ 33 & -33 & -101 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} + \begin{bmatrix} 3 & 1 \\ 3 & 1 \\ 2 & 0 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}.$$

for the differential equation for $\mathbf{y}$

Section 2.2 Problem 6 (a matrix identity)

Consider the expression Equation 17 with $A_4 = I$, $A_3 = A$, $A_1^{-1} = B^{-1}$ and $A_2 = C$, then the the left-hand-side of that expression is

$$(I - AB^{-1}C)^{-1},$$

while the right-hand-side of that expression becomes

$$I - A(CA - B)^{-1}C,$$

showing the desired expression.

Section 2.3 Problem 1 (numerical integration)

The rectangular or Euler integration method integrates $\dot{x} = f(x)$ by holding the right-hand-side constant at its earliest value of t_{k-1}, when performing the needed quadrature. Thus we have

$$\begin{aligned} x(t_k) &= x(t_{k-1}) + \int_{t_{k-1}}^{t_k} f[x(t), u(t), w(t), p(t), t]dt \qquad (61) \\ &= x(t_{k-1}) + f[x(t_{k-1}), u(t_{k-1}), w(t_{k-1}), p(t_{k-1}), t_{k-1}]\Delta t\,. \qquad (62)\end{aligned}$$

So for the specific differential equation $\dot{x}(t) = \cos(2\pi t)$ we would iterate

$$x(t_k) = x(t_{k-1}) + \cos(2\pi t_{k-1})\Delta t\,.$$

We are told to take $t_k = 0.1k$ for $0 \leq k \leq 10$.

For the trapezoidal rule with a strictly time-dependent right-hand-side in Equation 61 we evaluate

$$\int_{t_{k-1}}^{t_k} f[t]dt = \frac{1}{2}(f[t_{k-1}] + f[t_k])\Delta t\,.$$

For this problem this means that

$$x(t_k) = x(t_{k-1}) + \frac{1}{2}(\cos(2\pi t_{k-1}) + \cos(2\pi t_k))\Delta t\,.$$

For the Runge-Kutta algorithm the steps we compute are given by

$$\begin{aligned}
\Delta x_1 &= f[x(t_{k-1}), u(t_{k-1}), w(t_{k-1}), p(t_{k-1}), t_{k-1}]\Delta t = \cos(2\pi t_{k-1})\Delta t \\
\Delta x_2 &= f[x(t_{k-1}) + \Delta x_1/2, u(t_{k-1/2}), w(t_{k-1/2}), p(t_{k-1/2}), t_{k-1/2}]\Delta t = \cos(2\pi t_{k-1/2})\Delta t \\
\Delta x_3 &= f[x(t_{k-1}) + \Delta x_2/2, u(t_{k-1/2}), w(t_{k-1/2}), p(t_{k-1/2}), t_{k-1/2}]\Delta t = \cos(2\pi t_{k-1/2})\Delta t \\
\Delta x_4 &= f[x(t_{k-1}) + \Delta x_3, u(t_k), w(t_k), p(t_k), t_k]\Delta t = \cos(2\pi t_k)\Delta t \\
x(t_k) &= x(t_{k-1}) + \frac{1}{6}(\Delta x_1 + 2\Delta x_2 + 2\Delta x_3 + \Delta x_4) \\
&= x(t_{k-1}) + \frac{1}{6}(\cos(2\pi t_{k-1}) + 4\cos(2\pi t_{k-1/2}) + \cos(2\pi t_k))\Delta t\,.
\end{aligned}$$

The exact answer for anytime t where $0 \leq t \leq 1$ is

$$x(t) - x(0) = \left.\frac{\sin(2\pi t')}{2\pi}\right|_0^t = \frac{1}{2\pi}\sin(2\pi t)\,.$$

When $t = 1$ we get the answer of 0.

Section 2.3 Problem 2 (local linearization)

When one linearizes the original nonlinear dynamic equation

$$\frac{dx}{dt} = f[x(t), u(t), w(t)]\,,$$

one ends up with a linear system for the perturbation function $\Delta x(t)$ that in general will look like

$$\Delta\dot{x}(t) = F(t)\Delta x(t) + G(t)\Delta u(t) + L(t)\Delta w(t)\,,$$

where F, G, and L are the partial derivatives of the nonlinear function f, with respect to the variables x, u, and w respectively. In this problem we don't have a noise variable w and thus no L term. We now compute the other derivatives and find

$$\begin{aligned} F(t) &= \frac{\partial f}{\partial x} = \begin{bmatrix} a_1 + 2a_2x_1 & 3a_2x_1^2x_2^2 + a_3x_3\cos(x_2) & 0 \\ x_1 + x_3 & 0 & x_1 + x_3 \\ a_4 & 0 & -a_4 \end{bmatrix} \\ G(t) &= \frac{\partial f}{\partial u} = \begin{bmatrix} 2b_1u \\ b_2 \\ 0 \end{bmatrix}. \end{aligned}$$

Thus our linear system in terms of a system of scalar equations is

$$\begin{aligned} \frac{d}{dt}\begin{bmatrix} \Delta x_1 \\ \Delta x_2 \\ \Delta x_3 \end{bmatrix} &= \begin{bmatrix} a_1 + 2a_2x_1 & 3a_2x_1^2x_2^2 + a_3x_3\cos(x_2) & 0 \\ x_1 + x_3 & 0 & x_1 + x_3 \\ a_4 & 0 & -a_4 \end{bmatrix}\begin{bmatrix} \Delta x_1 \\ \Delta x_2 \\ \Delta x_3 \end{bmatrix} \\ &+ \begin{bmatrix} 2b_1u \\ b_2 \\ 0 \end{bmatrix}\Delta u\,. \end{aligned}$$

Since $(x, u) = (0, 0)$ is an equilibrium point, if we happen to linearize about this specific point we will get the system

$$\frac{d}{dt}\begin{bmatrix} \Delta x_1 \\ \Delta x_2 \\ \Delta x_3 \end{bmatrix} = \begin{bmatrix} a_1 & 0 & 0 \\ 0 & 0 & 0 \\ a_4 & 0 & -a_4 \end{bmatrix}\begin{bmatrix} \Delta x_1 \\ \Delta x_2 \\ \Delta x_3 \end{bmatrix} + \begin{bmatrix} 0 \\ b_2 \\ 0 \end{bmatrix}\Delta u\,.$$

Section 2.3 Problem 3 (Van der Pol's equation)

To begin this problem, we write the given scalar differential equation as a system. To do this let $x_1 = x$ and $x_2 = \dot{x}_1 = \dot{x}$, then our system is given by

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= \ddot{x} = -a(1 - x_1^2)x_2 - bx_1 + c\,. \end{aligned}$$

As a matrix system this is

$$\frac{d}{dt}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} x_2 \\ -a(1 - x_1^2)x_2 - bx_1 + c \end{bmatrix}. \tag{63}$$

Part (a): In the above form we can apply numerical integration techniques such as Euler's method, the trapezoidal rule, or a Runge Kutta method.

Part (b): The same as Part (a).

Part (c): The nominal equilibrium condition is given by setting the right-hand-side of the above matrix system equal to zero. This gives $x_2 = 0$ and

$$-bx_1 + c = 0 \quad \text{or} \quad x_1 = \frac{c}{b}\,.$$

In that case we can perform a linearization about the point $(\frac{c}{b}, 0)$ and find

$$\begin{aligned}\begin{bmatrix} x_2 \\ -a(1-x_1^2)x_2 - bx_1 + c \end{bmatrix} &= \begin{bmatrix} 0 \\ 0 \end{bmatrix} + \left.\frac{\partial f}{\partial(x_1, x_2)}\right|_{(x_1,x_2)=(\frac{c}{b},0)} \begin{bmatrix} x_1 - \frac{c}{b} \\ x_2 - 0 \end{bmatrix} + \cdots \\ &\approx \left.\begin{bmatrix} 0 & 1 \\ 2ax_1x_2 - b & -a(1-x_1^2) \end{bmatrix}\right|_{(x_1,x_2)=(\frac{c}{b},0)} \begin{bmatrix} x_1 - \frac{c}{b} \\ x_2 \end{bmatrix} \\ &= \begin{bmatrix} 0 & 1 \\ -b & -a\left(1 - \frac{c^2}{b^2}\right) \end{bmatrix} \begin{bmatrix} x_1 - \frac{c}{b} \\ x_2 \end{bmatrix}.\end{aligned}$$

Where f is the vector function that represents the right-hand-side of Equation 63.

Part (d): When $a = 1$, $b = 1$, and $c = 0$ this coefficient matrix becomes $\begin{bmatrix} 0 & 1 \\ -1 & -1 \end{bmatrix}$. Local linearization about this point would then require integrating the system

$$\frac{d}{dt}\begin{bmatrix} \Delta x_1(t) \\ \Delta x_2(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -1 & -1 \end{bmatrix} \begin{bmatrix} \Delta x_1(t) \\ \Delta x_2(t) \end{bmatrix},$$

to find $\begin{bmatrix} \Delta x_1(t) \\ \Delta x_2(t) \end{bmatrix}$ as a function of t.

When we numerically integrate the equations given by Part (a), Part (b), and Part (d) we obtain the vector functions $x_a(t)$, $x_b(t)$, and $\Delta x_d(t)$. We then expect if the linearization done in Part (c) is valid then

$$x_b(t) \approx x_a(t) + \Delta x_d(t).$$

When $t = 0$ (the initial condition) the above approximation is exact. We do this in the MATLAB script `sect_2_3_prob_3.m`. When that script is run we obtain the plot shown in Figure 2. We see that the linearization $\Delta x_d(t)$ when added to $x_a(t)$ matches $x_b(t)$ very closely.

Section 2.3 Problem 4 (state transition and control effect matrices)

Since the system matrix F for this problem is a constant the state-transition matrix $\Phi(t_2, t_1)$ has the explicit form given by

$$\Phi(t_2, t_1) = e^{F(t_2 - t_1)} = e^{F\Delta t}.$$

We can explicitly compute $e^{F\Delta t}$ if we can find a similarity transformation to diagonalize the matrix F as $F = E\Lambda E^{-1}$ then powers of F are easy to compute

$$F^2 = E\Lambda^2 E^{-1}, \cdots, F^n = E\Lambda^n E^{-1}.$$

Thus using the power series representation of $e^{F\Delta t}$ we can show that

$$e^{F\Delta t} = Ee^{\Lambda\Delta t}E^{-1} = E\begin{bmatrix} e^{\lambda_1\Delta t} & & \\ & \ddots & \\ & & e^{\lambda_n\Delta t} \end{bmatrix} E^{-1}.$$

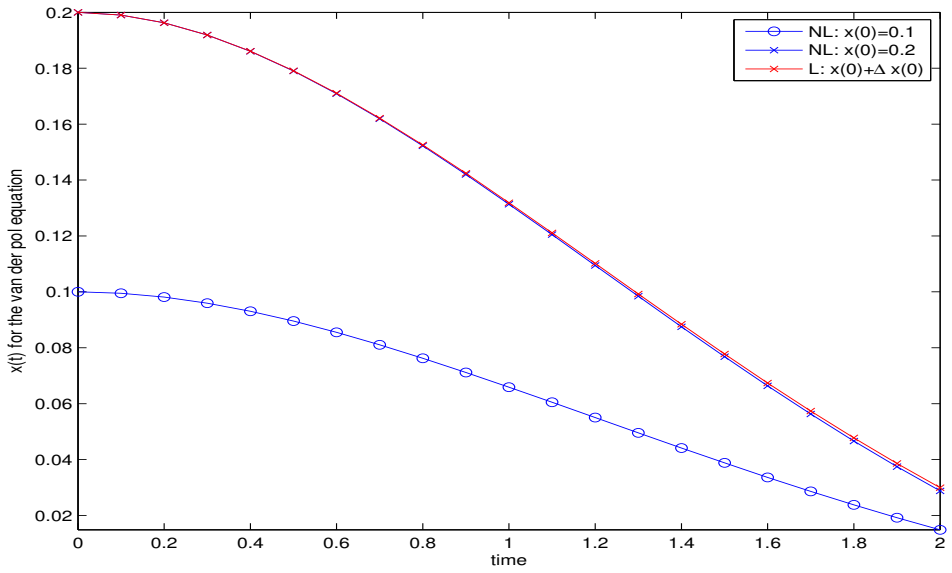

Figure 2: A comparison of the nonlinear integration of the Van Der Pol equation with its linearization.

For this problem because F is upper triangular we know that the eigenvalues are -0.5 and -0.7. To compute the eigenvalues for $\lambda_1 = -0.5$ we consider the null space of the matrix $F - (-0.5)I$ or

$$\begin{bmatrix} 0 & 1 \\ 0 & -0.7 + 0.5 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = 0 \,.$$

Which states that v_1 can be arbitrary while v_2 must be zero. Lets take $v_1 = 1$. For the eigenvectors for $\lambda_2 = -0.7$, we consider the null space of the matrix $F - (-0.7)I$ or

$$\begin{bmatrix} -0.5 + 0.7 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = 0 \,.$$

Which states that $v_2 = -0.2 v_1$ and v_1 arbitrary. Lets take $v_1 = 1$, so that $v_2 = -0.2$. With the eigenvectors specified we can form the matrix of eigenvectors E and its inverse E^{-1} given by

$$E = \begin{bmatrix} 1 & 1 \\ 0 & -0.2 \end{bmatrix} \quad \text{and} \quad E^{-1} = \begin{bmatrix} 1 & 5 \\ 0 & -5 \end{bmatrix} .$$

Then

$$e^{F\Delta t} = E \begin{bmatrix} e^{-0.5\Delta t} & 0 \\ 0 & e^{-0.7\Delta t} \end{bmatrix} E^{-1} = \begin{bmatrix} e^{-0.5\Delta t} & 5(e^{-0.5\Delta t} - e^{-0.7\Delta t}) \\ 0 & e^{-0.7\Delta t} \end{bmatrix} ,$$

when we do the matrix multiplications. This is also the expression for $\Phi(\Delta t)$. To evaluate $\Gamma(\Delta t)$, the control effect matrix, we will use Equation 31, as

$$\Gamma(\Delta t) = \Phi(\Delta t)[I_n - \Phi^{-1}(\Delta t)]F^{-1}G = (\Phi(\Delta t) - I_n)F^{-1}G \,,$$

where $G = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$ and $F^{-1} = \begin{bmatrix} -2 & -20/7 \\ 0 & -10/7 \end{bmatrix}$ to find $F^{-1}G = -\frac{10}{7}\begin{bmatrix} 2 \\ 1 \end{bmatrix}$ and

$$\Gamma(\Delta t) = [\Phi(\Delta t) - I_n]F^{-1}G = -\frac{10}{7}\begin{bmatrix} 7e^{-0.5\Delta t} - 5e^{-0.7\Delta t} - 2 \\ e^{-0.7\Delta t} - 1 \end{bmatrix} .$$

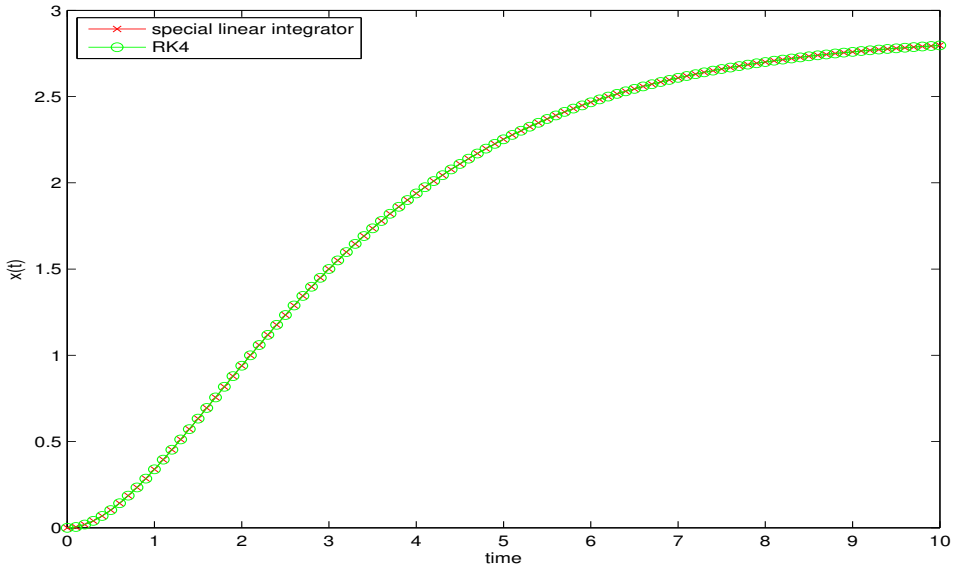

Figure 3: Numerical integration of the differential equation $\frac{d\mathbf{x}}{dt} = \begin{bmatrix} -0.5 & 1 \\ 0 & -0.7 \end{bmatrix} \mathbf{x} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u$, for $u \equiv 1$.

Part (a): With these definitions *one* way we can now obtain the value of $\mathbf{x}$ at the discrete times t_k is by iterating

$$x(t_k) = \Phi(\Delta t)x(t_{k-1}) + \Gamma(\Delta t)u(t_{k-1}) \,,$$

for $\Delta t = \frac{1}{10}$ and for $\frac{10}{\frac{1}{10}} = 100$ time steps.

Part (b): A *second* way to integrate this differential equation is to use Runge-Kutta method, say RK4. This requires us to pick a value for the time step $\Delta t = \frac{1}{10}$, an initial time $t_0 = 0$ and then iterate the following

$$\begin{aligned}
\mathbf{\Delta x}_1 &= \left\{ \begin{bmatrix} -0.5 & 1 \\ 0 & -0.7 \end{bmatrix} \begin{bmatrix} x_1(t_{k-1}) \\ x_2(t_{k-1}) \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \right\} \Delta t \\
\mathbf{\Delta x}_2 &= \left\{ \begin{bmatrix} -0.5 & 1 \\ 0 & -0.7 \end{bmatrix} \begin{bmatrix} x_1(t_{k-1}) + \frac{1}{2}\Delta x_{11} \\ x_2(t_{k-1}) + \frac{1}{2}\Delta x_{12} \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \right\} \Delta t \\
\mathbf{\Delta x}_3 &= \left\{ \begin{bmatrix} -0.5 & 1 \\ 0 & -0.7 \end{bmatrix} \begin{bmatrix} x_1(t_{k-1}) + \frac{1}{2}\Delta x_{21} \\ x_2(t_{k-1}) + \frac{1}{2}\Delta x_{22} \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \right\} \Delta t \\
\mathbf{\Delta x}_4 &= \left\{ \begin{bmatrix} -0.5 & 1 \\ 0 & -0.7 \end{bmatrix} \begin{bmatrix} x_1(t_{k-1}) + \Delta x_{31} \\ x_2(t_{k-1}) + \Delta x_{32} \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \right\} \Delta t \\
\begin{bmatrix} x_1(t_k) \\ x_2(t_k) \end{bmatrix} &= \begin{bmatrix} x_1(t_{k-1}) \\ x_2(t_{k-1}) \end{bmatrix} + \frac{1}{6}(\mathbf{\Delta x}_1 + 2\mathbf{\Delta x}_2 + 2\mathbf{\Delta x}_3 + \mathbf{\Delta x}_4) \,.
\end{aligned}$$

for $k = 1, 2, \cdots, \frac{10}{\Delta t} = 100$ time steps. When we do this in the code `sect_2_3_prob_4.m` we obtain the results given in Figure 3. There we see that the special linear integrator does quite well on this problem.

Section 2.3 Problem 5 (linearized equations of a satellite)

Part (a): For the linearized system

$$\begin{aligned}\Delta\dot{p} &= \frac{M_x}{I_x} \\ \Delta\dot{q} &= \frac{p_0(I_x - I_z)\Delta r}{I_y} + \frac{M_y}{I_y} \\ \Delta\dot{r} &= \frac{p_0(I_y - I_x)\Delta q}{I_z} + \frac{M_z}{I_z}\,.\end{aligned}$$

Then taking as the vector state $\mathbf{x}$ the variables Δp, Δq, and Δr and as the controls M_x, M_y, and M_z we get the following matrix system

$$\frac{d}{dt}\begin{bmatrix}\Delta p \\ \Delta q \\ \Delta r\end{bmatrix} = \begin{bmatrix}0 & 0 & 0 \\ 0 & 0 & \frac{p_0(I_x - I_z)}{I_y} \\ 0 & \frac{p_0(I_y - I_x)}{I_z} & 0\end{bmatrix}\begin{bmatrix}\Delta p \\ \Delta q \\ \Delta r\end{bmatrix} + \begin{bmatrix}1/I_x & 0 & 0 \\ 0 & 1/I_y & 0 \\ 0 & 0 & 1/I_z\end{bmatrix}\begin{bmatrix}M_x \\ M_y \\ M_z\end{bmatrix}.$$

This is a type of dynamical system in the form $\frac{d\mathbf{x}}{dt} = F\mathbf{x} + G\mathbf{u}$, where F and G are independent of time. Because of this, the state translation matrix $\Phi(\Delta t)$ is $e^{F\Delta t}$. We might be able to compute expressions like this this analytically based on the form of the matrix F, but it will be easier to compute everything numerically. One has to be careful in using Matlab for the calculation $e^{F\Delta t}$. If you have a matrix `F` and simply use the Matlab command `exp(F)`, you will be getting the exponential of each element in the matrix rather than the matrix exponential. To get the matrix exponential one needs to use `expm(F)`. The difference is substantial. For the F expressed here the two commands give

```
>> F = [ 0, 0, 0; 0, 0, p_0*(I_x - I_z)/I_y; 0, p_0*(I_y-I_x)/I_z, 0 ];
F =

         0         0         0
         0         0  -13.3333
         0    5.0000         0

>> exp(F) % the elementwise exponential:
ans =

    1.0000    1.0000    1.0000
    1.0000    1.0000    0.0000
    1.0000  148.4132    1.0000

>> expm(F) % the matrix exponential
ans =

    1.0000         0         0
         0   -0.3060   -1.5547
         0    0.5830   -0.3060
```

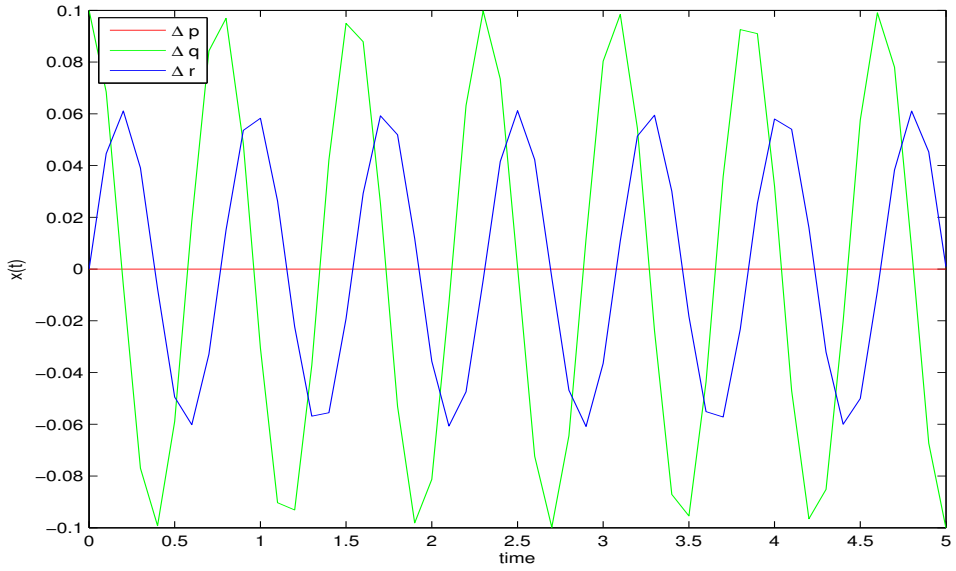

Figure 4: The integrated system for the equations of motion for a spinning orbiting satellite.

We do this in the MATLAB code `sect_2_3_prob_5.m`. When we specify the numerical values for I_x, I_y, I_z and p_0 we obtain

$$\Phi(\Delta t) = \begin{bmatrix} 1.0000 & 0 & 0 \\ 0 & 0.6848 & -1.1900 \\ 0 & 0.4463 & 0.6848 \end{bmatrix} \quad \text{and} \quad G = \begin{bmatrix} 0.0020 & 0 & 0 \\ 0 & 0.0013 & 0 \\ 0 & 0 & 0.0010 \end{bmatrix}.$$

For this system since F is singular so we can't use Equation 31 to compute $\Gamma(\Delta t)$ and we must instead use Equation 35. When we do that we find

$$\Gamma(\Delta t) = \frac{1}{1000} \begin{bmatrix} 0.2000 & 0 & 0 \\ 0 & 0.1190 & -0.0630 \\ 0 & 0.0315 & 0.0893 \end{bmatrix}.$$

Part (b): For this system when we assume no input forcing so that $M_x = M_y = M_z = 0$ we can integrate this system to get the result shown in Figure 4.

Section 2.4 Problem 1 (generating random variables)

Part (c): For this part of the problem we generate 1000 random variables that are themselves created from the sum of 2 or 3 uniform random variables. In the `R` code `sect_4_prob_1.R` we do this and plot the results in Figure 5.

Section 2.4 Problem 2 (statistics of the uniform distribution)

For this problem at first we will consider a uniform (rectangular) distribution between the values of α and β and then restrict this result to the case of interest. The uniform distribution

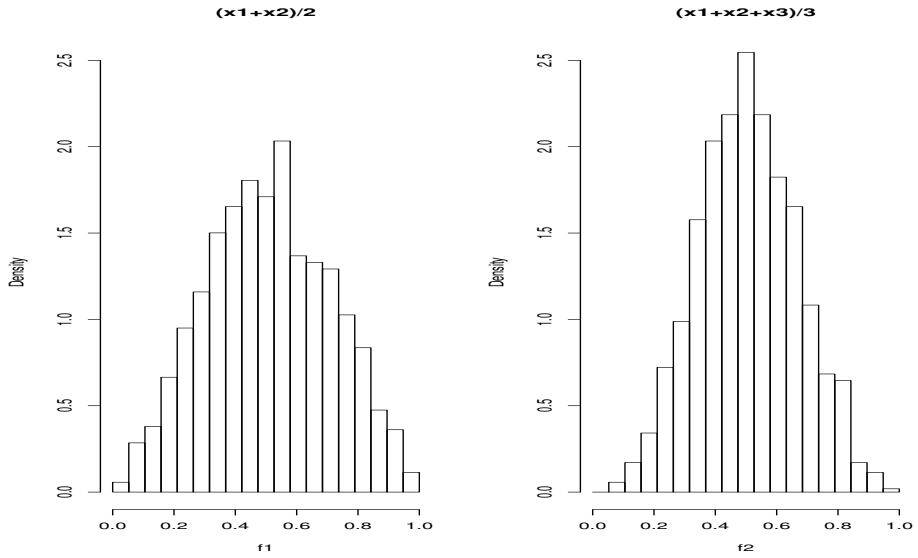

Figure 5: **Left:** The random variable V obtained by adding (and dividing by 2) two uniform random variables. **Right:** The random variable V obtained by adding (and dividing by 3) three uniform random variables. Note that this is more peaked around the population mean (1/2) than the previous histogram.

has a characteristic function that can be computed directly

$$\begin{aligned}\zeta(t) &= E(e^{itX}) = \int_\alpha^\beta e^{itx}\frac{1}{\beta-\alpha}dx \\ &= \frac{1}{\beta-\alpha}\left(\frac{e^{it\beta}-e^{it\alpha}}{it}\right).\end{aligned}$$

We could compute $E(X)$ using the characteristic function $\zeta(t)$ for a uniform random variable. Beginning this calculation we have

$$\begin{aligned}E(X) &= \frac{1}{i}\left.\frac{\partial\zeta(t)}{\partial t}\right|_{t=0} \\ &= \frac{1}{i}\frac{1}{\beta-\alpha}\left[\frac{1}{it}(i\beta e^{it\beta}-i\alpha e^{it\alpha})-\frac{1}{it^2}(e^{it\beta}-e^{it\alpha})\right]\Bigg|_{t=0} \\ &= -\frac{1}{\beta-\alpha}\left[\frac{t(i\beta e^{it\beta}-i\alpha e^{it\alpha})-(e^{it\beta}-e^{it\alpha})}{t^2}\right]\Bigg|_{t=0}.\end{aligned}$$

To evaluate this expression requires the use of L'Hopital's rule, and seems a somewhat complicated route to compute $E(X)$. The evaluation of $E(X^2)$ would probably be even more work when computed from the characteristic function. For this distribution, it is much easier to compute the expectations directly. We have

$$E(X) = \int_\alpha^\beta x\frac{1}{\beta-\alpha}dx = \frac{1}{\beta-\alpha}\left.\frac{x^2}{2}\right|_\alpha^\beta = \frac{1}{2}(\alpha+\beta).$$

In the same way we find $E(X^2)$ to be given by

$$\begin{aligned} E(X^2) &= \int_\alpha^\beta x^2 \frac{1}{\beta-\alpha} dx = \frac{1}{\beta-\alpha}\left(\frac{\beta^3-\alpha^3}{3}\right) \\ &= \frac{(\beta-\alpha)(\beta^2+\alpha\beta+\alpha^2)}{3(\beta-\alpha)} = \frac{1}{3}(\beta^2+\alpha\beta+\alpha^2)\,. \end{aligned}$$

Using these two results we thus have that the variance of a uniform random variable is

$$\begin{aligned} \mathrm{Var}(X) &= E(X^2) - E(X)^2 \\ &= \frac{1}{3}(\beta^2+\alpha\beta+\alpha^2) - \frac{1}{4}(\alpha^2+\beta^2+2\alpha\beta) \\ &= \frac{(\beta-\alpha)^2}{12}\,. \end{aligned}$$

If we specify the above results to the case where $\alpha = -a$ and $\beta = a$ we get

$$\begin{aligned} E(X) &= \frac{1}{2}(-a+a) = 0 \\ E(X^2) &= \frac{1}{3}(a^2-a^2+a^2) = \frac{1}{3}a^2 \\ \mathrm{Var}(X) &= \frac{(a-(-a))^2}{12} = \frac{1}{3}a^2\,. \end{aligned}$$

Section 2.4 Problem 3 (a discrete approximation)

WARNING: I was not sure how to do this problem. Perhaps the problem meant to evaluate z by simulation? If anyone has any ideas please email me.

Section 2.4 Problem 4 (a Rayleigh process)

Part (a): For the given density function

$$f(x) = \frac{x}{r} e^{-\frac{1}{2}\frac{x^2}{r}}\,,$$

for $x \geq 0$ and 0 otherwise, we will verify that this represents a valid probability density function by showing that it integrates to one. We consider

$$\int_{-\infty}^{\infty} f(x)dx = \int_0^\infty f(x)dx = \int_0^\infty \frac{x}{r} e^{-\frac{1}{2}\frac{x^2}{r}} dx\,.$$

To integrate this let $v = \frac{x^2}{r}$, so that $dv = \frac{2x}{r}dx$ or $xdx = \frac{r}{2}dv$. We thus see that the above is equal to

$$\int_0^\infty \frac{1}{r} e^{-v/2} \frac{r}{2} dv = \frac{1}{2}\int_0^\infty e^{-v/2} dv = \frac{1}{2}\left.\frac{e^{-v/2}}{(-1/2)}\right|_0^\infty = \frac{1}{2}(-2)\left[0-1\right] = 1\,,$$

as we were to show.

Part (b): For the Rayleigh density function

$$f(x) = \frac{x}{r} e^{-\frac{1}{2}\frac{x^2}{r}} ,$$

for $x \geq 0$ and 0 otherwise, we will compute the expectation of X and X^2 directly from the definition of the given Rayleigh density function. We have that

$$E(X) = \int_{x=0}^{\infty} \frac{x^2}{r} e^{-\frac{x^2}{2r}} dx \, .$$

To evaluate this integral let $v = \frac{x^2}{2r}$ so that $x = \sqrt{2rv}$ and $dx = \sqrt{\frac{r}{2}} v^{-1/2} dv$ to get

$$\begin{aligned} E(X) &= \frac{1}{r} \int_{v=0}^{\infty} (2rv) e^{-v} \sqrt{\frac{r}{2}} v^{-1/2} dv = \sqrt{2r} \int_0^{\infty} v^{\frac{3}{2}-1} e^{-v} dv \\ &= \sqrt{2r}\Gamma(\frac{3}{2}) = \sqrt{2r} \left(\frac{1}{2}\right) \Gamma(\frac{1}{2}) = \sqrt{\frac{\pi r}{2}} \, . \end{aligned}$$

Next we calculate $E(X^2)$. We find

$$E(X^2) = \frac{1}{r} \int_{v=0}^{\infty} x^3 e^{-\frac{x^2}{2r}} dx \, .$$

Using the same transformations as was used to evaluate $E(X)$ we get

$$\begin{aligned} E(X^2) &= \frac{1}{r} \int_{v=0}^{\infty} 2^{3/2} r^{3/2} v^{3/2} e^{-v} \sqrt{\frac{r}{2}} v^{-1/2} dv \\ &= 2r \int_{v=0}^{\infty} v^{0+1} v^{-v} dv = 2r\Gamma(0) = 2r \, . \end{aligned}$$

Thus the variance of X is given by

$$\mathrm{Var}(X) = E(X^2) - E(X)^2 = 2r - r\frac{\pi}{2} = r\left(2 - \frac{\pi}{2}\right) .$$

Section 2.4 Problem 5 (the p.d.f for y when $y = x^2$ and x is a Gaussian)

We are told that $x \sim \mathcal{N}(1, 1)$ and we let $y = x^2$ and want to determine the probability density function for y. Consider the distribution function for Y or $F_Y(y)$. We have when $y \geq 0$ that

$$\begin{aligned} G_Y(y) &= P\{Y \leq y\} = P\{X^2 \leq y\} = P\{|X| \leq y^{1/2}\} \\ &= F_X(\sqrt{y}) - F_X(-\sqrt{y}) \, , \end{aligned}$$

where $F_X(x)$ is the distribution function for X. Thus the density function $g_Y(y)$ is given by the first derivative of the above expression $G_Y(y)$ with respect to y. Since $F_X' = f_X$ we have

$$\begin{aligned} g_Y(y) &= f_X(\sqrt{y}) \left(\frac{1}{2\sqrt{y}}\right) - f_X(-\sqrt{y}) \left(\frac{-1}{2\sqrt{y}}\right) \\ &= \frac{1}{2\sqrt{y}} (f_X(\sqrt{y}) + f_X(-\sqrt{y})) \, . \end{aligned}$$

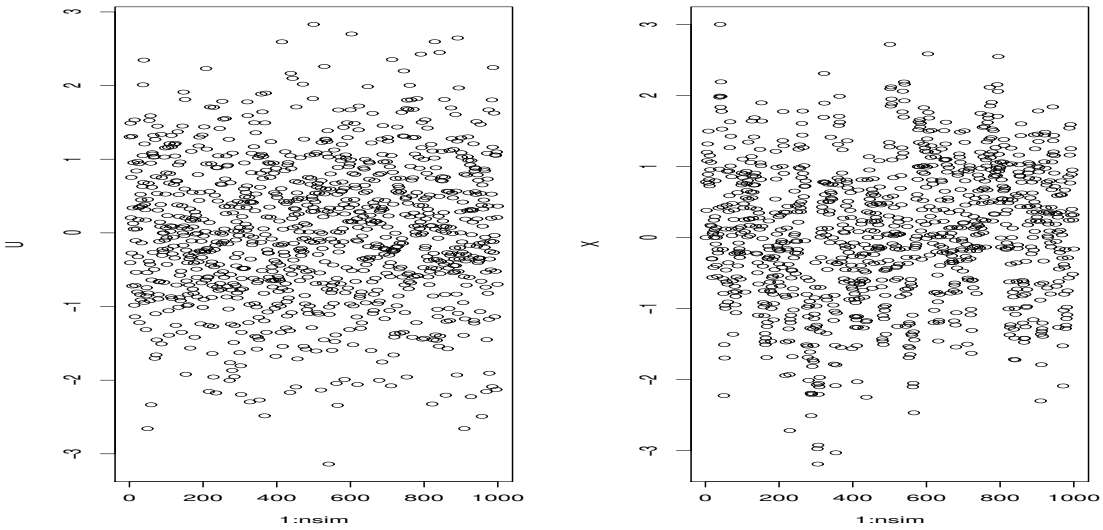

Figure 6: **Left:** The raw time series data for the u_k sequence where each sample, u_k, is drawn from the standard normal distribution. **Right:** The raw time series data for the requested x_k (an AR(1)) sequence. These plots look rather similar.

Since we know the functional form for $f_X(x)$ we can write the above as

$$g_Y(y) = \frac{1}{2\sqrt{2\pi y}} \left(e^{-\frac{(\sqrt{y}-1)^2}{2}} - e^{-\frac{(-\sqrt{y}-1)^2}{2}} \right) .$$

Section 2.4 Problem 6 (an AR(1) process)

This problem is worked in the `R` code `sect_4_prob_6.R`. To begin we plot both the time series of of u_k and x_k and display that in Figure 6. The left plot shows the random inputs u_k and the right plot shows the x_k time series.

Part (a): The continuous autocovariance function, ϕ is defined as

$$\phi[\tilde{x}(t_1), \tilde{x}(t_2)] = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} [x(t_1) - \bar{x}(t_1)][x(t_2) - \bar{x}(t_2)] p(x(t_1), x(t_2)) dx(t_1) dx(t_2) ,$$

which for discretely sampled ergodic sequences can be approximated as

$$\phi_{xx}(k) = E[\tilde{x}_k \tilde{x}_{n+k}] = \lim_{k \to +\infty} \frac{1}{N} \sum_{n=1}^{N} \tilde{x}_n \tilde{x}_{n+k} = \lim_{k \to +\infty} \frac{1}{N} \sum_{n=1}^{N} (x_n - \bar{x})(x_n - \bar{x}_n) .$$

In the same way the discrete cross-covariance is given by

$$\phi_{xy}(k) = E[x_n y_{n+k}] = \lim_{k \to +\infty} \frac{1}{N} \sum_{n=1}^{N} (x_n - \bar{x})(y_{n+k} - \bar{y}) .$$

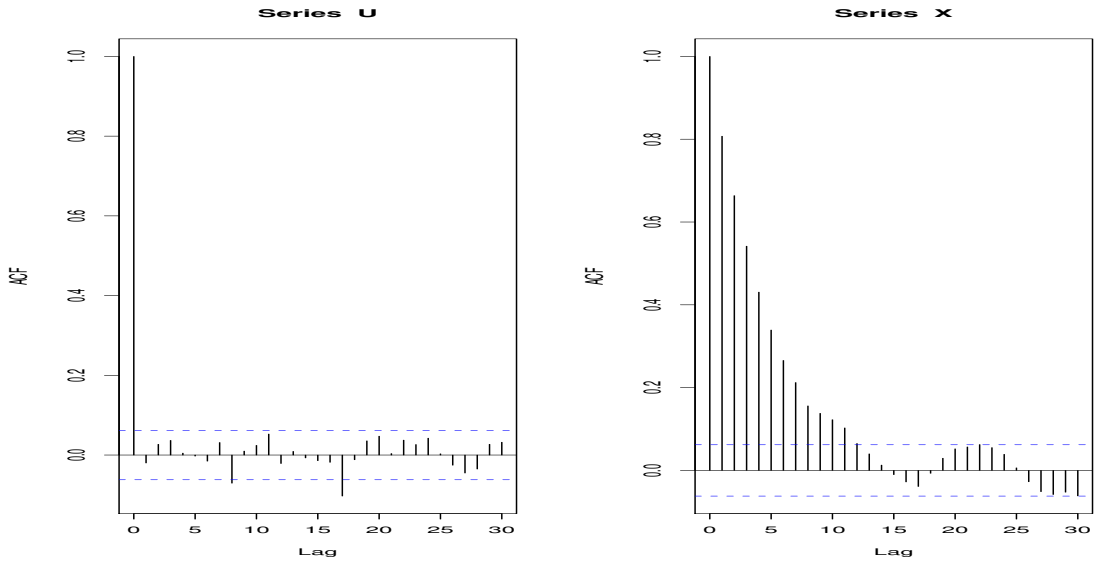

Figure 7: **Left:** The autocorrelation function for the u_k time series. This result shows that there is no memory in the u_k process. **Right:** The autocorrelation function for the x_k time series. This series shows a significant autocorrelation or a relationship between the current term and historical terms. These plots look nothing alike.

In `R` we can compute both of these with the functions `acf`. In Figure 7 we plot the autocorrelation function corresponding to the time series u_k and x_k.

Part (b): In Figure 8 we plot the histogram of the two time series u_k and x_k.

Section 2.5 Problem 1 (the steady-state response)

Part (a): The steady-state response to $\frac{dx}{dt} = Fx + Gu$, when F is invertible and u is specified at u^* is given by

$$x^* = -F^{-1}Gu^* .$$

Since in this part of the problem the given F is *not* invertible, we could look for quasistatic equilibrium solutions as suggested in the book. To find the solutions, the book suggested reordering the state elements to partition F as

$$F = \begin{bmatrix} F_1 & 0 \\ F_2 & 0 \end{bmatrix} .$$

The problem with this is that the state equation for x_2 depends on all three unknowns x_1, x_2, and x_3 thus I don't see how a reordering of the state elements would give a matrix of this form. We can consider the given vector $u^* = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$ and see if we can compute a state vector x such that satisfies

$$Fx = -Gu^* .$$

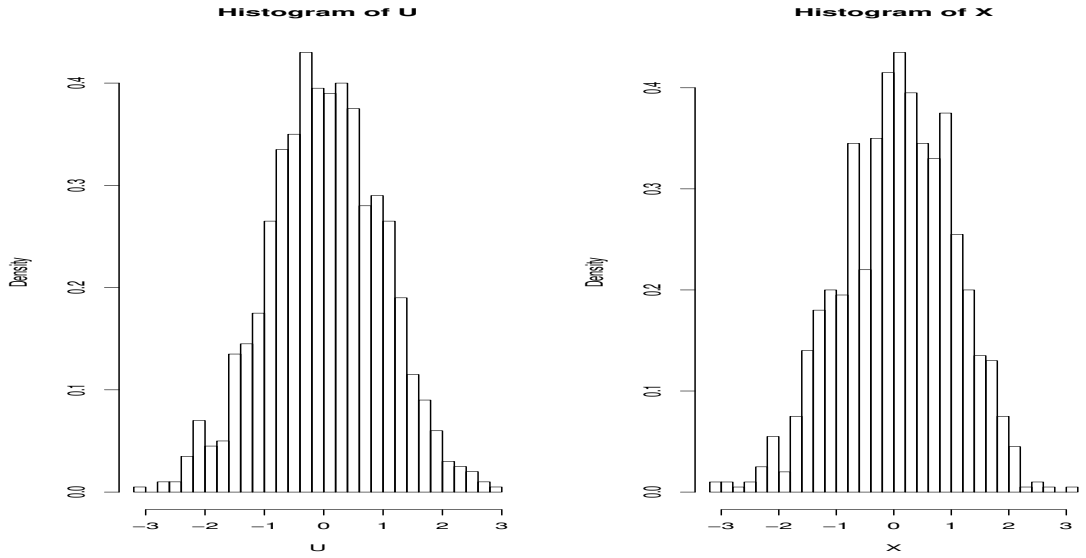

Figure 8: **Left:** A histogram of the noise u_k data. **Right:** A histogram of the state x_k data. Note how similar these two plots look.

For this u^* we find $-Gu^* = \begin{bmatrix} -1 \\ -1 \\ 0 \end{bmatrix}$ and a common way see if any solutions to the above system exist is to first form the augmented matrix

$$\left[\begin{array}{c|c} F & -Gu^* \end{array}\right] = \left[\begin{array}{ccc|c} -1 & 1 & 0 & -1 \\ 1 & -2 & 2 & -1 \\ 0 & 2 & 0 & 0 \end{array}\right],$$

and then perform elementary row operations on this matrix to reduce it to echelon form. MATLAB has a command to do just that called `rref`. When we call this command on the above matrix we get

$$\left[\begin{array}{ccc|c} 1 & 0 & -2 & 0 \\ 0 & 1 & -2 & 0 \\ 0 & 0 & 0 & 1 \end{array}\right].$$

The fact that the last row results in the contradiction $0 = 1$ we conclude that the expression $-Gu^*$ is not in the span of the column space and we *cannot* find a steady-state solution to this problem.

Part (b): If we perform the same steps as in Part (a) above we find we again cannot find a steady-state solution. It is more interesting if we keep the same value for $u^* = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$ but take the second column of G to be in the span of the column space of F. For example we could take

$$G = \begin{bmatrix} 1 & 0 \\ 1 & 1 \\ 0 & -2 \end{bmatrix},$$

where the second column of G is the sum of all of the columns in F. Then in that case forming the augmented matrix and then reducing it to echelon form gives the matrix

$$\left[\begin{array}{ccc|c} 1 & 0 & -2 & 1 \\ 0 & 1 & -2 & 1 \\ 0 & 0 & 0 & 0 \end{array}\right] .$$

This last matrix has an identity for the last row. This means that there is a non-trivial relationship between x_1, x_2, and x_3 in steady-state. This relationship is

$$\begin{aligned} x_1 &= 1 + 2x_3 \\ x_2 &= 1 + 2x_3 \,. \end{aligned}$$

Part (c): If the $(3,3)$ element is changed from -4 to 0 we get the coefficient matrix of

$$\left[\begin{array}{ccc} -1 & 1 & 0 \\ 1 & -2 & 2 \\ 0 & 2 & 0 \end{array}\right] .$$

This is a matrix of rank 3 and is invertible. Thus for the given u^* we find a constant steady-state response given by

$$\frac{1}{4}\left[\begin{array}{c} 2 \\ -2 \\ -7 \end{array}\right] .$$

This problem is worked in the MATLAB file `sect_5_prob_1.m`.

Section 2.5 Problem 3 (the Laplace transform description)

Part (a): For the system $\dot{x}(t) = Fx(t) + Gu(t)$ we have a Laplace transform given by

$$sx(s) = Fx(s) + Gu(s) + x(0) \,,$$

or in this case

$$s\left[\begin{array}{c} x_1(s) \\ x_2(s) \\ x_3(s) \end{array}\right] = \left[\begin{array}{ccc} -1 & 1 & 0 \\ 1 & -2 & 2 \\ 0 & 2 & -4 \end{array}\right]\left[\begin{array}{c} x_1(s) \\ x_2(s) \\ x_3(s) \end{array}\right] + \left[\begin{array}{cc} 1 & 0 \\ 1 & 1 \\ 0 & 1 \end{array}\right]\left[\begin{array}{c} u_1(s) \\ u_2(s) \end{array}\right] + \left[\begin{array}{c} x_1(0) \\ x_2(0) \\ x_3(0) \end{array}\right] .$$

Part (b): Let $u_1 = u_2 = 0$ and we get

$$\left[\begin{array}{ccc} s+1 & -1 & 0 \\ -1 & s+2 & -2 \\ 0 & -2 & s+4 \end{array}\right]\left[\begin{array}{c} x_1(s) \\ x_2(s) \\ x_3(s) \end{array}\right] = \left[\begin{array}{c} x_1(0) \\ x_2(0) \\ x_3(0) \end{array}\right] .$$

The coefficient matrix in the above expression is $sI - F$. We can use the adjoint expression for the inverse $(sI - F)^{-1} = \frac{\text{Adj}(sI-F)}{|sI-F|}$. We need to calculate $|sI - F|$ which in this case becomes

$$\begin{aligned} |sI - F| &= (s+1)[(s+2)(s+4) - 4] + 1[-(s+4)] \\ &= s(s^2 + 7s - 9) \,, \end{aligned} \tag{64}$$

when we simplify. Next we need to compute the adjoint matrix of $sI - F$. We find the matrix of cofactors (defined as C) is given by

$$C = \begin{bmatrix} +[(s+2)(s+4)-4] & -[-s-4] & +[2] \\ -[-s+4] & [(s+1)(s+4)] & -[-2(s+1)] \\ +[2] & -[-2(s+1)] & +[(s+1)(s+2)-1] \end{bmatrix}$$
$$= \begin{bmatrix} s^2+6s+4 & s+4 & 2 \\ s+4 & s^2+5s+4 & 2s+2 \\ 2 & 2s+2 & s^2+3s+1 \end{bmatrix}.$$

Thus we have that $(sI - F)^{-1}$ is given by

$$(sI-F)^{-1} = \frac{C^T}{|sI-F|} = \frac{1}{s(s^2+7s-9)} \begin{bmatrix} s^2+6s+4 & s+4 & 2 \\ s+4 & s^2+5s+4 & 2s+2 \\ 2 & 2s+2 & s^2+3s+1 \end{bmatrix}. \quad (65)$$

We can multiply this by $\begin{bmatrix} x_1(0) \\ x_2(0) \\ x_3(0) \end{bmatrix}$ to get $x(s)$ as a function of the initial conditions assuming no control inputs.

Part (c): If $x_1(0) = x_2(0) = x_3(0) = 0$ then we get

$$\begin{bmatrix} x_1(s) \\ x_2(s) \\ x_3(s) \end{bmatrix} = \frac{1}{s(s^2+7s-9)} \begin{bmatrix} s^2+6s+4 & s+4 & 2 \\ s+4 & s^2+5s+4 & 2s+2 \\ 2 & 2s+2 & s^2+3s+1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 1 & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u_1(s) \\ u_2(s) \end{bmatrix}.$$

We would need to multiply the above matrices and use partial fractions to solve for $x_i(s)$ for $i = 1, 2, 3$.

Part (d): The roots of the characteristic equation are given by solving Equation 64 set equal to zero 0 and are given by $s = 0$ and $s = \frac{1}{2}(-7 \pm \sqrt{13}) \approx \{-5.3, -1.6\}$.

Part (e): If we assume the initial conditions are all one i.e. $x_1(0) = x_2(0) = x_3(0) = 1$ and take the control vector equal to zero then when we consider Equation 65 we have

$$\begin{bmatrix} x_1(s) \\ x_2(s) \\ x_3(s) \end{bmatrix} = \frac{1}{s(s^2+7s-9)} \begin{bmatrix} s^2+7s+9 \\ s^2+7s+10 \\ s^2+5s+5 \end{bmatrix}.$$

We would then need to find partial fraction expansions of each component and then take the inverse Laplace transforms to get $x_i(t)$ for $i = 1, 2, 3$.

Section 2.5 Problem 4 (completely controllable)

To determine controllability we need to compute the controllability matrix given by Equation 50 for each of the specified set of matrices. If this matrix is of rank n then our system is completely controllable. Since $n = 2$ the above expression is equivalent to $\mathbb{C} = \begin{bmatrix} G & FG \end{bmatrix}$.

Part (a): We find $\mathbb{C}$ is rank 1 and this system is not completely controllable.

Part (b): We find $\mathbb{C}$ is rank 1 and this system is not completely controllable.

Part (c): We find $\mathbb{C}$ is rank 2 and this system is completely controllable.

Part (d): We find $\mathbb{C}$ is rank 1 and this system is not completely controllable.

Part (e): We find $\mathbb{C}$ is rank 1 and this system is not completely controllable.

Part (f): We find $\mathbb{C}$ is rank 2 and this system is completely controllable.

This problem is worked in the MATLAB script `sect_5_prob_4.m`.

Section 2.5 Problem 5 (observability)

The observability condition can be obtained via duality by applying controllability to the dual system

$$\dot{x}(t) = -F^T x(t) + H^T u(t) \, .$$

This results in considering the observability matrix given by Equation 54. Again since $n = 2$ this becomes $\mathbb{O} = \begin{bmatrix} H^T & F^T H^T \end{bmatrix}$.

Part (a): We find $\mathbb{O}$ is rank 2 and this system is observable.

Part (b): We find $\mathbb{O}$ is rank 1 and this system is not observable.

Part (c): We find $\mathbb{O}$ is rank 1 and this system is not observable.

Part (d): We find $\mathbb{O}$ is rank 1 and this system is not observable.

Part (e): We find $\mathbb{O}$ is rank 1 and this system is not observable.

Part (f): We find $\mathbb{O}$ is rank 2 and this system is observable.

This problem is worked in the MATLAB script `sect_5_prob_5.m`.

Section 2.5 Problem 6 (nilpotency)

We will consider a general matrix $M = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$, form the desired product M^2 and consider what values of a, b, c, and d we could take for to make this product equal zero. From the form for M^2 we could take $a = 0$, $d = 0$, and $b = 0$ to get the matrix $M = \begin{bmatrix} 0 & 0 \\ c & 0 \end{bmatrix}$.

Since the degree of a nilpotent $n \times n$ matrix is always less than or equal to n there are no 2×2 matrices with nilpotent degree higher than 2.

Section 2.6 Problem 1 (an example control systems)

Recall that for this problem α is the angle of attack, q is the pitching rate $\approx \dot{\alpha}$, δE is the elevator angle, and δF is the flap angle.

Part (a): For this part we want to evaluate the root locus for the pitch-rate q relative to the elevator angle δE, i.e. we need the transfer function for $\frac{q(s)}{\delta E(s)}$. In the same way as derived on Page we have

$$\frac{q(s)}{\delta E(s)} = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} s - M_q & -M_\alpha \\ -1 & s + \frac{L_\alpha}{V} \end{bmatrix}^{-1} \begin{bmatrix} M_{\delta E} \\ -\frac{L_{\delta E}}{V} \end{bmatrix} = \frac{\left(s + \frac{L_\alpha}{V}\right) M_{\delta E} - M_\alpha \left(\frac{L_{\delta E}}{V}\right)}{s^2 + 2\xi\omega_n s + \omega_n^2} .$$

As discussed in that section of the book we typically have $M_{(\cdot)} < 0$, so I'll negate the numbers M_α and $M_{\delta E}$ given in the book which are positive so that they satisfy $M_{(\cdot)} < 0$. Now for this transfer function if we consider the *closed-loop* roots of negative feedback we want to plot the roots of the equation

$$1 + k\left(\frac{q(s)}{\delta E(s)}\right) = 0 ,$$

as a function of positive and negative k. Recall k is denoted as the *loop gain*. This is equivalent to finding the roots s to

$$\delta E(s) + kq(s) = 0 ,$$

for various $-\infty < k < +\infty$ or the roots of

$$s^2 + 2\xi\omega_n s + \omega_n^2 + k\left(s + \frac{L_\alpha}{V}\right) M_{\delta E} - kM_\alpha \left(\frac{L_{\delta E}}{V}\right) = 0 ,$$

or grouping terms by powers of s this is

$$s^2 + (2\xi\omega_n + kM_{\delta E})s + \left(\omega_n^2 + k\left(\frac{L_\alpha}{V}M_{\delta E} - \frac{L_{\delta E}}{V}M_\alpha\right)\right) = 0 .$$

This is a quadratic equation and so we could solve for s explicitly using the quadratic formula. Rather than do that I choose to use the MATLAB `roots` command which is a more general technique and would work for more complicated transfer functions. In the MATLAB script `sect_6_prob_1_part_a.m` we do just that for $-100 \leq k \leq +100$. When that script is run it produces the plot shown in Figure 9 (left). Since there are two roots we plot one in red and the other in blue. We plot each pole when $k = 0$ as a large black dot. For negative values of k we plot the roots as a dot, while for positive values of k we plot the roots as a cross.

Part (b): For this part we want to consider the transfer function for the angle-of-attack $\frac{\alpha(s)}{\delta E(s)}$ which we have computed on Page and the corresponding close-loop root locus. That is we desire the solutions for s of

$$1 + k\left[-\frac{\left(\frac{L_{\delta E}}{V}\right)\left(s - \left(M_q + \frac{M_{\delta E}}{L_{\delta E}/V}\right)\right)}{s^2 + 2\xi\omega_n s + \omega_n^2}\right] = 0 ,$$

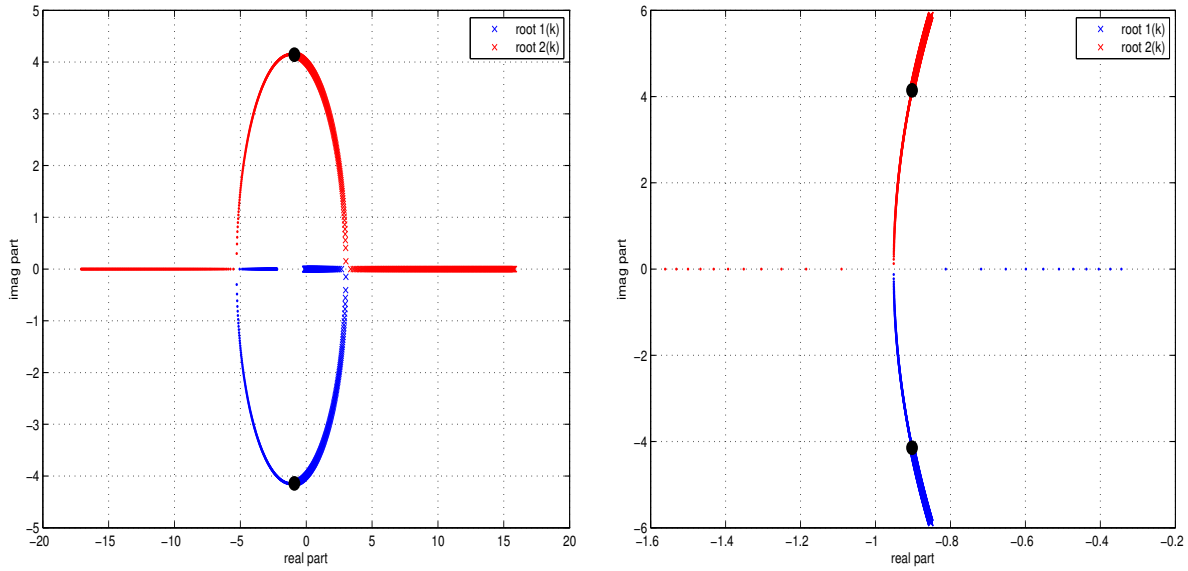

Figure 9: **Left:** The root loci for the transfer function $\frac{q(s)}{\delta E(s)}$. **Right:** The root loci for the transfer function $\frac{\alpha(s)}{\delta E(s)}$.

for various value of the loop gain k. This is equivalent to solving for s in (when we group everything by powers of s)

$$s^2 + \left[2\xi\omega_n - k\frac{L_{\delta E}}{V}\right] s + \left[\omega_n^2 + K\left(\frac{L_{\delta E}}{V}\right)\left(M_q - \frac{M_{\delta E}}{L_{\delta E}/V}\right)\right] = 0\,.$$

In the MATLAB script `sect_6_prob_1_part_b.m` plots of these roots are shown in Figure 9 (right).

Part (c-d): The *open-loop* frequency response is defined as

$$Y(j\omega) = H(j\omega I - F)^{-1}G\,,$$

evaluated for $\omega \in (0, +\infty)$. If we assume that $H = I$ i.e. both components are observable, then the above expression has been evaluated in Part (a) and (b) of this problem. We will consider only pitch rate since the other calculation is similar. We find

$$\frac{q(j\omega)}{\delta E(j\omega)} = \frac{\left(j\omega + \frac{L_\alpha}{V}\right) M_{\delta E} - M_\alpha \left(\frac{L_{\delta E}}{V}\right)}{-\omega^2 + 2j\xi\omega_n\omega + \omega_n^2}\,.$$

We will plot this using a *Bode* plot where we consider the polar representation of $Y(j\omega)$ as $A(\omega)e^{j\phi(\omega)}$ and then plot $20\log(A(\omega))$ and $\phi(\omega)$ in degrees. In the MATLAB script `sect_6_prob_1_part_c_N_d.m` we do this. When we do we get the plots shown in Figure 10 (left) and (right).

Section 2.6 Problem 2 (control dynamics)

Warning: I'm not sure that what I have answered is what was intended for this problem.

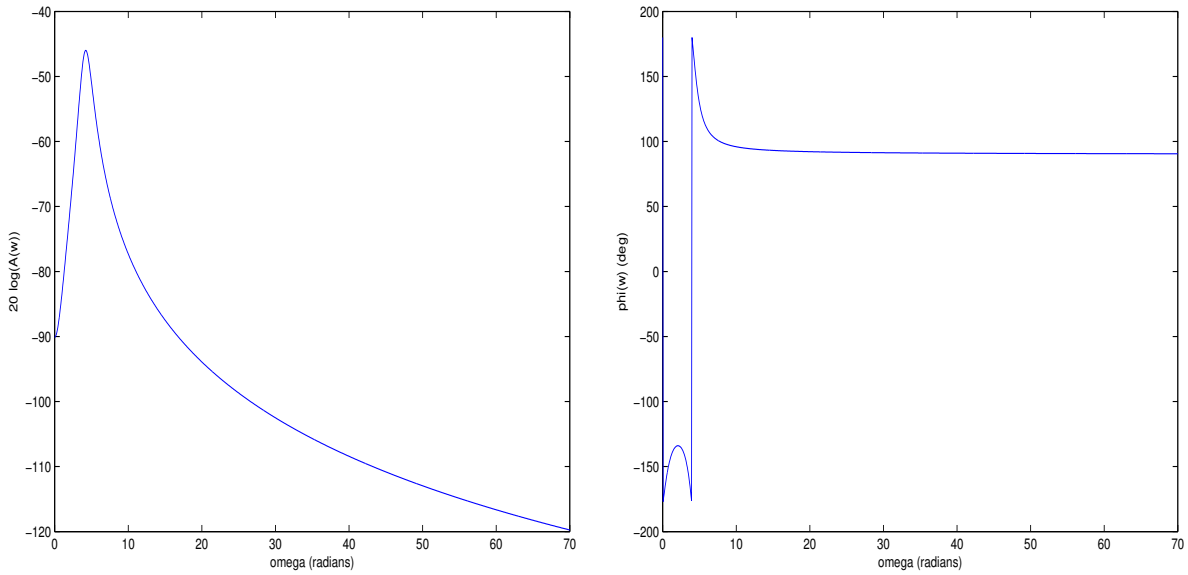

Figure 10: For the open-loop frequency response for $\frac{q(j\omega)}{\delta E(j\omega)} = A(\omega)e^{j\phi(\omega)}$. **Left:** A plot of $20\log(A(\omega))$ as a function of ω. **Right:** A plot of $\phi(\omega)$ (in degrees) as a function of ω.

If anyone has a better ideas please contact me.

We are told that δE has a differential equation given by

$$\delta\dot{E} = -\frac{1}{\tau}\delta E + \frac{1}{\tau}\delta E_C\,,$$

where now δE_c is the control to the elevation that is applied. Then our system has to be extended to include this equation for δE as

$$s\begin{bmatrix} q(s) \\ \alpha(s) \\ \delta E(s) \end{bmatrix} = \begin{bmatrix} M_q & M_\alpha & 0 \\ 1 & \frac{L_\alpha}{V} & 0 \\ 0 & 0 & -1/\tau \end{bmatrix}\begin{bmatrix} q(s) \\ \alpha(s) \\ \delta E(s) \end{bmatrix} + \begin{bmatrix} M_{\delta E} & M_{\delta F} & 0 \\ -\frac{L_{\delta E}}{V} & -\frac{L_{\delta F}}{V} & 0 \\ 0 & 0 & 1/\tau \end{bmatrix}\begin{bmatrix} \delta E(s) \\ \delta F(s) \\ \delta E_c(s) \end{bmatrix}.$$

Thus we would need to solve for $\begin{bmatrix} q(s) \\ \alpha(s) \\ \delta E(s) \end{bmatrix}$ and then consider transfer functions defined by $\frac{q(s)}{\delta E_c(s)}$ and $\frac{\alpha(s)}{\delta E_c(s)}$ as in the previous problem.

Section 2.6 Problem 4 (more control systems)

For this problem we have that x_1 and x_2 are the angular position and rate of the *rigid body motion*, while x_3 and x_4 are the angular position and rate of the *bending motion*. The variable y is the total net position and is the sum of x_1 and x_3. Notice that the system

specified is nonlinear but when we take $a_{21} = a_{23} = a_{41} = 0$ we have the matrix system

$$\frac{d}{dt}\begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \\ x_4(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -\omega_n^2 & -2\zeta\omega_n \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \\ x_4(t) \end{bmatrix} + \begin{bmatrix} 0 \\ b_2 \\ 0 \\ b_4 \end{bmatrix} u(t)\,,$$

which is a linear system. If we take the Laplace transform of the above system and drop the initial conditions on $x_i(0)$ for $i = 1, 2, 3, 4$ we have

$$s\begin{bmatrix} x_1(s) \\ x_2(s) \\ x_3(s) \\ x_4(s) \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -\omega_n^2 & -2\zeta\omega_n \end{bmatrix} \begin{bmatrix} x_1(s) \\ x_2(s) \\ x_3(s) \\ x_4(s) \end{bmatrix} + \begin{bmatrix} 0 \\ b_2 \\ 0 \\ b_4 \end{bmatrix} u(s)\,.$$

Or solving for the vector of $x_i(s)$ we have

$$\begin{bmatrix} s & -1 & 0 & 0 \\ 0 & s & 0 & 0 \\ 0 & 0 & s & -1 \\ 0 & 0 & \omega_n^2 & s + 2\zeta\omega_n \end{bmatrix} \begin{bmatrix} x_1(s) \\ x_2(s) \\ x_3(s) \\ x_4(s) \end{bmatrix} = \begin{bmatrix} 0 \\ b_2 \\ 0 \\ b_4 \end{bmatrix} u(s)\,.$$

We need to take the inverse of the matrix above. We could use the fact that

$$(sI_n - F)^{-1} = \frac{\text{Adj}(sI_n - F)}{|sI_n - F|}\,,$$

or we could just compute the inverse directly. Using Mathematica in `sect_6_prob_4_inverse.nb` we find

$$\begin{bmatrix} s & -1 & 0 & 0 \\ 0 & s & 0 & 0 \\ 0 & 0 & s & -1 \\ 0 & 0 & \omega_n^2 & s + 2\zeta\omega_n \end{bmatrix}^{-1} = \begin{bmatrix} \frac{1}{s} & \frac{1}{s^2} & 0 & 0 \\ 0 & \frac{1}{s} & 0 & 0 \\ 0 & 0 & \frac{s+2\omega_n\zeta}{s^2+2\zeta\omega_n+\omega_n^2} & \frac{1}{s^2+2\zeta\omega_n+\omega_n^2} \\ 0 & 0 & -\frac{\omega_n^2}{s^2+2\zeta\omega_n+\omega_n^2} & \frac{s}{s^2+2\zeta\omega_n+\omega_n^2} \end{bmatrix}.$$

Thus we have

$$\begin{bmatrix} x_1(s) \\ x_2(s) \\ x_3(s) \\ x_4(s) \end{bmatrix} = \begin{bmatrix} \frac{1}{s} & \frac{1}{s^2} & 0 & 0 \\ 0 & \frac{1}{s} & 0 & 0 \\ 0 & 0 & \frac{s+2\omega_n\zeta}{s^2+2\zeta\omega_n+\omega_n^2} & \frac{1}{s^2+2\zeta\omega_n+\omega_n^2} \\ 0 & 0 & -\frac{\omega_n^2}{s^2+2\zeta\omega_n+\omega_n^2} & \frac{s}{s^2+2\zeta\omega_n+\omega_n^2} \end{bmatrix} \begin{bmatrix} 0 \\ b_2 \\ 0 \\ b_4 \end{bmatrix} u(s) = \begin{bmatrix} \frac{b_2}{s^2} \\ \frac{b_2}{s} \\ \frac{b_4}{s^2+2\zeta\omega_n+\omega_n^2} \\ \frac{b_4 s}{s^2+2\zeta\omega_n+\omega_n^2} \end{bmatrix} u(s)\,.$$

Part (a): From the above we now have transfer functions for $\frac{x_i(s)}{u(s)}$ for $i = 1, 2, 3, 4$, and to compute the root loci we want to consider the roots of

$$1 + k\left(\frac{x_i(s)}{u(s)}\right) = 0\,,$$

for negative value of k. This would be done in the same way as for Problem 1 from this section and presented on Page 50.

Part (b): To plot the open-loop frequency response for $\frac{y(s)}{u(s)}$, since $y(t) = x_1(t) + x_3(t)$ the Laplace transform of this expression gives $y(s) = x_1(s) + x_3(s)$. From this expression the transfer function $\frac{y(s)}{u(s)}$ is then given by

$$\frac{y(s)}{u(s)} = \frac{x_1(s)}{u(s)} + \frac{x_3(s)}{u(s)} .$$

The two terms in the right-hand-side of this last expression were computed in the previous part of this problem.

Part (c): The Nyquist plot of $\frac{y(s)}{u(s)}$ simply evaluates $\frac{y(s)}{u(s)}$ at $s = j\omega$ and then plots these numbers in the polar plane as ω varies.

Optimal Trajectories and Neighboring Optimal Solutions

Notes on the text

The Cart on a Track Example: Part 1

Part (a): For this example we start with the simple constraint

$$J = q(x_{1_f} - 100)^2 + r\int_0^{t_f} u^2 dt\,.$$

Now $u^2 = k^2$ is a constant and $t_f = 10$ so we can evaluate the above some to get

$$J = q(x_{1_f} - 100)^2 + rk^2 t_f\,.$$

Since $x_1(t) = \frac{1}{2}kt^2$ we have that $x_{1_f} = x_1(t_f) = 50k$ then J becomes

$$J = q(50k - 100)^2 + 10rk^2\,.$$

From this we have that

$$\frac{\partial J}{\partial k} = 2q(50k - 100)(50) + 20rk = (5000q + 20r)k - 10000q\,.$$

Setting $\frac{\partial J}{\partial k} = 0$ we get the optimal k given by

$$k_{\text{opt}} = \frac{1000q}{500q + 2r} = \frac{1000\left(\frac{q}{r}\right)}{500\left(\frac{q}{r}\right) + 2}\,.$$

As $\frac{q}{r} \to \infty$ we get $k_{\text{opt}} \to \frac{1000}{500} = 2$, while when $\frac{q}{r} \to 0$ we get $k_{\text{opt}} \to 0$.

Part (b): Consider the model of control given by $u = k_1 + k_2 t$ where k_1 and k_2 need to be specified. Our system dynamics are still

$$\dot{x}_1 = x_2 \tag{66}$$

$$\dot{x}_2 = k_1 + k_2 t\,. \tag{67}$$

Then integrating the second equation gives $x_2(t) = k_1 t + \frac{1}{2}k_2 t^2$ when we impose the initial condition that $x_2(0) = 0$. Then $x_1(t)$ is given by integrating $x_2(t)$ to get $x_1(t) = \frac{1}{2}k_1 t^2 + \frac{1}{6}k_2 t^3$ again using $x_1(0) = 0$. To make $x_1(t_f) = 100$ when $t_f = 10$ requires that

$$100 = 50k_1 + 166.7k_2 \quad \text{or} \quad k_1 = 2 - \frac{166.7}{50}k_2\,.$$

Evaluating our constraint J under this control gives

$$\begin{aligned} J = J(k_1, k_2) &= q(x_{1_f} - 100)^2 + r\int_0^{t_f} (k_1 + k_2 t)^2 dt \\ &= q(x_{1_f} - 100)^2 + r\int_0^{t_f} (k_1^2 + 2k_1 k_2 t + k_2^2 t^2) dt \\ &= q(x_{1_f} - 100)^2 + r\left(k_1^2 t_f + k_1 k_2 t_f^2 + \frac{1}{3}k_2^2 t_f^3\right)\,. \end{aligned}$$

Since $t_f = 10$ and $x_{1_f} = 50k_1 + 166.7k_2$ this becomes

$$J = q(50k_1 + 166.7k_2 - 100)^2 + r\left(10k_1^2 + 100k_1k_2 + 333.3k_2^2\right) .$$

Now computing $\frac{\partial J}{\partial k_1}$ and $\frac{\partial J}{\partial k_2}$ we have

$$\begin{aligned}\frac{\partial J}{\partial k_1} &= 2q(50k_1 + 166.7k_2 - 100)(50) + r(20k_1 + 100k_2) \\ \frac{\partial J}{\partial k_2} &= 2q(50k_1 + 166.7k_2 - 100)(166.7) + r(100k_1 + 666.6k_2) .\end{aligned}$$

Setting these two derivatives equal to zero, dividing by 10, and forming a linear system we find that k_1 and k_2 must satisfy

$$\begin{aligned}(500q + 2r)k_1 + (1666.7q + 10r)k_2 &= 1000q \\ (1666.7q + 10r)k_1 + (5555.6q + 66.6r)k_2 &= 3333.3q .\end{aligned}$$

This is a linear system such that that given values for q and r we can solve to get the optimal solutions for k_1 and k_2.

If we *also* want to impose the "constraint" that the velocity at $x_{f_1} = 100$ is zero we would like $x_2(t_f) = 0$. Under the constraint $u(t) = k_1 + k_2 t$ and what that makes the functional form for $x_2(t)$, we could impose this as a "hard" constraint. This means that

$$x_2(t_f) = x_2(10) = 10k_1 + 50k_2 = 0 . \tag{68}$$

If we impose this constraint as specified then since the velocity $x_2(t)$ looks like

$$x_2(t) = k_1 t + \frac{k_2}{2} t^2 ,$$

we see that the acceleration $a(t)$ is given by

$$a(t) = \dot{x}_2(t) = k_1 + k_2 t ,$$

Evaluating $a(t)$ at $t = 5$ seconds (the midpoint of the trajectory), to get $a(5) = k_1 + 5k_2 = \frac{1}{10}(10k_1 + 50k_2) = 0$ by Equation 68 the requirement that $v(t_f) = v(10) = 0$. Thus we have shown that the acceleration at the midpoint of the trajectory must vanish. We could also enforce this constraint "softly" by including it as a cost our new cost function J could look like

$$J = q_1(x_{1_f} - x_{1_D})^2 + q_2 x_{2_f}^2 + r\int_0^{t_f} u^2 dt .$$

Part (d): We now consider the control $u(t) = k_1 \cos(\omega_1 t) + k_2 \sin(\omega_2 t)$ with $\omega_1 = \frac{\pi}{10}$ and $\omega_2 = \frac{\pi}{5}$. Since the differential equation for $x_2(t)$ is $\dot{x}_2(t) = u(t)$ and we know $u(t)$ we can integrate this equation to find an expression for $x_2(t)$. We have

$$x_2(t) = \frac{k_1}{\omega_1} \sin(\omega_1 t) - \frac{k_2}{\omega_2} \cos(\omega_2 t) + C .$$

Since $x_2(0) = 0 = 0 - \frac{k_2}{\omega_2} + C$ we see that $C = \frac{k_1}{\omega_1}$ and thus

$$x_2(t) = \frac{k_1}{\omega_1}\sin(\omega_1 t) + \frac{k_2}{\omega_2}(1 - \cos(\omega_2 t))\,.$$

Since the differential equation for $x_1(t)$ is $\dot{x}_1(t) = x_2(t)$ and we know $x_2(t)$ we can integrate this equation to find an expression for $x_1(t)$. We have

$$x_1(t) = -\frac{k_1}{\omega_1}\cos(\omega_1 t) - \frac{k_2}{\omega_2^2}\sin(\omega_2 t) + \frac{k_2}{\omega_2}t + C\,.$$

Since $x_1(0) = 0 = -\frac{k_1}{\omega_1^2} + C$ we see that $C = \frac{k_1}{\omega_1^2}$ and

$$x_1(t) = \frac{k_1}{\omega_1}(1 - \cos(\omega_1 t)) + k_2\left[\frac{t}{\omega_2} - \frac{1}{\omega_2^2}\sin(\omega_2 t)\right]\,.$$

With these expressions at $t_f = 10$ we have

$$\begin{aligned} x_2(10) &= \frac{k_1}{\omega_1}\sin(\pi) - \frac{k_2}{\omega_2}(1 - \cos(2\pi)) = 0 \\ x_1(10) &= \frac{k_1}{\omega_1^2}(1 - \cos(\pi)) + k_2\left[\frac{10}{(\pi/5)} - \frac{25}{\pi^2}\sin(2\pi)\right] = \frac{200}{\pi^2}k_1 + \frac{50}{\pi}k_2\,. \end{aligned}$$

The cost function from Part (c) then in terms of k_1 and k_2 becomes given what $x_1(t)$, $x_2(t)$ and $u(t)$ are

$$J(k_1, k_2) = q_1\left(\frac{200}{\pi^2}k_1 + \frac{50}{\pi}k_2 - 100\right)^2 + q_2(0)^2 + r\int_0^{10}(k_1\cos(\omega_1 t) + k_2\sin(\omega_2 t))^2 dt\,.$$

We next solve $\frac{\partial J}{\partial k_1} = 0$ and $\frac{\partial J}{\partial k_2} = 0$ to find the optimal solution for k_1 and k_2. This is made easier by taking the derivatives inside the t integrand and then evaluating the integrand of the derived result.

Notes on Example 3.4-1 the Cart on Track Part 2

Part (b): Our objective function J in this case is given by

$$J = \phi(x_f) + \int_{t_0}^{t_f}\mathcal{L}(t)dt\,,$$

where the end point condition is given by $\phi(x_{1f}) = q(x_{1f} - 100)^2$ and the expression for $\mathcal{L}(t)$ is given by $\mathcal{L}(t) = ru^2(t)$. To find the Lagrangian multipliers or $\lambda(t)$ for this problem we solve the ordinary differential equations

$$\begin{aligned} \frac{d\lambda(t)}{dt} &= -F^T\lambda(t) - \left[\frac{\partial\mathcal{L}}{\partial x}\right]^T \\ &= -\begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}^T\begin{bmatrix} \lambda_1(t) \\ \lambda_2(t) \end{bmatrix} = -\begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix}\begin{bmatrix} \lambda_1(t) \\ \lambda_2(t) \end{bmatrix} = -\begin{bmatrix} 0 \\ \lambda_1(t) \end{bmatrix}\,, \end{aligned} \tag{69}$$

with *final* conditions at $t = t_f$ given by

$$\lambda(t_f) = \left(\frac{\partial \phi}{\partial x}\right)^T \Bigg|_{t=t_f} = \begin{bmatrix} 2q(x_{1f} - 100) \\ 0 \end{bmatrix} = \begin{bmatrix} \lambda_1(t_f) \\ \lambda_2(t_f) \end{bmatrix}.$$

From the ordinary differential equation for $\lambda(t)$ we see that λ_1 can be solved since

$$\dot{\lambda}_1 = 0 \Rightarrow \lambda_1(t) = \lambda_1(t_f) = 2q(x_{1f} - 100)\,.$$

While for $\lambda_2(t)$ we have

$$\dot{\lambda}_2(t) = -\lambda_1(t) = -2q(x_{1f} - 100)\,,$$

thus

$$\lambda_2(t) = -2q(x_{1f} - 100)t + C\,,$$

for some constant C. Let $t = t_f$ where $\lambda_2(t_f) = 0$ to get $-2q(x_{1f} - 100)t_f + C = 0$, therefore $C = 2q(x_{1f} - 100)t_f$ so

$$\lambda_2(t) = -2q(x_{1f} - 100)(t - t_f)\,.$$

To find the control history we need to solve

$$\left[\frac{\partial \mathcal{L}}{\partial u}\right]^T + G^T(t)\lambda(t) = 0\,, \tag{70}$$

for $u(t)$. For this problem we have $G \equiv \frac{\partial f}{\partial u} = \frac{\partial}{\partial u}\begin{bmatrix} 0 \\ u \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$ and $\mathcal{L} = ru^2$ so the equation above becomes

$$2ru + \begin{bmatrix} 0 & 1 \end{bmatrix}\begin{bmatrix} \lambda_1(t) \\ \lambda_2(t) \end{bmatrix} = 0\,.$$

Solving the above for $u(t)$ gives

$$u(t) = -\frac{1}{2r}\lambda_2(t) = -\frac{1}{2r}(-2q(x_{1f} - 100)(t - t_f)) = \frac{q}{r}(x_{1f} - 100)(t - t_f)\,.$$

Note that this control can be written as an affine function of t as $u(t) = k_1 + k_2 t$ with

$$k_1 = -\frac{q}{r}(x_{1f} - 100)t_f \tag{71}$$

$$k_2 = \frac{q}{r}(x_{1r} - 100)\,. \tag{72}$$

In that case by integrating Equation 67 we get

$$x_{1f} = \frac{k_1}{2}t_f^2 + \frac{k_2}{6}t_f^3\,, \tag{73}$$

but since we have just shown that the *optimal* control $u(t) = k_1 + k_2 t$ has k_1 and k_2 given by Equations 71 and 72 we can put this into Equation 73 to get

$$x_{1f} = -\frac{1}{2}\frac{q}{r}(x_{1f} - 100)t_f^3 + \frac{1}{6}\frac{q}{r}(x_{1f} - 100)t_f^3\,.$$

Which we can solve for x_{1f}. Putting x_{1f} on one side we find

$$\left(1 + \frac{q}{2r}t_f^3 - \frac{q}{6r}t_f^3\right)x_{1f} = \frac{q}{2r}(100)t_f^3 - \frac{q}{6r}(100)t_f^3$$

or simplifying we get

$$x_{1f} = \frac{\frac{100q}{r}\left(\frac{1}{2} - \frac{1}{6}\right) t_f^3}{\left(1 + \frac{q}{r} t_f^3 \left(\frac{1}{2} - \frac{1}{6}\right)\right)} = \frac{100}{1 + \frac{3r}{qt_f^3}} = \frac{100}{1 + \frac{3r}{1000q}} \,.$$

Part (b): For this part we change our endpoint objective function $\phi(x_f)$ to be $\phi = q_1(x_{1f} - 100)^2 + q_2 x_{2f}^2$. With this expression for ϕ the final condition for the function $\lambda(t)$ is now

$$\lambda(t_f) = \left(\frac{\partial \phi}{\partial x}\right)^T \Bigg|_{t=t_f} = \begin{bmatrix} 2q_1(x_{1f} - 100) \\ 2q_2 x_{2f} \end{bmatrix} .$$

The ordinary differential equation for $\lambda(t)$ does not change so again we have

$$\lambda_1(t) = 2q_1(x_{1f} - 100) \,,$$

and $\lambda_2(t) = -2q_1(x_{1f} - 100)t + C$ for some constant C as before. From the final condition above C must satisfy

$$-2q_1(x_{1f} - 100)t_f + C = 2q_2 x_{2f} \,.$$

Or solving for C we get

$$C = 2q_2 x_{2f} + 2q_1(x_{1f} - 100)t_f \,.$$

Thus the entire function $\lambda_2(t)$ then becomes

$$\lambda_2(t) = -2q_1(x_{1f} - 100)(t - t_f) + 2q_2 x_{2f} \,,$$

To evaluate the control function $u(t)$ we consider Equation 70 where again we have $\frac{\partial \mathcal{L}}{\partial u} = 2ru$ and $G = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$. Thus Equation 70 becomes

$$2ru + \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} \lambda_1(t) \\ \lambda_2(t) \end{bmatrix} = 0 \,,$$

or $2ru + \lambda_2(t) = 0$ so $u(t) = -\frac{\lambda_2(t)}{2r}$. We again write the optimal control $u(t)$ in the affine form $k_1 + k_2 t$ where in this case from the above expression for $\lambda_2(t)$ we find

$$k_1 = -\frac{q_2 x_{2f} + q_1(x_{1f} - 100)t_f}{r}$$
$$k_2 = \frac{q_1(x_{1f} - 100)}{r} \,.$$

If we solve then for x_{1f} and x_{2f} in terms of k_1 and k_2 we find from the second equation above that

$$x_{1f} = \frac{r}{q_1} k_2 + 100 \,, \tag{74}$$

Then using this expression and the first equation above we see that

$$x_{2f} = -\frac{r}{q_2} k_1 - \frac{q_1}{q_2} t_f \left(\frac{r}{q_1} k_2\right) = -\frac{r}{q_2} k_1 - \frac{r}{q_2} k_2 t_f \,. \tag{75}$$

Since once the control $u(t)$ is specified in the affine form $u(t) = k_1 + k_2 t$ the final values of x_{1f} and x_{2f} are given by (we need to integrate $u(t)$ once to get the velocity and a second time to get the position) or

$$x_{2f} = k_1 t_f + \frac{k_2}{2} t_f^2 \quad \text{and} \quad x_{1f} = \frac{k_1}{2} t_f^2 + \frac{k_2}{6} t_f^3 .$$

We can put in what we know about x_{1f} from Equation 74 and x_{2f} from Equation 75 to get

$$\frac{r}{q_1} k_2 + 100 = \frac{k_1}{2} t_f^2 + \frac{k_2}{6} t_f^3$$
$$-\frac{r}{q_2} k_1 - \frac{r}{q_2} k_2 t_f = k_1 t_f + \frac{k_2}{2} t_f^2 ,$$

which is a linear system for k_1 and k_2.

Warning: When we solve for k_1 and k_2 in the above linear system we don't seem to get a result that matches the book. If anyone sees an error in what I have done (or agrees with me) please contact me.

Notes on the Hamilton-Jacobi-Bellman Equation

For the optimal value function $V^*[x(t), t]$ we have the total derivative at t_1 given by

$$\left.\frac{dV^*}{dt}\right|_{t=t_1} = \left.\left(\frac{\partial V^*}{\partial t} + \frac{\partial V^*}{\partial x}\dot{x} + \frac{\partial V^*}{\partial u}\dot{u}\right)\right|_{t=t_1} .$$

If we are on the optimal path then $V^*(t) = J_{\max} - J(t)$ when $t < t_f$ so

$$\frac{\partial V^*}{\partial u} = -\frac{\partial J}{\partial u} ,$$

which vanishes due to the first order optimality conditions of u on J. Therefore dropping the term $\frac{\partial V^*}{\partial u}$ in the expression for $\left.\frac{dV^*}{dt}\right|_{t=t_1}$ we have that

$$\left.\frac{dV^*}{dt}\right|_{t=t_1} = \left.\left(\frac{\partial V^*}{\partial t} + \frac{\partial V^*}{\partial x}\dot{x}\right)\right|_{t=t_1} . \tag{76}$$

Since the total derivative of the optimal value function is also the (negative) Lagrangian evaluated pointwise on the optimal trajectory or

$$\left.\frac{dV^*}{dt}\right|_{t=t_1} = -\mathcal{L}[x^*(t_1), u^*(t_1), t_1] , \tag{77}$$

we can equate this with the result in Equation 76 to get the partial derivative of V^* with respect to t or

$$\left.\frac{\partial V^*}{\partial t}\right|_{t=t_1} = -\mathcal{L}[x^*(t_1), u^*(t_1), t_1] - \left.\frac{\partial V^*}{\partial x}\right|_{t=t_1} f[x^*(t_1), u^*(t_1), t_1] \tag{78}$$
$$= -\mathrm{Min}_u \mathcal{H}(t) , \tag{79}$$

on the optimal trajectory. These last two equations are known as the **Hamilton-Jacobi-Bellman** (HJB) equation.

Notes on terminal state equality constraints

In this section we want to consider control problems where we have a "hard" end point constraints i.e. where the constraints take the form $\psi[x(t_f), t_f] = 0$. We will combine the original optimization objective J_0 with one used to enforce these new end point constraints $\psi[x(t_f), t_t] = 0$ (denoted J_1). We combine these two expressions in our total objective function

$$J_c = J_0 + \mu J_1 \,. \tag{80}$$

We assume a function J_1 where the optimal solution is to minimize the value of $\psi[x(t_f), t_f]$ subject to the dynamic constraints $\dot{x} = f[x(t), u(t), t]$. Thus we need to consider the function

$$J_1 = \psi[x(t_f), t_f] + \int_{t_0}^{t_f} \lambda_1^T (f - \dot{x}) dt = \psi + \int_{t_0}^{t_f} (\mathcal{H}_1 - \lambda_1^T \dot{x}) dt \,.$$

The first two Euler-Lagrange equations to minimize this J_1 are given by

$$\dot{\lambda}_1(t) = -\frac{\partial \mathcal{H}_1}{\partial x}^T = -F^T \lambda_1 \quad \text{with} \tag{81}$$

$$\lambda_1(t_f) = \left(\frac{\partial \psi}{\partial x} \right)^T \Bigg|_{t=t_f} . \tag{82}$$

These equations must hold for all t in the domain $t_1 \leq t \leq t_f$ and will be the equation used to determine $\lambda_1(t)$. To enforce the dynamic constraint under the part J_0 part of the objective function J we will require to solve

$$\dot{\lambda}_0 = -\left(\frac{\partial \mathcal{H}_0}{\partial x} \right)^T = -F^T \lambda_0 - \left(\frac{\partial \mathcal{L}}{\partial x} \right)^T , \tag{83}$$

with a final condition as we would have in the case where we don't worry about the boundary constraint.

$$\lambda_0(t_f) = \left(\frac{\partial \phi}{\partial x} \right)^T \Bigg|_{t=t_f} \tag{84}$$

We are finished when we can link the control $u(t)$ to these two functions $\lambda_0(t)$ and $\lambda_1(t)$ with

$$\frac{\partial \mathcal{H}_c}{\partial u} = \left(\frac{\partial \mathcal{L}}{\partial u} + \lambda_0^T G \right) + \mu \lambda_1^T G = 0 \,, \tag{85}$$

which introduced the constant μ. This in tern is given by

$$\mu = -\frac{a}{b} \quad \text{where} \tag{86}$$

$$a = \int_{t_0}^{t_f} \lambda_1^T G \left[\left(\frac{\partial \mathcal{L}}{\partial u} \right)^T + G^T \lambda_0 \right] dt \tag{87}$$

$$b = \int_{t_0}^{t_f} \lambda_1^T G G^T \lambda_1 dt \,. \tag{88}$$

In summary, to solve this type of control problem we need to specify that all of the above expressions hold true and a solution requires the specification of the functions $\lambda_0(t)$, $\lambda_1(t)$ and $u(t)$. Once we have $x(t)$ and $u(t)$ expressed in terms of μ we might be able to solve for the scalar Lagrangian μ using the point constraint $\psi[x(t_f), t_f] = 0$. An example of this type of problem is given on Page 76.

Notes on singular control

WWX: Proof this section

$$\mathcal{H} = u^T(t)R + \lambda(t)^T G\,.$$

If we take $R = 0$ we have $\mathcal{H}u = \lambda^T(t)G$ so that

$$\frac{d}{dt}(\mathcal{H}u) = \dot{\lambda}^T G\,.$$

From the Euler-Lagrange conditions we have $\dot{\lambda}(t) = -\frac{\partial \mathcal{H}}{\partial x}^T$ we get

$$\frac{d}{dt}\mathcal{H}_u = -\left(\frac{\partial H}{\partial x}\right) G\,.$$

From the form of the Hamiltonian with $R = 0$ we get

$$\mathcal{H} = \frac{1}{2}(x^T(t)Qx(t)) + \lambda^T(Fx + Gu)\,.$$

so we have

$$\frac{d}{dt}\mathcal{H}_u = -(x^T Q + \lambda^T F)G = 0\,. \tag{89}$$

$$\begin{aligned}\frac{d^2}{dt^2}\mathcal{H}_u &= -(\dot{x}(t)^T Q + \dot{\lambda}^T F)G \\ &= -\left[(Fx(t) + Gu(t))^T Q - (x^T(t)Q + \lambda^T F)^T F\right] G\,.\end{aligned} \tag{90}$$

Notes on finding optimal controls numerically with the quasilinear method

WWX: Proof this section

$$\dot{x}(t) = f[x(t), u_0(t), t] = f'[x(t), \lambda_0(t), t]$$

We guess $x_0(t)$ and $\lambda_0(t)$ in $[t_0, t_f]$. Let $x(t) = x_0(t) + \Delta x_0(t)$ and put

$$\lambda = \lambda_0 + \Delta\lambda_0\,.$$

Thus we get

$$\begin{aligned}\dot{x} = \dot{x}_0 + \dot{\Delta x}_0 &= f'[x_0 + \Delta x_0, \lambda_0 + \Delta\lambda_0, t] \\ &= f'[x_0, \lambda_0, t] \\ &+ \frac{\partial}{\partial x} f'[x_0, \lambda_0, t]\Delta x_0 + \frac{\partial}{\partial \lambda_0} f'[x_0, \lambda_0, t]\Delta\lambda_0\,.\end{aligned}$$

Thus

$$\dot{x}_0 - f'[x_0, \lambda_0, t] = -\Delta\dot{x}_0 + \frac{\partial}{\partial x} f'[x_0, \lambda_0, t]\Delta x_0 + \frac{\partial}{\partial \lambda_0} f'[x_0, \lambda_0, t]\Delta\lambda_0\,. \tag{91}$$

Let $\lambda(t) = \lambda_0(t) + \Delta\lambda(t)$ we have

$$\begin{aligned}\dot{\lambda}(t) &= \dot{\lambda}_0(t) + \dot{\Delta\lambda}(t) \\ &= -\frac{\partial}{\partial x}\mathcal{H}^T[x_0 + \Delta x_0, \lambda_0 + \Delta\lambda_0] \\ &= -\left[\frac{\partial}{\partial x}\mathcal{H}^T[x_0, \lambda_0] + \frac{\partial}{\partial x}\mathcal{H}_x^T[x_0, \lambda_0]\Delta x_0 + \frac{\partial}{\partial \lambda}\mathcal{H}_x^T[x_0, \lambda_0]\Delta\lambda_0\right].\end{aligned}$$

Thus

$$\begin{aligned}\dot{\lambda}_0(t) + \mathcal{H}_x^T[x_0(t), \lambda_0(t), t] \approx &-\dot{\Delta\lambda}(t) \\ &- \frac{\partial}{\partial x}\mathcal{H}_x^T[x_0(t), \lambda_0(t)]\Delta x_0 \\ &- \frac{\partial}{\partial \lambda}\mathcal{H}_x^T[x_0(t), \lambda_0(t)]\Delta\lambda_0\,.\end{aligned}$$

Think want to write this system as if x_0 and λ_0 are solutions to $\dot{x}_0 - f'[x_0, \lambda_0, t] = 0$, then

$$\dot{\lambda x}_0 = \frac{\partial f'}{\partial x}[]\Delta x_0 + \frac{\partial f'}{\partial \lambda_0}[]\Delta\lambda_0 + [\dot{x}_0 - f']\,.$$

WWX: place these notes somewhere

$G = \begin{bmatrix} 0 \\ u_{\max} \end{bmatrix}$ so

$$Gu^* = \begin{bmatrix} 0 \\ u_{\max} \end{bmatrix}\tilde{u}^* = \begin{bmatrix} 0 \\ u_{\max}\tilde{u}^* \end{bmatrix}.$$

we have

$$\lambda^{*T} G u^* = \cdots = \lambda_2^*(t) u_{\max}\tilde{u}^*\,,$$

so minimum control picks $u^*(t)$ such that $\lambda_2^*(t)u_{\max}\tilde{u}^*$ is as negative (small) as possible. Solve adjoint equation for $\lambda(t)$.

$$\dot{\lambda}(t) = -F^T\lambda(t) = -\begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix}\lambda\,,$$

with

$$\lambda(t_f) = \left.\frac{\partial\psi}{\partial x}\right|_{t=t_f} = \begin{bmatrix} 2c_1(x(t_f) - 1) \\ 2c_2 x_2(t_f) \end{bmatrix}.$$

Then

$$\psi(x) = c_1(x_1 - 1)^2 + c_2(x_2 - 0)^2\,.$$

Thus

$$\dot{\lambda}_1 = 0\,,$$

$$\lambda_1(t_f) = 2c_1(x_1(t_f) - 1) \Rightarrow \lambda_1(t) = 2c_1(x_1(t_f) - 1)\,.$$

The two point boundary problem is

$$\begin{aligned}
\dot{x}_1 &= -\frac{1}{T_1}(x_1 - x_{\text{air}}) + k_1(x_2 - x_1) - \frac{k_2^2}{2c_2}\lambda_1 \\
\dot{x}_2 &= -\frac{1}{T_2}(x_2 - x_{\text{air}}) + k_1(x_1 - x_2) \\
\dot{\lambda}_1 &= \left(\frac{1}{T_1} + k_1\right)\lambda_1 + k_1\lambda_2 - 2c_1(x_1 - x_d) \\
\dot{\lambda}_2 &= -k_1\lambda_1 + \left(\frac{1}{T_2} + k_1\right)\lambda_2 - 2c_1(x_2 - x_d) \\
x_1(0) &= x_{01} \\
x_2(0) &= x_{02} \\
\lambda_1(t_f) &= 2c_1(x_{1f} - x_d) \\
\lambda_2(t_f) &= 2c_1(x_{2f} - x_d)\,.
\end{aligned}$$

Set up MMA to solve the problem needed.

Problem Solutions

Problem 3.3.1 (a different final velocity at the time $t_f = 10$)

In this case where we desire the final velocity to be 10 we should take $x_2(t_f) = 10$ rather than 0. This means that the constant (i.e. hard) constraint becomes

$$10 = 10k_1 + 50k_2\,.$$

If we want to use a "soft" constraint all that changes is to add a term

$$q_2\left(k_1 t_f + \frac{1}{2}k_2 t_f^2 - 10\right)^2\,,$$

to the expression for J. We could then solve $\frac{\partial J}{\partial k_1} = 0$ and $\frac{\partial J}{\partial k_2} = 0$ for k_1 and k_2 to compute the optimal solution.

Problem 3.3.2 (control of rocket equations)

For the problem given here $v(t)$ is the velocity, $\gamma(t)$ is the flight path angle, $h(t)$ is the altitude, and $m(t)$ is the mass of an ascending rocket. The dynamical equations that govern

its motion are given by

$$\begin{aligned}
\dot{v} &= \frac{1}{m}(T\cos(\alpha) - D - g\sin(\gamma)) \\
\dot{\gamma} &= \frac{1}{mv}(T\sin(\alpha) + L - g\cos(\gamma)) \\
\dot{h} &= v\sin(\gamma) \\
\dot{m} &= f(T)\,.
\end{aligned}$$

Since we are told that T is a constant this last equation can be integrated to give $m(t) = f(T)t + m(0)$, were $m(0)$ is the initial condition on $m(t)$.

Part (a): With the suggested explicit linear form for the control $\alpha(t) = \alpha_0 + \alpha_1 t$ the above ordinary differential equations become

$$\begin{aligned}
\dot{v} &= \frac{1}{m}(T\cos(\alpha_0 + \alpha_1 t) - D - g\sin(\gamma)) \\
\dot{\gamma} &= \frac{1}{mv}(T\sin(\alpha_0 + \alpha_1 t) + L - g\cos(\gamma)) \\
\dot{h} &= v\sin(\gamma)\,.
\end{aligned} \tag{92}$$

The objective function we seek to minimize is

$$J = \frac{1}{2}\left\{(x(t_f) - x_d)^T Q(x(t_f) - x_d) + r\int_0^{t_f} \alpha^2(t)dt\right\}\,, \tag{93}$$

where Q is a positive definite matrix. Given the functional form for $\alpha(t)$ we can simplify the integral term as

$$\int_0^{t_f} \alpha^2(t)dt = \int_0^{t_f} (\alpha_0 + \alpha_1 t)^2 dt = \alpha_0 t_f + \alpha_0\alpha_1 {t_f}^2 + \frac{1}{3}\alpha_1^2 {t_f}^3\,.$$

Thus the total expression for J becomes

$$J = \frac{1}{2}\left\{(x(t_f) - x_d)^T Q(x(t_f) - x_d) + r\left(\alpha_0 t_f + \alpha_0\alpha_1 {t_f}^2 + \frac{1}{3}\alpha_1^2 {t_f}^3\right)\right\}\,, \tag{94}$$

In the above expression the final value of our state $x(t_f)$ depend on the explicit values taken for α_0 and α_1 during the integration of the nonlinear differential equations 92. Since we cannot integrate these equations analytically we will will implicitly view the expression $x(t_F)$ as $x(t_f; \alpha_0, \alpha_1)$ since the final state value at the time t_f depends on values for α_0 and α_1. The minimum of J considered as a function of α_0 and α_1 can be obtained by solving

$$\frac{\partial J}{\partial \alpha_0} = \frac{\partial J}{\partial \alpha_1} = 0\,,$$

for α_0 and α_1. I don't see how to explicitly take the α_0 and α_1 derivative of the endpoints expression $x(t_f)$. Thus to compute the minimization of J we will have to result in evaluating it *numerically*.

Part (b): When we specify the numbers given for this part of the problem we have $f(T) = 0$ so $m(t) = m(0) = 20$. The differential equation for the other functions v, γ, and h becomes

$$\begin{aligned}
\dot{v} &= \frac{1}{20}(10000\cos(\alpha_0 + \alpha_1 t) - 32\sin(\gamma)) \quad v(0) = 100\,, \quad v_d \text{ unspecified} \\
\dot{\gamma} &= \frac{1}{20v}(10000\sin(\alpha_0 + \alpha_1 t) - 32\cos(\gamma)) \quad \gamma(0) = \frac{\pi}{2}\,, \quad \gamma_d = \frac{\pi}{3} \\
\dot{h} &= v\sin(\gamma) \quad h(0) = 0\,, \quad h_d \text{ unspecified}\,.
\end{aligned} \tag{95}$$

The objective function with $Q = \mathrm{Diag}(0, d_\gamma, 0)$ we seek to minimize is given by

$$J = \frac{1}{2}\left\{ d_\gamma \left(\gamma(t_f) - \frac{\pi}{3}\right)^2 + \alpha_0^2 t_f + \alpha_0\alpha_1 {t_f}^2 + \frac{1}{3}\alpha_1^2 {t_f}^3 \right\}\,. \tag{96}$$

Note that in Equations 95 once the first two equations are solved for $v(t)$ and $\gamma(t)$ the third function $h(t)$ is then directly computed (using the third equation) from the previous two. Since our objective function J in Equation 96 in this case does not depend on $h(t)$ when we numerically evaluate the above ordinary differential equation we don't need to compute it. To finish this problem then we will need to numerically evaluate the function J. This is done in the following way

- First integrate the first two Equations 95 starting at $t = 0$ to get the values of $v(t)$ and $\gamma(t)$ at $t = t_f$ (only $\gamma(t_f)$ is needed).
- Second evaluate Equation 96 with the just computed value of $\gamma(t_f)$.

This procedure is implemented in the Matlab code `sect_3_prob_2_part_b.m`. When that script is run we find the optimal values given by $\alpha_0 = -0.1379$ and $\alpha_1 = 0.0170$. At that value we find $J_{\text{opt}} = 0.0349$.

Part (c): For this part matrix Q is now $\mathrm{Diag}(0, 10, 0)$ so the constant d_γ has increased 10 times. We can use the same script as above to solve this problem. When that script is run we find the optimal values given by $\alpha_0 = -0.1785$ and $\alpha_1 = 0.0220$. At that value we find $J_{\text{opt}} = 0.0451$. Since this change has increased the importance of the final value of $\gamma(t_f)$ matching the desired value of $\gamma_d = \frac{\pi}{3} = 1.0472$ we can compare how close the value of $\gamma(t_f)$ from Part (a) and Part (b) are to this target.

$$\gamma_d - \gamma_{f,\text{Part(b)}} = -0.1328\,, \quad \gamma_d - \gamma_{f,\text{Part(c)}} = -0.0170\,,$$

showing that our error in γ_d is better in this case.

Part (d): I'll assume that there is a typo in the book and it is supposed to read $f(T) = -1$ (rather than $f(T) = +1$). Then in this case then $m(t) = -t + m(0) = -t + 20$, so that $m(10) = 10$. This means that the mass of the rocket *decreases* as time passes (presumably from burning fuel). Our dynamic equations are now given by

$$\begin{aligned}
\dot{v} &= \frac{1}{20 - t}(10000\cos(\alpha_0 + \alpha_1 t) - 0.0004v^2 - 32\sin(\gamma)) \\
\dot{\gamma} &= \frac{1}{(20 - t)v}(10000\sin(\alpha_0 + \alpha_1 t) - 32\cos(\gamma)) \\
\dot{h} &= v\sin(\gamma)\,.
\end{aligned}$$

The initial conditions are the same as in Part (c). Again nothing in J depends on $h(t)$ and it can be dropped from consideration. This is the same type of problem as discussed before and to solve it requires some simple modifications of the previous codes. It is worked in the Matlab script `sect_3_prob_2_part_d.m`. When that script is run with $d_\gamma = 1$ we find the optimal values given by $\alpha_0 = -0.1285$ and $\alpha_1 = 0.0144$ with $J_{\text{opt}} = 0.0320$.

Part (e): In this case we introduced a dependence on $h(t)$ and must evaluate it as a function of time as we progress. This is because when $h_d = 20000$ with $Q = \text{Diag}(0, 10, 0.01)$ then the objective function J becomes

$$J = \frac{1}{2}\left\{10\left(\gamma(t_f) - \frac{\pi}{3}\right)^2 + 0.01(h(t_f) - 20000)^2 + \alpha_0^2 t_f + \alpha_0\alpha_1 {t_f}^2 + \frac{1}{3}\alpha_1^2 {t_f}^3\right\} .$$

This part is worked in the Matlab script `sect_3_prob_2_part_e.m`. When that script is run we find the optimal values given by $\alpha_0 = 0.4290$ and $\alpha_1 = -0.1689$ with $J_{\text{opt}} = 2.4712$.

Problem 3.3.3 (more control of the rocket equations)

For the suggested functional form for $\alpha(t)$ we have that the state variables are governed by the following ordinary differential equations

$$\begin{aligned}\dot{v} &= \frac{1}{m}(T\cos(\alpha_0 + \alpha_1\sin(\omega t) + \alpha_2\cos(\omega t)) - D - g\sin(\gamma)) \\ \dot{\gamma} &= \frac{1}{mv}(T\sin(\alpha_0 + \alpha_1\sin(\omega t) + \alpha_2\cos(\omega t)) + L - g\sin(\gamma)) .\end{aligned}$$

with $\omega = \frac{2\pi}{t_f}$. We don't need the consider the differential equation for $h(t)$ since it is not used in the expression for J. With the numbers given in the text these equations become

$$\begin{aligned}\dot{v} &= \frac{1}{20}(10000\cos(\alpha_0 + \alpha_1\sin(\omega t) + \alpha_2\cos(\omega t)) - 32\sin(\gamma)) \\ \dot{\gamma} &= \frac{1}{20v}(10000\sin(\alpha_0 + \alpha_1\sin(\omega t) + \alpha_2\cos(\omega t)) - 32\sin(\gamma)) ,\end{aligned}$$

where we have taken $m(t) = 20$. The expression J we want to minimize is given by

$$J = \frac{1}{2}\left\{d_\gamma(\gamma(t_f) - \gamma_d)^2 + r\int_0^{t_f}(\alpha_0 + \alpha_1\sin(\omega t) + \alpha_2\cos(\omega t))^2 dt\right\} .$$

When we evaluate the integral above we get for J

$$J = \frac{1}{2}\left\{d_\gamma(\gamma(t_f) - \gamma_d)^2 + \frac{r}{2}(2\alpha_0^2 + \alpha_1^2 + \alpha_2^2)t_f\right\} .$$

Since $\gamma(t_f)$ depends on the control inputs α_0, α_1, and α_2 we will complete the minimization of J numerically. This is done in the Matlab function `sect_3_prob_3.m`. When that script is run we find optimal values given by $\alpha_0 = -0.0835$, $\alpha_1 = -0.0467$, and $\gamma_2 = -0.0706$, with a J value given by $J = 0.0549$.

Problem 3.3.4 (even more control of the rocket equations)

For this problem we assume that α and T are the input controls and are constant. Since if T is a input control then to integrate the equation $\dot{m} = f(T)$ we would need to know what the functional form for $f(\cdot)$ is. Since this is not given we will assume that the ordinary differential equation for $m(t)$ is $\dot{m} = -1$ with $m(0) = 20$ and thus $m(t) = 20 - t$. The other dynamical variables evolve according to

$$\begin{aligned}
\dot{v} &= \frac{1}{20-t}(T\cos(\alpha) - 32\sin(\gamma)) \\
\dot{\gamma} &= \frac{1}{(20-t)v}(T\sin(\alpha) - 32\sin(\gamma)) \\
\dot{h} &= v\sin(\gamma)\,.
\end{aligned}$$

with the expression for J given by

$$J = \frac{1}{2}\left\{q_v(v(t_f) - v_d)^2 + q_\gamma(\gamma(t_f) - \gamma_d)^2 + q_h(h(t_f) - h_d)^2 + r_1\alpha^2 t_f + r_2 T^2 t_f\right\}\,,$$

when we integrate the integral term in J. Note that $v(t_f)$, $\gamma(t_f)$, and $h(t_f)$ depend implicitly on the input control values α and T and thus we will do the minimization of J numerically. We implement the numerical minimization in the Matlab routine `sect_3_prob_4.m`. When it is run we get $\alpha = 0.0$ and $T = 91.42$, with $J = 2.9927\,10^6$.

Problem 3.3.5 (control of an oscillator)

Part (a): For this problem we assume a control $u(t)$ in terms of the state $x_1(t)$ and $x_2(t)$ given by

$$u(t) = -c_1(x_1(t) - x_{1d}) - c_2 x_2(t)\,,$$

with a cost function given by

$$J = \frac{1}{2}\left\{q(x_1(t_f) - x_{1d})^2 + \int_0^{t_f} ru^2(t)dt\right\}\,.$$

We can't evaluate J explicitly since we don't know the optimal values of c_1 and c_2 and thus don't know the functional form for $u(t)$. To solve this problem we will select values for c_1 and c_2, solve the following ordinary differential equation

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\omega_n^2 & -2\xi\omega_n \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ \omega_n^2 \end{bmatrix} (-c_1(x_1 - x_{1d}) - c_2 x_2)\,,$$

starting with $x_1(0) = x_2(0) = 0$ and for $0 \le t \le t_f$. This gives the final value $x_1(t_f)$ and the function $u(t)$ which we can use to evaluate J. This is done in the Matlab file `sect_3_prob_5.m`. When we run that script we find $c_1 = 3.5121$, $c_2 = 31.1545$, and $J = 0.3234$ with $u(t)$ with these values plotted in Figure 11.

Part (b): This part is worked in the same way as Part (a). It can be found in the same Matlab file as above.

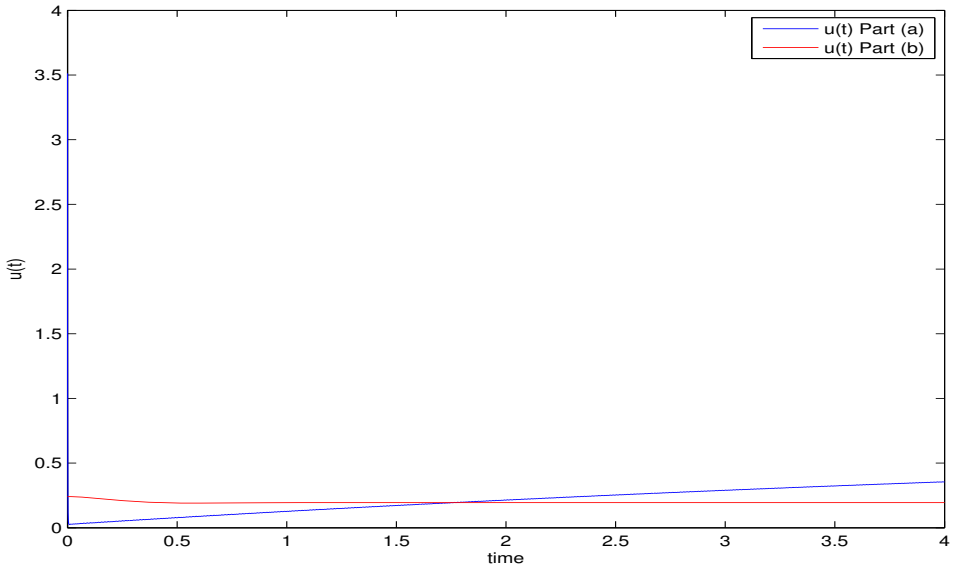

Figure 11: The controls $u(t)$ for Part (a) in blue and Part (b) in red for Problem 5.

Problem 3.4.1 (rate of pollution increase)

For this problem we assume a Lagrangian given by

$$\mathcal{L}[x(t), u(t), t] \equiv (1 - e^{-u/u_0}) \left(1 - \left(\frac{x}{x_0}\right)^n\right) e^{-ct} ,$$

such that we want to minimize the functional J defined as

$$J = \int_0^{t_f} \mathcal{L}[x(t), u(t), t] dt .$$

The dynamic equations for this problem are governed by

$$\dot{x}(t) = f(x(t), u(t)) = -ax(t) + bu(t) .$$

To derive the Euler-Lagrange equations we augment to the cost function J, the dynamic constraint by introducing a Lagrangian multiplier $\lambda(t)$ to get

$$J_A = \int_{t_0}^{t_f} \{\mathcal{L}[x(t), u(t), t] + \lambda(t)(\dot{x}(t) - f(x(t), u(t), t))\} \, dt .$$

Then the Euler-Lagrange equations in terms of the Lagrangian $\mathcal{L}[x, u, t]$ the and the dynamic function $f(x, u)$ are the following coupled differential and algebraic equations for $x(t)$, $\lambda(t)$, and $u(t)$.

- Solve the ordinary differential equation

$$\dot{\lambda}(t) = -\left(\frac{\partial f}{\partial x}\right)^T \lambda(t) - \left(\frac{\partial \mathcal{L}}{\partial x}\right)^T , \tag{97}$$

with a *final* condition given by

$$\lambda(t_f) = \left.\frac{\partial \phi}{\partial x}^T\right|_{t=t_f} . \tag{98}$$

- Solve the ordinary differential equation

$$\dot{x} = f(x,u)\,, \tag{99}$$

with a *initial* condition given by

$$x(t_0) = x_0\,. \tag{100}$$

- Solve the algebraic equation

$$\left(\frac{\partial \mathcal{L}}{\partial u}\right)^T + \left(\frac{\partial f}{\partial u}\right)^T \lambda(t) = 0 \tag{101}$$

These give three equation for the three unknowns $x(t)$, $\lambda(t)$, and $u(t)$. For the specific $f(x,u)$ and $\mathcal{L}$ given in this problem we have

$$\begin{aligned} F \equiv \frac{\partial f}{\partial x} = -a \quad \text{and} \quad G \equiv \frac{\partial f}{\partial u} = b \\ \frac{\partial \mathcal{L}}{\partial x} = (1 - e^{-u/u_0}) \left(-n \left(\frac{x}{x_0}\right)^{n-1} \frac{1}{x_0} \right) e^{-ct} \\ \frac{\partial \mathcal{L}}{\partial u} = \frac{1}{u_0} e^{-u/u_0} \left(1 - \left(\frac{x}{x_0}\right)^n \right) e^{-ct}\,, \end{aligned}$$

and we get for the Euler-Lagrange equations

- Solve the ordinary differential equations

$$\begin{aligned} \dot{\lambda} &= a\lambda + \frac{n}{x_0}(1 - e^{-u/u_0}) \left(\frac{x}{x_0}\right)^{n-1} e^{-ct} \quad \text{with} \quad \lambda(t_f) = 0 \\ \dot{x} &= -ax + bu \quad \text{with} \quad x(t_0) = x_0\,. \end{aligned}$$

- The functions $x(t)$, $\lambda(t)$, and $u(t)$ must also satisfy

$$\frac{1}{u_0} e^{-u/u_0} \left(1 - \left(\frac{x}{x_0}\right)^n \right) e^{-ct} + b\lambda = 0\,.$$

Problem 3.4.2 (rate of pollution increase – a numerical example)

For the numbers specified in the problem, the Euler-Lagrange equations become

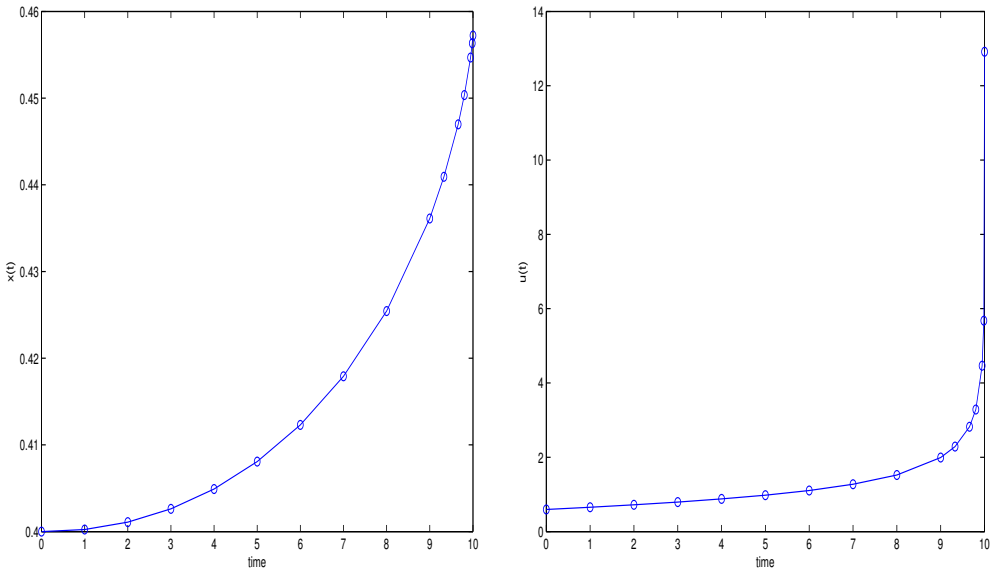

Figure 12: **Left:** The state estimate $x(t)$ for Chapter 3 Section 4 Problem 2 (rate of pollution increase). **Right:** The control $u(t)$ for the same problem.

- Solve the ordinary differential equations

$$\dot{\lambda} = 0.015\lambda + 8(1 - e^{-u})x \quad \text{with} \quad \lambda(10) = 0$$
$$\dot{x} = -0.015x + 0.01u \quad \text{with} \quad x(0) = 0.4\,.$$

- With the constraint that

$$e^{-u}(1 - 4x^2) + 0.01\lambda = 0\,.$$

We have to solve these expressions numerically to determine the functions $x(t)$, $\lambda(t)$, and $u(t)$. To do that we will use the "neighboring extremal method" which is a shooting based method applicable for nonlinear two-point boundary value problem. This numerical procedure is implemented in the Matlab files `sect_4_prob_2.m` which calls the functions:

- `sect_4_prob_2_J_min_fn.m`
- `sect_4_prob_2_J_ode_fn.m`
- `sect_4_prob_2_u_from_x_N_lambda.m`.

When that script is run we find that the initial condition for $\lambda(t)$ is $\lambda_0 \approx -19.8753$ and obtain the plots shown in Figure 12.

Problem 3.4.3 (a mechanical spring mass system)

Our dynamic equation is specified with $\dot{x} = f(x,u)$ where $f(x,u)$ is given by

$$f(x,u) = \begin{bmatrix} 0 & 1 \\ -\omega_n^2 & -2\xi\omega_n \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ \omega_n^2 \end{bmatrix} u\,.$$

Here the variable x is a 2×1 vector and u is a scalar. We then can compute $F \equiv \frac{\partial f}{\partial x}$ and $G \equiv \frac{\partial f}{\partial u}$ and find

$$F = \begin{bmatrix} 0 & 1 \\ -\omega_n^2 & -2\xi\omega_n \end{bmatrix} \quad \text{and} \quad G = \begin{bmatrix} 0 \\ \omega_n^2 \end{bmatrix}$$

We will assume that our cost function J is quadratic

$$\begin{aligned} J &= \int_{t_0}^{t_f} \begin{bmatrix} x(t)^T & u(t) \end{bmatrix} \begin{bmatrix} Q & m \\ m^T & r \end{bmatrix} \begin{bmatrix} x(t) \\ u(t) \end{bmatrix} dt \\ &= \int_{t_0}^{t_f} [x^T(t)Qx(t) + 2u(t)x^T(t)m + ru(t)^2]dt\,. \end{aligned} \tag{102}$$

Here Q is a 2×2 matrix, m is a 2×1 vector, and r is a scalar. Thus for this problem

$$\mathcal{L} = x^T(t)Qx(t) + 2u(t)x^T(t)m + ru(t)^2 \quad \text{and} \quad \phi[x(t_f), t_f] = 0\,.$$

Thus the derivatives needed for the Euler-Lagrange Equations 97 and 101 are

$$\left(\frac{\partial \mathcal{L}}{\partial x}\right)^T = 2Qx + 2um \quad \text{and} \quad \frac{\partial \mathcal{L}}{\partial u} = 2x^T m + 2ru\,.$$

With these we can now write down the Euler-Lagrange equation. We must find consistent solutions to the ordinary differential equations

$$\begin{aligned} \frac{d}{dt}\begin{bmatrix} \lambda_1 \\ \lambda_2 \end{bmatrix} &= -\begin{bmatrix} 0 & -\omega_n^2 \\ 1 & -2\xi\omega_n \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \lambda_2 \end{bmatrix} - 2\begin{bmatrix} q_{11} & q_{12} \\ q_{21} & q_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} - 2\begin{bmatrix} m_1 \\ m_2 \end{bmatrix} u \\ &\qquad \text{with} \begin{bmatrix} \lambda_1(t_f) \\ \lambda_2(t_f) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \\ \frac{d}{dt}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} &= \begin{bmatrix} 0 & 1 \\ -\omega_n^2 & -2\xi\omega_n \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ \omega_n^2 \end{bmatrix} u \\ &\qquad \text{with} \begin{bmatrix} x_1(t_0) \\ x_2(t_0) \end{bmatrix} = \begin{bmatrix} x_{10} \\ x_{20} \end{bmatrix}, \end{aligned}$$

and the algebraic equation

$$2x^T m + 2ur + \begin{bmatrix} 0 & \omega_n^2 \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \lambda_2 \end{bmatrix} = 0\,,$$

or in component form

$$2(x_1 m_1 + x_2 m_2) + 2ru + \omega_n^2 \lambda_2 = 0\,.$$

Problem 3.4.4 (the temperature in an oven)

The dynamics of our state $x(t)$ is given by

$$\begin{aligned}\dot{x} &= -k_1(x - T_s) - k_2(x^4 - T_s^4) - k_3 u + k_4 \\ &= -k_1 x - k_2 x^4 + k_3 u + (k_1 T_s + k_2 T_s^3)\,,\end{aligned}$$

the book defines the constant k_4 to be $k_4 \equiv k_1 T_s + k_2 T_s^4$. We assume there is a typo, the book is missing the square on the difference $x(t_f) - x_d$, in the expression for J such that our cost function is

$$J = q(x(t_f) - x_d)^2 + \int_0^{t_f} r u^2 dt\,.$$

Thus for this problem we have

$$\begin{aligned}\mathcal{L}[x, u, t] &= r u^2 \\ f(x, u) &= -k_1 x - k_2 x^4 + k_3 u + k_4\,.\end{aligned}$$

Since $\mathcal{L}$ is not an explicit function of time we know that $\frac{d\mathcal{H}}{dt} = 0$ and thus $\mathcal{H}$ is independent of time. Our control "law" is then

$$\begin{aligned}\mathcal{H}[x, u] &= \mathcal{L}[x, u] + \lambda^T f(x, u) \\ &= r u^2 + \lambda(-k_1 x - k_2 x^4 + k_3 u + k_4) = \text{a constant}\,.\end{aligned} \tag{103}$$

The functions $\lambda(t)$, $x(t)$, and $u(t)$ must be consistent with Equations 97 and 99 or

$$\begin{aligned}\dot{\lambda} &= (k_1 + 3k_2 x^3)\lambda \quad \text{with} \quad \lambda(t_f) = 2q(x(t_f) - x_d) \\ \dot{x} &= -k_1 x - k_2 x^4 + k_3 u + k_4 \quad \text{with} \quad x(0) = x_0\,.\end{aligned}$$

Problem 3.4.5 (chemical reactor dynamics)

Warning: I'm not entirely sure this is what is requested for this problem. If anyone has a better solution (or agrees with me) please contact me.

The Hamilton-Jacobi-Bellman (HJB) equation for the optimal value function V^* and this problem is given by specifying Equation 78 to this specific problem. When we do that we get

$$\begin{aligned}\left.\frac{\partial V^*}{\partial t}\right|_{t=t_1} &= -\left[\mathcal{L}[x^*(t_1), u^*(t_1), t_1] + \left.\frac{\partial V^*}{\partial x}\right|_{t=t_1} f[x^*(t_1), u^*(t_1), t_1]\right] \\ &= -\left[(x(t) - x_d)^2 + u^2 + (ax^n + bu)\frac{\partial V}{\partial x}\right]\Bigg|_{t=t_1}\,.\end{aligned}$$

In the second equation we have dropped the asterisk on $x(t)$ and $u(t)$. If we assume that the functional form for $V(t)$ takes the general form given by $V = p(t)x^2(t)$ then $\frac{\partial V}{\partial t} = 0$ (since

this expression does not explicitly depend on the variable t) and $\frac{\partial V}{\partial x} = 2p(t)x(t)$, so we get for the HJB equation the constraint

$$0 = (x(t) - x_d)^2 + u(t)^2 + 2p(t)x(t)(ax(t)^2 + bu(t)) \,.$$

This is to be coupled with the dynamic constraint

$$\dot{x}(t) = ax(t)^n + bu(t) \quad \text{and} \quad x(0) = 0.1 \,.$$

Problem 3.5.1 (rotations of a nonspinning satellite)

This is a minimum time problem with a linear dynamic system. Following the section on minimum time problems we first redefine the problem so that the control we apply is bounded by 1 i.e $\tilde{u} \leq 1$. To do that, we assume that the original control u, is bounded as $|u| \leq u_{\max}$ (the book does not have the absolute value). Thus we consider the problem

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + u_{\max} \begin{bmatrix} 0 \\ 1 \end{bmatrix} \tilde{u} \,.$$

The cost function we desire to minimize is $J = \int_{t_0}^{t_f} 1 dt$. With this setup introduce a Hamiltonian given by

$$\mathcal{H} = 1 + \lambda^T (Fx^* + Gu^*) \,,$$

and the minimum principle gives

$$1 + \lambda^{*T}[Fx^* + Gu^*] \leq 1 + \lambda^{*T}[Fx^* + Gu] \,,$$

that must be satisifed by the optimal u^*. The above can be written as

$$\lambda^{*T} G u^* \leq \lambda^{*T} G u \,.$$

The minimum control picks u^* to make $\lambda^{*T} G u$ must be negative.

$$u^*(t) = \begin{cases} +1 & \lambda^{*T} < 0 \\ -1 & \lambda^{*T} > 0 \end{cases}$$

is how you specify the optimal control.

We will take the functional we want to minimize to be given by

$$J = \int_{t_0}^{t_f} 1 dt = t_f - t_0 \,,$$

so $\mathcal{L} \equiv 1$ (a constant value). Our dynamics of $\theta(t)$ are given by the differential equation $\ddot{\theta} = u$ with initial conditions $\theta(0) = 1$ and $\dot{\theta}(0) = 0$. We can write this in system form by introducing the variables $x_1 = \theta$ and $x_2 = \dot{\theta}$ where we see that

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= u \,. \end{aligned}$$

To enforce the terminal constraints that $\theta(t_f) = \dot{\theta}(t_f) = 0$ we will introduce a *vector* end point constraint

$$\psi[x(t_f), t_f] = \begin{bmatrix} x_1(t_f) \\ x_2(t_f) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}.$$

For the system above we find our functions F and G given by

$$F = \frac{\partial f}{\partial x} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \quad \text{and} \quad G = \begin{bmatrix} 0 \\ 1 \end{bmatrix}.$$

Note that with the above expression for G we have $GG^T = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}$ and thus $\mathbf{x}GG^T\mathbf{y} = x_2y_2$ or the product of the second elements in the vectors $\mathbf{x}$ and $\mathbf{y}$. Following the book on vector terminal constraints we have the associated r adjoint vectors λ_i which must solve

$$\dot{\lambda}_i = -F^T\lambda_i \quad \text{with} \quad \lambda(t_f) = \frac{\partial \psi_i}{\partial \mathbf{x}} \quad \text{for} \quad i = 1, 2, \cdots r\,.$$

In this problem, the differential equations and initial conditions for $i = 1$ are

$$\begin{aligned} \frac{d}{dt}\begin{bmatrix} \lambda_{11}(t) \\ \lambda_{12}(t) \end{bmatrix} &= -\begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix}\begin{bmatrix} \lambda_{11}(t) \\ \lambda_{12}(t) \end{bmatrix} \quad \text{with} \\ \begin{bmatrix} \lambda_{11}(t_f) \\ \lambda_{12}(t_f) \end{bmatrix} &= \frac{\partial \psi_1}{\partial \mathbf{x}} = \frac{\partial}{\partial \mathbf{x}} x_1(t_f) = \begin{bmatrix} 1 \\ 0 \end{bmatrix}. \end{aligned}$$

To solve for $\lambda_1(t)$ we see that the first equation is $\frac{d}{dt}\lambda_{11}(t) = 0$ with $\lambda_{11}(t_f) = 1$, thus $\lambda_{11}(t) = 1$. The second equation is $\frac{d}{dt}\lambda_{12}(t) = -\lambda_{11}(t) = -1$ with $\lambda_{12}(t_f) = 0$ thus $\lambda_{12}(t) = -t + t_f$. As a vector we then have

$$\lambda_1(t) = \begin{bmatrix} 1 \\ t_f - t \end{bmatrix}.$$

For this problem, the differential equations and initial conditions for $i = 2$ are

$$\begin{aligned} \frac{d}{dt}\begin{bmatrix} \lambda_{21}(t) \\ \lambda_{22}(t) \end{bmatrix} &= -\begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix}\begin{bmatrix} \lambda_{21}(t) \\ \lambda_{22}(t) \end{bmatrix} \quad \text{with} \\ \begin{bmatrix} \lambda_{21}(t_f) \\ \lambda_{22}(t_f) \end{bmatrix} &= \frac{\partial \psi_2}{\partial \mathbf{x}} = \frac{\partial}{\partial \mathbf{x}} x_2(t_f) = \begin{bmatrix} 0 \\ 1 \end{bmatrix}. \end{aligned}$$

To solve for $\lambda_2(t)$ we see that the first equation is $\frac{d}{dt}\lambda_{21}(t) = 0$ with $\lambda_{21}(t_f) = 0$, thus $\lambda_{21}(t) = 0$. The second equation is $\frac{d}{dt}\lambda_{22}(t) = -\lambda_{11}(t) = 0$ with $\lambda_{12}(t_f) = 1$ thus $\lambda_{22}(t) = 1$. As a vector we then have

$$\lambda_2(t) = \begin{bmatrix} 0 \\ 1 \end{bmatrix}.$$

Now we need to solve for μ in $B\mu = -A$ with

$$\begin{aligned} b_{ir} &= \int_{t_0}^{t_f} \lambda_i^T GG^T \lambda_r dt \\ a_i &= \int_{t_0}^{t_f} \lambda_i^T G \left[\frac{\partial \mathcal{L}}{\partial u}^T + G^T\lambda_0\right] dt\,. \end{aligned}$$

For the functions λ_i just computed we have

$$
\begin{aligned}
b_{11} &= \int_{t_0}^{t_f} \lambda_{12}(t)^2 dt = \int_{t_0}^{t_f} (t-t_f)^2 dt = \left. \frac{(t-t_f)^3}{3} \right|_{t_0}^{t_f} = \frac{(t_f-t_0)^3}{3} \\
b_{12} &= \int_{t_0}^{t_f} \lambda_{12}(t)\lambda_{22}(t) dt = \int_{t_0}^{t_f} (t_f-t) dt = -\left. \frac{(t-t_f)^2}{2} \right|_{t_0}^{t_f} = \frac{(t_f-t_0)^2}{2} \\
b_{21} &= \int_{t_0}^{t_f} \lambda_{22}(t)\lambda_{12}(t) dt = b_{12} = \frac{(t_f-t_0)^2}{2} \\
b_{22} &= \int_{t_0}^{t_f} \lambda_{22}(t)^2 dt = \int_{t_0}^{t_f} dt = t_f - t_0 \,.
\end{aligned}
$$

We have to find λ_0 which satisfies

$$
\frac{d}{dt} \begin{bmatrix} \lambda_{01} \\ \lambda_{02} \end{bmatrix} = -F^T \lambda_0 - \frac{\partial \mathcal{L}}{\partial x}^T \,.
$$

with

$$
\lambda_0(t_f) = \left. \frac{\partial \phi}{\partial \mathbf{x}}^T \right|_{t=t_f} \,.
$$

Once one has μ we solve

$$
\frac{\partial \mathcal{L}}{\partial u} + \lambda_0^T G + (\mu_1 \lambda_1^T + \cdots + \lambda_r \lambda_r^T) G = 0 \,,
$$

for u.

The dynamic model

$$
\begin{aligned}
\dot{x}_1 &= x_1(1-x_2) \\
\dot{x}_2 &= x_2(1-x_1) - 0.5(e^{-a(t-t_i)} + b) x_2 u \,.
\end{aligned}
$$

Now since u kills the predictor x_2 and not the prey x_1.

Problem 3.5.3 (single room temperature control: hard end constraints)

For this problem we assume we have the *hard* constraint on x_1 at t_f of $x_1(t_f) = x_d$ as discussed on Page 61 of these notes. We will assume that the cost function that penalizes power consumption is given by $J = \int_{t_0}^{t_f} u^2 dt$ (the book does not have the square on the function $u(t)$) with $\psi = x_1(t_f) - x_d$. For the system

$$
\dot{x}_1 = -\frac{1}{T}(x_1 - x_{\text{air}}) + k_1 u \,, \tag{104}
$$

we have $F = \frac{\partial f}{\partial x} = -\frac{1}{T}$ and $G = \frac{\partial f}{\partial u} = k_1$. Following the discussion on Page 61 we need to solve Equation 81 or

$$\dot{\lambda}_1(t) = -\left(\frac{\partial \mathcal{H}_1}{\partial x}\right)^T = -F^T(t)\lambda_1(t) = \frac{1}{T}\lambda_1(t) \quad \text{with}$$
$$\lambda_1(t_f) = \left(\frac{\partial \psi}{\partial x}\right)^T\Bigg|_{t=t_f} = 1\,,$$

We solve for $\lambda_1(t)$ and find $\lambda_1(t) = e^{\frac{t-t_f}{T}}$. Next we need to consider Equation 83 for $\lambda_0(t)$ or

$$\dot{\lambda}_0(t) = -\left(\frac{\partial \mathcal{H}_0}{\partial x}\right)^T = -F^T(t)\lambda_0(t) - \left(\frac{\partial \mathcal{L}}{\partial x}\right)^T = \frac{1}{T}\lambda_0(t) \quad \text{with}$$
$$\lambda_0(t_f) = \left(\frac{\partial \phi}{\partial x}\right)^T\Bigg|_{t=t_f} = 0\,.$$

The only way we can have a function $\lambda_0(t)$ that satisfies these conditions is if $\lambda_0 \equiv 0$. To determine $u(t)$ we need to use Equation 85 where we get

$$\frac{\partial \mathcal{H}_c}{\partial u} = \frac{\partial \mathcal{L}}{\partial u} + \lambda_0^T G + \mu \lambda_1^T G = 0\,,$$

or

$$2u(t) + \mu k_1 \lambda_1(t) = 0 \quad \text{so} \quad u(t) = -\frac{\mu k_1}{2}\lambda_1(t) = -\frac{\mu k_1}{2} e^{\frac{t-t_f}{T}}\,.$$

Now that we know explicit formulas for $\lambda_0(t)$ and $\lambda_1(t)$ we can evaluate a and b using Equation 87 and 88. We find

$$\begin{aligned} a &= \int_{t_0}^{t_f} \lambda_1^T G \left[\left(\frac{\partial \mathcal{L}}{\partial u}\right)^T + G^T \lambda_0\right] dt \\ &= 2\int_{t_0}^{t_f} \lambda_1^T G u dt = 2k_1 \int_{t_0}^{t_f} \lambda_1 u dt = -\frac{k_1^2 \mu T}{2}\left(1 - e^{\frac{2(t_0-t_f)}{T}}\right) \\ b &= \int_{t_0}^{t_f} \lambda_1^T G G^T \lambda_1 dt = k_1^2 \int_{t_0}^{t_f} \lambda_1^2 dt = k_1^2 \frac{T}{2}\left(1 - e^{\frac{2(t_0-t_f)}{T}}\right)\,. \end{aligned}$$

Note that $a \neq 0$ as is required. Now that we know the functional form for $u(t)$ we can put this expression into Equation 104 and then solve for $x_1(t)$. When we do this we find (see the Mathematica code `sect_5_prob_3.nb`)

$$x_1(t) = x_{\text{air}} + (x_0 - x_{\text{air}})e^{-\frac{t-t_0}{T}} - \frac{\mu k_1^2 T}{4}\left(e^{-\frac{t_f - t}{T}} + e^{-\frac{t+t_f-2t_0}{T}}\right)\,.$$

Then since our pointwise constraint is $\psi = x_1(t_f) - x_d = 0$ we can evaluate the above at t_f where we find

$$x_1(t_f) = x_{\text{air}} + (x_0 - x_{\text{air}})e^{-\frac{t_f - t_0}{T}} - \frac{\mu k_1^2 T}{4}\left(1 + e^{-\frac{2(t_0-t_f)}{T}}\right)\,,$$

as this expression must equal x_d we can use the above to compute the value of μ. Using the numbers given in the book in the Matlab script `sect_5_prob_3.m` we find $\mu = -24.9382$.

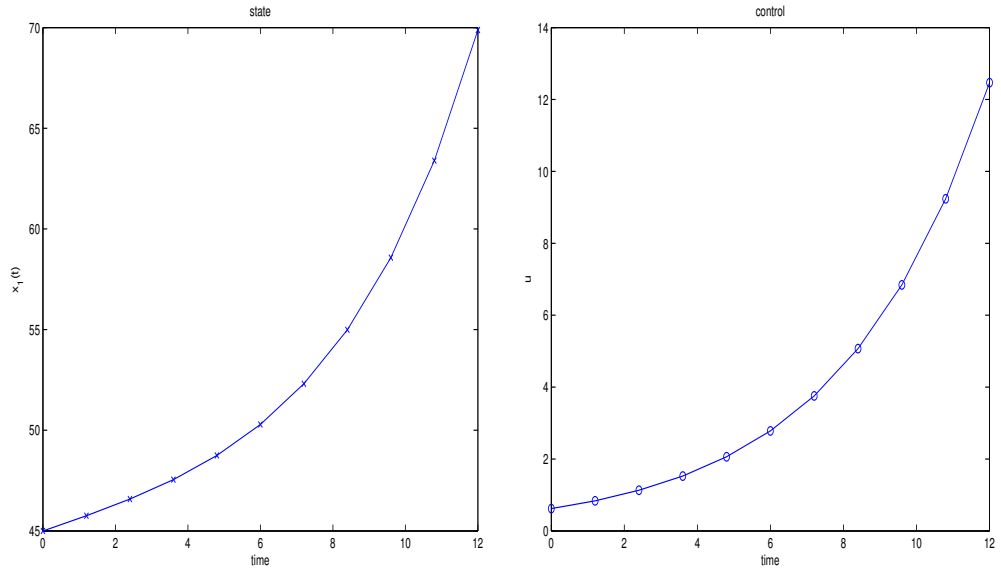

Figure 13: **Left:** The state estimate $x_1(t)$ for Chapter 3 Section 5 Problem 3 (single room temperature control: hard end constraints). Note that $x_1(t)$ goes from the initial temperature of 45 to the final value x_{1f}^* which is exactly equal to (as it should be) to the desired temperature $x_d = 70$. **Right:** The control $u(t)$ for the same problem. Note that we steadily apply more and more heat as time progress to heat the room.

This script also plots the optimal control $u(t)$ and the corresponding state $x_1(t)$ in Figure 13.

Since $x_1(0) = 45$ and $x_1(t_f) = 70$ we know that $u > 0$ since we must add heat to increase the temperature. Thus if we want to original optimality condition

$$J = \int_{t_0}^{t_f} |u| dt = \int_{t_0}^{t_f} u dt \,.$$

The Hamilton for this problem is

$$\mathcal{H} = \mathcal{L} + \lambda^T f = u + \lambda(-\frac{1}{T}(x_1 - x_{\text{air}}) + k_1 u)$$

since $\mathcal{H}$ is *linear* in u the control dose not appear in $\mathcal{H}_u$ thus convexity cannot be established. Thus we must use the minimization principle to find u. If $\lambda > 0$ we pick $u = 0$. If $\lambda < 0$ we pick $u = u_{\max}$.

Problem 3.5.4 (single room temperature control: soft end constraint)

This is a direct application of the Euler-Lagrange equations from the earlier section with the "integrated" cost of $\mathcal{L}[x(t), u(t), t] = c_2 u^2$ and with an "end point" cost of $\phi = c_1(x_{1f} - x_d)^2$. We thus need to solve for $\lambda(t)$ in

$$\dot{\lambda}(t) = -F^T(t)\lambda(t) - \left(\frac{\partial \mathcal{L}}{\partial x}\right)^T \quad \text{with} \quad \lambda(t_f) = \left.\frac{\partial \phi}{\partial x}\right|_{t=t_f} .$$

For this problem where the state dynamics are given by

$$\dot{x}_1 = -\frac{1}{T}(x_1 - x_{\text{air}}) + k_1 u\,,$$

we have $F = -\frac{1}{T}$, $G = k_1$, $\frac{\partial \mathcal{L}}{\partial x} = 0$, $\frac{\partial \mathcal{L}}{\partial u} = 2c_2 u$. Using these the above equations becomes

$$\dot{\lambda}(t) = \frac{1}{T}\lambda(t) \quad \text{and} \quad \lambda(t_f) = 2c_1(x_{1f} - x_d) \tag{105}$$

$$\dot{x}_1(t) = -\frac{1}{T}(x_1 - x_{\text{air}}) + k_1 u \quad \text{and} \quad x_1(t_0) = x_{10} \tag{106}$$

$$\frac{\partial \mathcal{L}}{\partial u}^T + G^T\lambda = 0 \quad \text{or} \quad 2c_2 u + k_1\lambda = 0\,. \tag{107}$$

The solution to Equation 105 is $\lambda(t) = 2c_1(x_{1f} - x_d)e^{\frac{t-t_f}{T}}$. The solution to Equation 107 for $u(t)$ in terms of $\lambda(t)$ is

$$u(t) = -\frac{k_1}{2c_2}\lambda(t) = -\frac{k_1 c_1}{c_2}(x_{1f} - x_d)e^{\frac{t-t_f}{T}}\,.$$

The dynamic equation for $x_1(t)$ or Equation 105 with this control is then

$$\dot{x}_1(t) = -\frac{1}{T}(x_1 - x_{\text{air}}) - \frac{k_1^2 c_1}{c_2}(x_{1f} - x_d)e^{\frac{t-t_f}{T}}\,.$$

Which gives for $x_1(t)$ the following (see the Mathematica code `sect_5_prob_4.nb`)

$$x_1(t) = x_{\text{air}} + (x_0 - x_{\text{air}})e^{-\frac{t+t_0}{T}} - \frac{k_1^2 T(x_{1f} - x_d)c_1}{2c_2}\left(e^{\frac{t-t_f}{T}} + e^{\frac{2t_0-t-t_f}{T}}\right)\,.$$

When we put $t = t_f$ in the above we find we then get a single equation for x_{1f} given by

$$x_{1f} = x_{\text{air}} + (x_0 - x_{\text{air}})e^{-\frac{t_0+t_f}{T}} - \frac{k_1^2 T(x_{1f} - x_d)c_1}{2c_2}\left(1 + e^{\frac{2(t_0-t_f)}{T}}\right)\,. \tag{108}$$

which we can solve for the unknown state end point x_{1f} in terms of numbers from the problem like k_2, c_1, c_2 etc. We do this in the MATLAB script `sect_5_prob_4.m` where we find the solution for x_{1f} above given by $x_{1f}^* = 67.7324$ where the star denotes that we have the optimal value. In the same MATLAB script we plot $x_1(t)$ and $u(t)$ for $t_0 \le t \le t_f$ and get the results shown in Figure 14.

Problem 3.5.5 (single room temperature control: a penalty when $x_1 \neq x_d$)

If we change the Lagrangian from the previous problem to now be $\mathcal{L} = c_1(x_1(t) - x_d)^2 + c_2 u^2$, we now have the x and u partial derivatives given by

$$\frac{\partial \mathcal{L}}{\partial x} = 2c_1(x_1(t) - x_d)$$
$$\frac{\partial \mathcal{L}}{\partial u} = 2c_2 u\,.$$

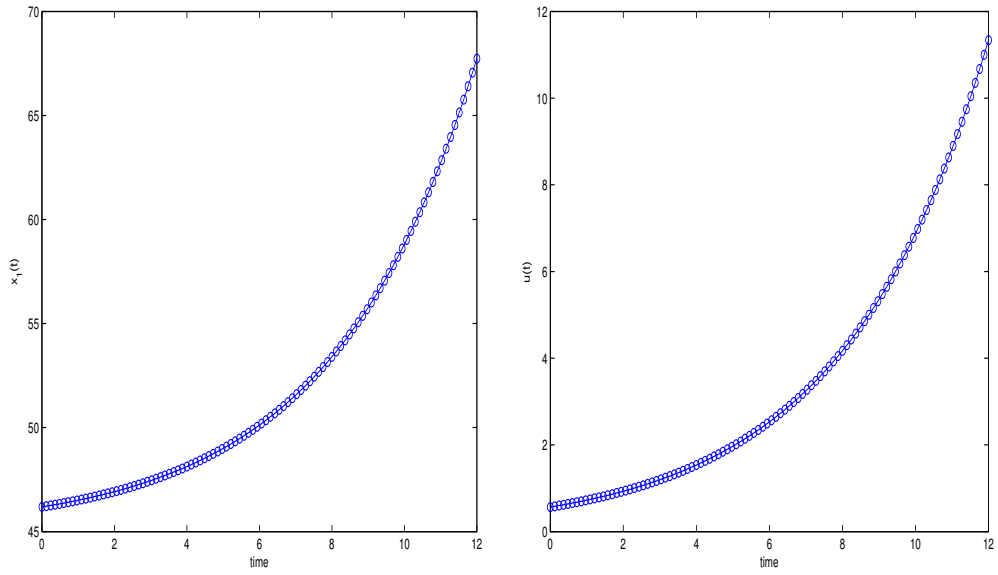

Figure 14: **Left:** The state estimate $x_1(t)$ for Chapter 3 Section 5 Problem 4 (single room temperature control: soft end constraint). Here we imposed soft end point constraints and as such note that $x_1(t)$ goes from the initial room temperature of 45 to the final value x_{1f}^* which is close but not exactly equal to the desired temperature $x_d = 70$ (as it should be). How close x_{1f} gets to x_d depends on the relative size of c_1 and c_2. **Right:** The control $u(t)$ for the same problem. Note that we steady apply more and more heat as time progress.

Keeping the "end point" cost of $\phi = c_1(x_{1f} - x_d)^2$ as was present in the previous problem, the Euler-Lagrange equations we must solve are given by

$$\dot{\lambda} = \frac{1}{T}\lambda(t) - 2c_1(x_1(t) - x_d) \quad \text{with} \quad \lambda(t_f) = 2c_1(x_{1f} - x_d)$$
$$\dot{x}_1 = -\frac{1}{T}(x_1(t) - x_{\text{air}}) + k_1 u(t) \quad \text{with} \quad x_1(t_0) = x_{10}$$
$$2c_2 u + k_1 \lambda = 0 \quad \text{or} \quad u = -\frac{k_1}{2c_2}\lambda\,.$$

We can put this last equation into the equation for x_1 to get the coupled system of equations which we must solve

$$\dot{x}_1 = -\frac{1}{T}(x_1(t) - x_{\text{air}}) - \frac{k_1^2}{2c_2}\lambda(t) \quad \text{with} \quad x_1(t_0) = x_{10}$$
$$\dot{\lambda} = \frac{1}{T}\lambda(t) - 2c_1(x_1(t) - x_d) \quad \text{with} \quad \lambda(t_f) = 2c_1(x_{1f} - x_d)\,.$$

Since these equations are *linear* there are analytic solutions to the above mixed two-point boundary value problem, see Problem 3.5.6 on Page 82 where that method is discussed and used. For this problem however, we will try to solve it numerically using the "neighboring extremal method". This is a type of shooting method that can solve two-point nonlinear boundary value problems. In it we will guess a value for the *initial condition* for the function $\lambda(t)$ and then solve the two ordinary differential equations forward in time for $t_0 \le t \le t_f$. This gives us a value for the function $\lambda(t)$ at the time t_f. The correct value to assign to the

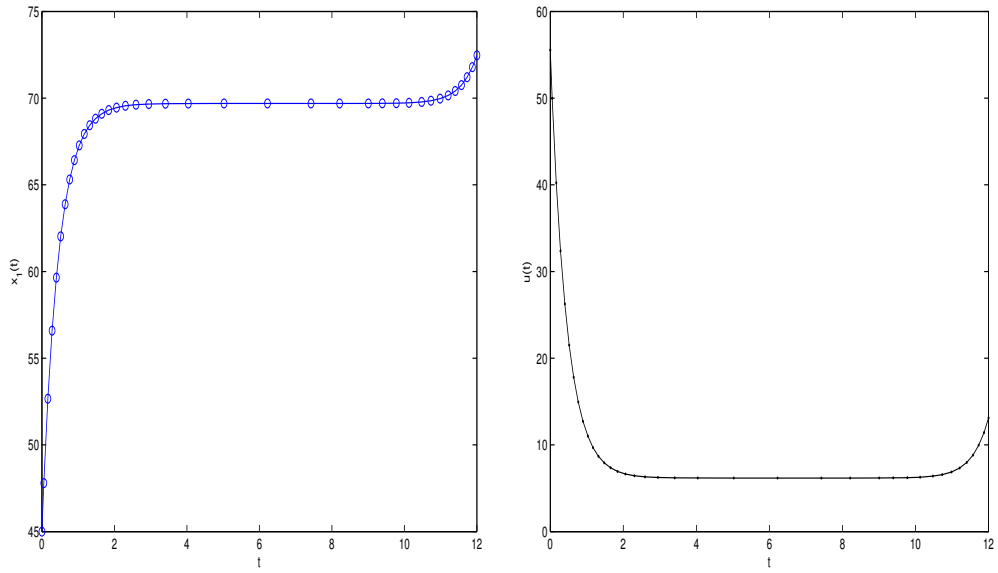

Figure 15: **Left:** The state estimate $x_1(t)$ for Chapter 3 Section 5 Problem 5 single room temperature control: a penalty when $x_1 \neq x_d$. Note that due to the term in the Lagrangian of $c_1 \int_{t_0}^{t_f}(x_1(t) - x_d)^2 dt$ the function $x_1(t)$ very quickly moves to a level close to x_d and then does not change for much of the remaining time. The system This penalty is an effective way to make the system reach its desired state quickly. **Right:** The control $u(t)$ for the same problem. The system effectively inputs a heat delta function and then maintains a constant heat supply.

initial condition for $\lambda(t_0) = \lambda_0$ is the one that matches the known end point condition at $t = t_f$. This means we want $\lambda(t_f) = 2c_1(x_{1f} - x_d)$. Thus we will try to pick different values of λ_0 such that the best one selected minimizes a measure of the error in the final value of $\lambda(t)$ or

$$\tilde{J} = \frac{1}{2}[\lambda(t_f) - 2c_1(x(t_f) - x_d)]^2 .$$

This procedure is implemented in the MATLAB files:

- `sect_5_prob_5.m`
- `sect_5_prob_5_J_min_fn.m`
- `sect_5_prob_5_ode_fn.m`.

When we run the main script function we calculate that $\lambda_0 \approx -111.1111$ and then plots of $x_1(t)$ and $u(t)$ are given in Figure 15.

Problem 3.5.6 (heating a two-room building)

From what I understand the problem suggested is for the system dynamics given by

$$\dot{x}_1 = -\frac{1}{T_1}(x_1 - x_{\text{air}}) + k_1(x_2 - x_1) + k_2 u$$
$$\dot{x}_2 = -\frac{1}{T_2}(x_2 - x_{\text{air}}) + k_1(x_1 - x_2)\,,$$

with $x_1(t_0) = x_{\text{air}}$, $x_2(t_0) = x_{\text{air}}$ (to match an earlier problem), a Lagrangian given by

$$\mathcal{L} = c_1(x_1(t) - x_d)^2 + c_2 u^2\,, \tag{109}$$

and the desired final condition

$$\phi = c_1(x_{1f} - x_d)^2\,.$$

I'm not sure if there is a typo in the book at this point. A motivation for this comment comes from the fact that the suggested Lagrangian $\mathcal{L}$ above does not depend on the state $x_2(t)$ in anyway. If the control u is to modify the temperature of the second room $x_2(t)$ optimally I would expect to see the function $x_2(t)$ in the Lagrangian. If it really does not appear there than it seems we could just effectively "drop" the dynamic equation for $x_2(t)$. If we are to really control the temperature in the second room, I can see two alternative (perhaps better) Lagriangians to consider.

$$\mathcal{L} = c_1(x_2(t) - x_d)^2 + c_2 u^2$$
$$\mathcal{L} = c_1\left((x_1(t) - x_d)^2 + (x_2(t) - x_d)^2\right) + c_2 u^2\,,$$

We would use the first form if we had to heat the second room to the desired temperature x_d "though" the first room, and the second Lagrangian if we wanted both rooms at the desired temperature x_d. The same comments hold for the end point constraint ϕ. Since I was not able to get the given Lagrangian to work "as is" I will try the second Lagrangian formulation above with a similar end point condition

$$\phi = c_1(x_{1f} - x_d)^2 + c_1(x_{2f} - x_d)^2\,.$$

In this case we have the partial derivatives of $\mathcal{L}$ given by

$$\frac{\partial \mathcal{L}}{\partial x_1} = 2c_1(x_1(t) - x_d)$$
$$\frac{\partial \mathcal{L}}{\partial x_2} = 2c_1(x_2(t) - x_d)$$
$$\frac{\partial \mathcal{L}}{\partial u} = 2c_2 u\,.$$

The system dynamics $\dot{x} = f[x(t), u(t), t]$ give expressions for the Jacobian matrices F and G of

$$F = \frac{\partial f}{\partial x} = \begin{bmatrix} -\frac{1}{T_1} - k_1 & k_1 \\ k_1 & -\frac{1}{T_2} - k_1 \end{bmatrix}$$
$$G = \frac{\partial f}{\partial u} = \begin{bmatrix} k_2 \\ 0 \end{bmatrix}.$$

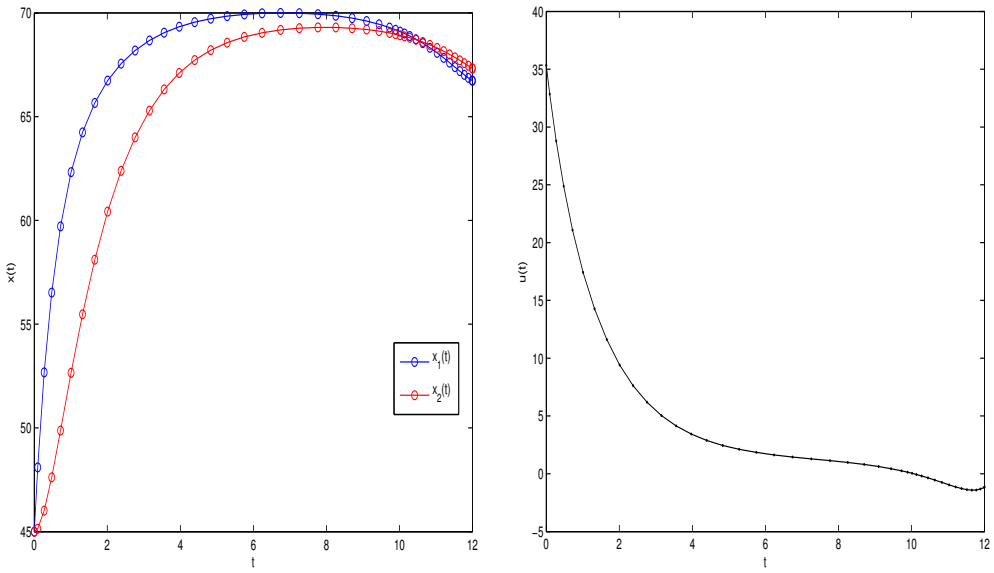

Figure 16: **Left:** The state estimates for $x_1(t)$ and $x_2(t)$ for Chapter 3 Section 5 Problem 6 heating a two-room building. **Right:** The control $u(t)$ for the same problem.

From these the Euler-Lagrange equations we must solve are then given by

$$\frac{d}{dt}\begin{bmatrix} \lambda_1(t) \\ \lambda_2(t) \end{bmatrix} = -\begin{bmatrix} -\frac{1}{T_1} - k_1 & k_1 \\ k_1 & -\frac{1}{T_2} - k_1 \end{bmatrix}\begin{bmatrix} \lambda_1 \\ \lambda_2 \end{bmatrix} - \begin{bmatrix} 2c_1(x_1 - x_d) \\ 2c_1(x_2 - x_d) \end{bmatrix}.$$

$$\frac{\partial \mathcal{L}}{\partial u}^T + G^T\lambda = 0 \quad \text{or} \quad 2c_2u + \begin{bmatrix} k_2 & 0 \end{bmatrix}\begin{bmatrix} \lambda_1 \\ \lambda_2 \end{bmatrix} = 0 \quad \text{or} \quad 2c_2u + k_2\lambda_1 = 0\,.$$

The final conditions on λ_1 and λ_2 are

$$\lambda_1(t_f) = 2c_1(x_{1f} - x_d) \quad \text{and} \quad \lambda_2(t_f) = 2c_1(x_{2f} - x_d)\,,$$

due to the chosen ϕ function.

As in the previous problem we can solve for $u(t)$ in terms of $\lambda_1(t)$ and then put this expression into the differential equation for x_1. We then have a coupled system of ordinary differential equations for the unknown functions $x_1(t)$, $x_2(t)$, $\lambda_1(t)$, and $\lambda_2(t)$. Since we don't know the initial conditions of the two functions λ_1 and λ_2 (we do however know their final conditions at $t = t_f$) we will find initial conditions for these two functions that minimize the error between their computed final conditions and the known desired final conditions as well as possible. To do that we will find initial conditions that minimize the functional $\tilde{J} = \frac{1}{2}(\lambda_1(t_f)^2 + \lambda_2(t_f)^2)$ numerically. We do this in the MATLAB files `sect_5_prob_6.m`, `sect_5_prob_6_J_min_fn.m`, and `sect_5_prob_6_ode_fn.m`. When we run the main script function we calculate that $\lambda_1(t_0) \approx -70.654$ and $\lambda_2(t_0) \approx -77.0126$ then plots of $x_1(t)$, $x_2(t)$ and $u(t)$ are given in Figure 16. If the mixed boundary value problem is linear then it can be solved exactly by the principle of superposition. Taken from the book "Optimal Control Theory: An Introduction" by Donald E. Kirk the procedure for this is as follows, we consider

$$\dot{x}(t) = a_{11}(t)x(t) + a_{12}(t)p(t) + e_1(t)$$
$$\dot{p}(t) = a_{21}(t)x(t) + a_{22}(t)p(t) + e_2(t)\,.$$

We assume that this linear problem has split boundary conditions where

$$x(t_0) = x_0 \quad \text{and} \quad p(t_f) = p_f \,,$$

We first solve the homogeneous problem obtained by dropping $e_1(t)$ and $e_2(t)$ from the above system to get $x^H(t)$ and $p^H(t)$ for $t \in [t_0, t_f]$. With arbitrary initial conditions say $x^H(t_0) = 0$ and $p^H(t) = 1$. We next define the particular solution to the above full system with $x^p(t)$ and $p^p(t)$ with with the initial conditions

$$x^p(t_0) = x_0 \quad \text{and} \quad p^p(t_f) = 0 \,.$$

Then since the equations are linear we have

$$\begin{aligned} x(t) &= c_1 x^H(t) + p^P(t) \\ p(t) &= c_1 x^H(t) + p^P(t) \,. \end{aligned} \tag{110}$$

is a solution for all c_1. To make this match $p(t_f) = p_f$ or

$$c_1 p^H(t_f) + p^p(t_f) = p_f \,.$$

So we pick

$$c_1 = \frac{p_f - p^p(t_f)}{p^H(t_f)} \,.$$

we have the required solution.

Optimal State Estimation

Notes on the text

Notes on least squares estimate of constant vectors

For the objective function $J(z)$ given by

$$J(z) = \frac{1}{2}(z - H\hat{x})^T(z - H\hat{x}) = \frac{1}{2}(z^T z - z^T H\hat{x} - \hat{x}^T H^T z + \hat{x}^T H^T H\hat{x})\,, \tag{111}$$

with $\hat{x} = H^L z$ where the left pseudoinverse, H^L, given by Equation 10 or

$$H^L = (H^T H)^{-1} H^T\,. \tag{112}$$

Then note that the transpose of this matrix is

$$(H^L)^T = H(H^T H)^{-1}\,. \tag{113}$$

Using this we can write $J(z)$ in terms of z only as

$$\begin{aligned} J(z) &= \frac{1}{2}(z^T z - z^T H(H^T H)^{-1}H^T z - z^T H(H^T H)^{-1}H^T z + z^T H(H^T H)^{-1}H^T H(H^T H)^{-1}H^T z \\ &= \frac{1}{2}(z^T z - z^T H(H^T H)^{-1}H^T z)\,. \end{aligned} \tag{114}$$

Since our measurement z is related to the state x via $z = Hx + n$ we can replace z with an expression in x in $J(z)$ with an expression in terms of x. We find that the two terms in the above expression for $J(z)$ transform as

$$\begin{aligned} z^T z &= (Hx + n)^T(Hx + n) \\ &= x^T H^T Hx + x^T H^T n + n^T Hx + n^T n \quad \text{and} \\ z^T H(H^T H)^{-1}H^T z &= (x^T H^T + n^T)H(H^T H)^{-1}H^T(Hx + n) \\ &= (x^T H^T + n^T)(H(H^T H)^{-1}H^T Hx + H(H^T H)^{-1}H^T n) \\ &= x^T H^T Hx + x^T H^T n + n^T Hx + n^T H(H^T H)^{-1}H^T n\,. \end{aligned}$$

Then the difference $z^T z - z^T H(H^T H)^{-1} z$ needed to evaluate $J(z)$ is given by

$$J(z) = \frac{1}{2}\left[n^T n - n^T H(H^T H)^{-1}H^T n\right] = \frac{1}{2}n^T[I_{k_1} - H(H^T H)^{-1}H^T]n\,. \tag{115}$$

Another expression for J can be obtained by taking $z = Hx + n$ and $\hat{x} = H^L z$ and putting them into $J = \frac{1}{2}(x - \hat{x})^T(x - \hat{x})$. To evaluate this we consider the difference $x - \hat{x}$

$$\begin{aligned} x - \hat{x} &= x - H^L z = x - (H^T H)^{-1}H^T(Hx + n) = x - x - (H^T H)^{-1}H^T n \\ &= -(H^T H)^{-1}H^T n\,, \end{aligned}$$

so J then becomes

$$J(x) = \frac{1}{2} n^T H (H^T H)^{-1} (H^T H)^{-1} H^T n \,. \tag{116}$$

To evaluate these two we consider a simple example where we are evaluating a random constant x from k_1 noisy measurements $z_i = x + n_i$. If we take $k_1 = 2$ as a vector system our two measurements are

$$\begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix} x + \begin{bmatrix} n_1 \\ n_2 \end{bmatrix} .$$

To make this match the matrix notation $z = Hx + n$ we have the H matrix given by $H = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$. Then $H^T H = 2$ and $H^L = (H^T H)^{-1} H^T = \frac{1}{2} \begin{bmatrix} 1 & 1 \end{bmatrix}$. The best estimate of x is then

$$x = H^L z = \frac{1}{2}(z_1 + z_2) \,.$$

To evaluate $J(z)$ using Equation 115 we need

$$H(H^T H)^{-1} H^T = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \frac{1}{2} \begin{bmatrix} 1 & 1 \end{bmatrix} \quad \text{so}$$

$$I_2 - H(H^T H)^{-1} H^T = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} - \frac{1}{2} \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 1 & -1 \\ 1 & -1 \end{bmatrix} ,$$

so that

$$\begin{aligned} J(z) &= \frac{1}{2} \begin{bmatrix} n_1 & n_2 \end{bmatrix} \frac{1}{2} \begin{bmatrix} 1 & -1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} n_1 \\ n_2 \end{bmatrix} = \frac{1}{4} \begin{bmatrix} n_1 & n_2 \end{bmatrix} \begin{bmatrix} n_1 - n_2 \\ n_1 + n_2 \end{bmatrix} \\ &= \frac{1}{4}(n_1^2 - n_1 n_2 - n_1 n_2 + n_2^2) = \frac{1}{4}(n_1 - n_2)^2 \,. \end{aligned}$$

To evaluate $J(x)$ using Equation 116 since $(H^T H)^{-1} H^T = \frac{1}{2} \begin{bmatrix} 1 & 1 \end{bmatrix}$ its matrix transpose is given by $H(H^T H)^{-1} = \frac{1}{2} \begin{bmatrix} 1 \\ 1 \end{bmatrix}$ and the matrix product needed is

$$H(H^T H)^{-1} (H^T H)^{-1} H^T = \frac{1}{4} \begin{bmatrix} 1 \\ 1 \end{bmatrix} \begin{bmatrix} 1 & 1 \end{bmatrix} = \frac{1}{4} \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} .$$

Using this we find $J(x)$ given by

$$\begin{aligned} J(x) &= \frac{1}{8} \begin{bmatrix} n_1 & n_2 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} n_1 & n_2 \end{bmatrix} = \frac{1}{8} \begin{bmatrix} n_1 & n_2 \end{bmatrix} \begin{bmatrix} n_1 + n_2 \\ n_1 + n_2 \end{bmatrix} \\ &= \frac{1}{8}(n_1^2 + n_1 n_2 + n_1 n_2 + n_2^2) = \frac{1}{8}(n_1 + n_2)^2 \,. \end{aligned}$$

We can generalize the above expressions where $k_1 = 2$ to that with a general $k_1 > 0$. In this case H is a column vector of all ones with k_1 elements. The matrix $H^T H = k_1$ and HH^T is a $k_1 \times k_1$ matrix of all ones. Then the two parts of $J(z)$ given by Equation 115 are

$$n^T n = \sum_{k=1}^{k_1} n_k^2 \quad \text{and} \quad H^T n = \sum_{k=1}^{k_1} n_k \quad \text{so}$$

$$n^T H H^T n = \left(\sum_{k=1}^{k_1} n_k \right)^2 = \sum_{i=1}^{k_1} n_i \left(\sum_{j=1}^{k_1} n_j \right) .$$

Thus we have

$$J(z) = \frac{1}{2}\sum_{k=1}^{k_1} n_k^2 - \frac{1}{2k_1}\sum_{i=1}^{k_1} n_i \left(\sum_{j=1}^{k_1} n_j\right) .$$

In the same way we find for $J(x)$ using Equation 116

$$J(x) = \frac{1}{2k_1^2} n^T H H^T n = \frac{1}{2k_1^2}\left(\sum_{k=1}^{k_1} n_k\right)^2 .$$

To introduce **weighted least squares (WLS)** we define the *normalized* error in our predicted measurement $\hat{y}$ relative to the actual measurement or ϵ_z' as

$$\epsilon_z' = N^{-1}\epsilon_z = N^{-1}(z - \hat{y}) = N^{-1}(z - H\hat{x}) .$$

With this we have $J'(z)$ given by

$$\begin{aligned} J'(z) &= \frac{1}{2}\epsilon_z'^T \epsilon_z = \frac{1}{2}(z - H\hat{x})^T N^{-T} N^{-1}(z - H\hat{x}) \\ &= \frac{1}{2}\left[z^T S^{-1} z - z^T S^{-1} H\hat{x} - \hat{x}^T H^T S^{-1} z + \hat{x}^T H^T S^{-1} H\hat{x}\right] . \end{aligned} \tag{117}$$

We define S^{-1} as the matrix in the inner product or $S^{-1} \equiv N^{-T}N^{-1}$ so $S = NN^T$. To pick the optimal $\hat{x}$ we take the derivative of the above with respect to $\hat{x}$, set the result equal to zero, and solve for $\hat{x}$. The first derivative is given by

$$\left(\frac{\partial J'(z)}{\partial \hat{x}}\right)^T = H^T S^{-1} H\hat{x} - \frac{1}{2}H^T S^{-T} z - \frac{1}{2}H^T S^{-1} z = H^T S^{-1} H\hat{x} - H^T S^{-1} z , \tag{118}$$

since S is symmetric. Setting this equal to zero and solving for $\hat{x}$ gives

$$\hat{x} = (H^T S H)^{-1} H^T S^{-1} z . \tag{119}$$

This is the procedure used to solve the weighted least squares problem. It involves the **weighted left pseudoinverse** matrix H^{WL} defined by

$$H^{\mathrm{WL}} = (H^T S H)^{-1} H^T S^{-1} . \tag{120}$$

Having introduced how to *solve* the weighed least squares problem the book then introduces two possible matrices for use as S the matrix that represents the squares of the measurement errors in each component of $z - \hat{y}$. The first choice for S, denoted as S_1, is based on the measurement error covariance as

$$S_1 \equiv E[(z - y)(z - y)^T] = R . \tag{121}$$

This version of S incorporates *only* error randomness that comes from the measurements we take i.e. measurement errors that arise from the term n. The second choice for S or S_2, is the expected measurement residual covariance and is based on the variance in the *predicted* error $z - H\hat{x}$. This form for S incorporates errors in both the measurement (from n) *and*

errors that come from our estimate of the state $\hat{x}$. When we use $z = Hx + n$ we have that S_2 can be written as

$$S_2 \equiv E[(z - H\hat{x})(z - H\hat{x})^T] = E[(H\epsilon_x + n)(H\epsilon_x + n)^T], \tag{122}$$

since $\epsilon_x \equiv x - \hat{x}$. Expanding the quadratic above we get that S_2 can be written as

$$\begin{aligned} S_2 &= HE[\epsilon_x \epsilon_x^T]H^T + HE[\epsilon_x n^T] + E[n\epsilon_x^T]H^T + E[nn^T] \\ &= HPH^T + HM + HM^T + R. \end{aligned} \tag{123}$$

Where we have defined $P \equiv E[\epsilon_x \epsilon_x^T]$ and $M \equiv E[\epsilon_x n^T]$, which we now need to evaluate. Using $z = y + n$ and $\hat{x} = H^{\mathrm{WL}} z$ with H^{WL} given by Equation 120 we have

$$\begin{aligned} \hat{x} &= H^{\mathrm{WL}} y + H^{\mathrm{WL}} n = (H^T S H)^{-1} H^T S^{-1} H x + H^{\mathrm{WL}} n \\ &= x + (H^T S H)^{-1} H^T S^{-1} n, \end{aligned} \tag{124}$$

and we have computed $\epsilon_x = x - \hat{x}$. Using this we have that P is given by

$$\begin{aligned} P \equiv E[(x - \hat{x})(x - \hat{x})^T] &= E[(H^T S H)^{-1} H^T S^{-1} n n^T S^{-1} H (H^T S^{-1} H)^{-1}] \\ = (H^T S H)^{-1} H^T S^{-1} R S^{-1} H (H^T S^{-1} H)^{-1}. \end{aligned} \tag{125}$$

While M is given by

$$\begin{aligned} M \equiv E[(x - \hat{x}) n^T] &= -E[(H^T S H)^{-1} H^T S^{-1} n n^T] \\ &= -(H^T S H)^{-1} H^T S^{-1} R. \end{aligned} \tag{126}$$

In the above formulas we have expressed a set of relationships that must hold between the matrices P, M, and S. The relationships are more or less coupled depending on the choice for S. In the *simplest* case $S = S_1 = R$ and P and M become

$$P = (H^T R^{-1} H)^{-1} H^T R^{-1} H (H^T R^{-1} H)^{-1} = (H^T R^{-1} H)^{-1} \tag{127}$$

$$M = -(H^T R^{-1} H)^{-1} H^T. \tag{128}$$

When $S = S_2$ then S depends on P and M via Equation 123 so we have to find values for all three matrices that are consistent. These three matrices must satisfy Equations 123, 125, and 126 and in general there maybe only a numerical solution to this problem. To obtain a numerical solution put Equation 125 (the expression for P) into Equation 126 (the expression for M) into Equation 123 (the expression for S_2). We find

$$\begin{aligned} S_2 &= H\left[(H^T S_2^{-1} H)^{-1} H^T S_2^{-1} R S_2^{-1} H (H^T S_2^{-1} H)^{-1}\right] H^T \\ &\quad - H(H^T S_2 H)^{-1} H^T S_2^{-1} R - R S_2^{-1} H (H^T S_2 H)^{-1} H^T + R \\ &= [H(H^T S_2^{-1} H)^{-1} H^T S_2^{-1} - I_{k_1}] R [H(H^T S_2^{-1} H)^{-1} H^T S_2^{-1} - I_{k_1}]^T \\ &= L(S_2) R L(S_2)^T. \end{aligned} \tag{129}$$

Where we have defined $L(\cdot)$ a function of S_2 as

$$L(S_2) \equiv H(H^T S_2^{-1} H)^{-1} H^T S_2^{-1} - I_{k_1}. \tag{130}$$

Using this representation we can iteratively compute estimates for S_2 by first picking an initial value for S_2, say $R = S_1$, computing L via Equation 130, and computing a new estimate of S_2 using Equation 129.

Notes on a least squares parameter estimation problem example

For the system given in this example we can create a vector containing all the parameters by taking p equal to $p \equiv \begin{bmatrix} p_1 \\ p_2 \\ p_3 \end{bmatrix} = \begin{bmatrix} \omega_n^2 \\ 2\zeta\omega_n \\ g \end{bmatrix}$, then given the two unknowns x_1, x_2, and the forcing u, the exact measurement of $\dot{x}_2$, is given by

$$\dot{x}_2 = -\omega_n^2 x_1 - 2\zeta\omega_n x_2 + gu\,.$$

In terms of the components of the parameter vector p this is

$$\dot{x}_2 = -p_1 x_1 - p_2 x_2 + p_3 u\,.$$

If was assume that we now *measure* noise values x_2 we can write this single constraint in the form $z = hp + n$ for a row vector h given by $h = \begin{bmatrix} -x_1 & -x_2 & u \end{bmatrix}$. Since this hold for only a single time point we assemble all h's by way of stacking to get a H matrix of

$$H = \begin{bmatrix} -x_{11} & -x_{21} & u_1 \\ -x_{12} & -x_{22} & u_2 \\ \vdots & \vdots & \vdots \\ -x_{1n} & -x_{2n} & u_n \end{bmatrix}.$$

Then as a system our parameter estimate for p will satisfy $z = Hp + n$, for a column vector z containing all measurements. The least-squares solution for this system for p is then $\hat{p} = (H^T H)^{-1} H^T z$.

Notes on recursive least squares estimation

In this section we derive a form for recursive estimation of our unknown state x. We assume that have k_1 measurements in the vector z_1 and k_2 measurements in the vector z_2, where the vector z_1 is observed first and then the vector z_2 is observed. Our first set of measurements are related to our state x in the usual way with $z_1 = H_1 x + n_1$, with $E[n_1 n_1^T] = R_1$, and the optimal estimate of x is given by

$$\hat{x}_1 = (H_1^T R_1^{-1} H_1)^{-1} H_1^T R_1^{-1} z_1\,. \tag{131}$$

Then given a new set of k_2 measurements in the vector z_2 we can derive a *recursive* update formula used to process these measurements by asking what the optimal estimator, $\hat{x}_2$, would be after we processed the single vector containing *all* the measurements or $z = \begin{bmatrix} z_1 \\ z_2 \end{bmatrix}$. To determine this we can form a criterion function to determine optimality or $J(z_1, z_2)$ as

$$J(\hat{x}_2; z_1, z_2) \equiv \frac{1}{2} \begin{bmatrix} z_1 - H_1\hat{x}_2 & z_2 - H_2\hat{x}_2 \end{bmatrix} \begin{bmatrix} R_1^{-1} & 0 \\ 0 & R_2^{-1} \end{bmatrix} \begin{bmatrix} z_1 - H_1\hat{x}_2 \\ z_2 - H_2\hat{x}_2 \end{bmatrix}.$$

Notice how the variable $\hat{x}_2$ is located in both coordinate positions in the block vectors in the above inner product. This is because we want to find the vector that will optimally predict

both sets of measurements z_1 and z_2. Expanding the block inner product in the quadratic above into two other inner products gives

$$J(\hat{x}_2) = \frac{1}{2}(z_1 - H_1\hat{x}_2)R_1^{-1}(z_1 - H_1\hat{x}_2) + \frac{1}{2}(z_2 - H_2\hat{x}_2)R_2^{-1}(z_2 - H_2\hat{x}_2)\,.$$

Taking the derivative with respect to $\hat{x}_2$ and setting the result equal to 0 gives

$$H_1^T R_1^{-1} H_1\hat{x}_2 - H_1^T R_1^{-1} z_1 + H_2^T R_2^{-1} H_2\hat{x}_2 - H_2^T R_2^{-1} z_2 = 0\,.$$

When we solve for $\hat{x}_2$ we find

$$\hat{x}_2 = (H_1^T R_1^{-1} H_1 + H_2^T R_2^{-1} H_2)^{-1}(H_1^T R_1^{-1} z_1 + H_2^T R_2^{-1} z_2)\,. \tag{132}$$

This is the *non-recursive* form of the estimator for $\hat{x}_2$, since any work we did to compute $\hat{x}_1$ must be "redone" when we compute the above expression. We next introduce P_1 as in Equation 127 as

$$P_1^{-1} \equiv H_1^T R_1^{-1} H_1^T\,, \tag{133}$$

by which we see that the estimate $\hat{x}_1$ from Equation 131 in terms of P_1 is given by

$$\hat{x}_1 = P_1 H_1^T R_1^{-1} z_1\,. \tag{134}$$

This functional form for $\hat{x}_1$ when we consider Equation 132 (in that we have a matrix inverse times expressions in z_i to compute our estimate $\hat{x}$) leads us to introduced a matrix P_2 defined as

$$P_2 \equiv (H_1^T R_1^{-1} H_1 + H_2^T R_2^{-1} H_2)^{-1}\,. \tag{135}$$

Next using the matrix inversion lemma Equation 18 we can write P_2 as

$$(P_1^{-1} + H_2^T R_2^{-1} H_2)^{-1} = P_1 - P_1 H_2^T (H_2 P_1 H_2^T + R_2)^{-1} H_2 P_1\,.$$

Using this in Equation 132 and the estimate of $\hat{x}_1$ from $\hat{x}_1 = P_1 H_1^T R_1^{-1} z_1$ the expression for $\hat{x}_2$ derived above can be written

$$\hat{x}_2 = \hat{x}_1 - P_1 H_2^T (H_2 P_1 H_2^T + R_2)^{-1} H_2\hat{x}_1 + P_1 H_2^T [I_{k_2} - (H_2 P_1 H_2^T + R_2)^{-1} H_2 P_1 H_2^T] R_2^{-1} z_2\,.$$

We replace the identity matrix I_{k_2} in the above with the expression

$$I_{k_2} = (H_2 P_1 H_2^T + R_2)^{-1}(H_2 P_1 H_2^T + R_2)\,,$$

so that we get for $\hat{x}_2$ the following

$$\begin{aligned}\hat{x}_2 &= \hat{x}_1 - P_1 H_2^T (H_2 P_1 H_2^T + R_2)^{-1}\left[H_2\hat{x}_1 + \left((H_2 P_1 H_2^T + R_2) - H_2 P_1 H_2^T\right) R_2^{-1} z_2\right] \\ &= \hat{x}_1 + P_1 H_2^T (H_2 P_1 H_2^T + R_2)^{-1}(z_2 - H_2\hat{x}_1)\,. \end{aligned} \tag{136}$$

Often we define the coefficient matrix in front of the vector $z_2 - H_2\hat{x}_2$ as K_2 the **recursive weighted-least-squares gain matrix**

$$K_2 = P_1 H_2^T (H_2 P_1 H_2^T + R_2)^{-1}\,. \tag{137}$$

Notes on recursive estimation of a scalar constant

Returning to the scalar estimation example on Page 85, where we have k_1 noisy measurements of a scalar $z_i = x + n_i$, for $i = 1, 2, \cdots, k_1$ followed by one new measurement. Then in the notation of the previous section the matrix H_1 is a column vector of all 1's and the measurement noise covariance R_1 is given by $R_1 = I_{k_1}$. Then $P_1 = (H_1^T R_1^{-1} H_1)^{-1} = \frac{1}{k_1}$ our estimate of x after the first set of k_1 measurements is

$$\hat{x}_1 = (H_1^T R_1^{-1} H_1)^{-1} H_1^T R_1^{-1} z_1 = \frac{1}{k_1} \sum_{k=1}^{k_1} z_k \,.$$

When a single new measurement z_{k_1+1} is observed we have $H_2 = 1$ with $R_2 = 1$. Thus using Equation 135 and 133 we have

$$P_2 = (P_1^{-1} + H_2^T R_2^{-1} H_2)^{-1} = (k_1 + 1)^{-1} \,.$$

Finally Equation 137 gives

$$k_2 = \left(\frac{1}{k_1}\right)(1)\left(\frac{1}{k_1} + 1\right)^{-1} = \frac{1}{k_1 + 1} \,,$$

so that the updated estimate of x or $\hat{x}_2$ is

$$\hat{x}_2 = \hat{x}_1 + k_2(z_{k_1+1} - \hat{x}_1) \,.$$

Notes on propagation of the state estimate and its uncertainty

When $n = 2$ the matrices P, Φ, Λ and Q' in terms of their components, are given by

$$P = \begin{bmatrix} p_{11} & p_{12} \\ p_{12} & p_{22} \end{bmatrix}, \quad \Phi = \begin{bmatrix} \phi_{11} & \phi_{12} \\ \phi_{21} & \phi_{22} \end{bmatrix}, \quad \Lambda = \begin{bmatrix} \lambda_{11} & \lambda_{12} \\ \lambda_{21} & \lambda_{22} \end{bmatrix}, \quad Q' = \begin{bmatrix} q_{11} & 0 \\ 0 & q_{22} \end{bmatrix}.$$

Thus we can compute the right-hand-side of the covariance update equation or the right-hand-side of

$$P_k = \Phi_{k-1} P_{k-1} \Phi_{k-1}^T + \Lambda_{k-1} Q'_{k-1} \Lambda_{k-1}^T \,. \tag{138}$$

In the Mathematica file `direct_propagation_of_state_uncertainty.nb` we perform the matrix multiplications above to get three equations for the $(1,1)$, $(1,2)$ and $(2,2)$ spots (by symmetry the equation for $(2,1)$ is a duplicate of the equation for $(1,2)$)

$$\begin{aligned}
&p_{11}\phi_{11}^2 + 2p_{12}\phi_{11}\phi_{12} + p_{22}\phi_{22}^2 + \lambda_{11}^2 q_{11} + \lambda_{12}^2 q_{22} \\
&p_{11}\phi_{11}\phi_{21} + p_{12}(\phi_{12}\phi_{21} + \phi_{11}\phi_{22}) + p_{22}\phi_{12}\phi_{22} + \lambda_{11}\lambda_{21} q_{11} + \lambda_{12}\lambda_{22} q_{22} \\
&p_{11}\phi_{21}^2 + 2p_{12}\phi_{21}\phi_{22} + p_{22}\phi_{22}^2 + \lambda_{21}^2 q_{11} + \lambda_{22}^2 q_{22} \,.
\end{aligned}$$

When these equations are place in a linear system with input vectors $\begin{bmatrix} p_{11} & p_{12} & p_{22} \end{bmatrix}^T$ and $\begin{bmatrix} q_{11} & q_{22} \end{bmatrix}^T$ (evaluated at t_{k-1}) we get the linear system given in the book.

Notes on autocorrelated process noise: state augmentation

We consider the original discrete state update equation

$$x_k = \Phi_{k-1}x_{k-1} + \Gamma_{k-1}u_{k-1} + \Lambda_{k-1}w_{k-1}\,, \tag{139}$$

with the Markov model for the noise w_k given by

$$w_k = A_{k-1}w_{k-1} + \eta_{k-1}\,.$$

To fit this situation into the Kalman filtering framework we have developed thus far we will extend the original state x_k to include the noise w_k to get the expanded state $X_k = \begin{bmatrix} x_k \\ w_k \end{bmatrix}$. For this enlarged state we find that the state dynamics propagate with

$$\begin{bmatrix} x_k \\ w_k \end{bmatrix} = \begin{bmatrix} \Phi_{k-1} & \Lambda_{k-1} \\ 0 & A_{k-1} \end{bmatrix} \begin{bmatrix} x_{k-1} \\ w_{k-1} \end{bmatrix} + \begin{bmatrix} \Gamma_{k-1} \\ 0 \end{bmatrix} u_{k-1} + \begin{bmatrix} 0 \\ I_s \end{bmatrix} \eta_{k-1} \tag{140}$$

Thus this augmented system has process noise vector of the form η_{k-1} multiplied by the augmented Λ matrix $\begin{bmatrix} 0 \\ I_s \end{bmatrix}$. Then using Equation 138 on this augmented system the matrix propagation equation is given by

$$\begin{bmatrix} P & 0 \\ 0 & W \end{bmatrix}_k = \begin{bmatrix} \Phi & \Lambda \\ 0 & A \end{bmatrix}_{k-1} \begin{bmatrix} P & 0 \\ 0 & W \end{bmatrix}_{k-1} \begin{bmatrix} \Phi & \Lambda \\ 0 & A \end{bmatrix}_{k-1}^T + \begin{bmatrix} 0 \\ I_s \end{bmatrix} Q_{k-1} \begin{bmatrix} 0 & I_s \end{bmatrix}\,. \tag{141}$$

Notes on sampled-data representation of continuous time systems

From the theory of linear systems we can represent the solution at t_k in terms of the state at t_{k-1} using the state propagation matrix, the control $u(\tau)$, and the noise $w(\tau)$ as

$$x_k = \Phi_{k-1}x_{k-1} + \int_{t_{k-1}}^{t_k} \Phi(t_k,\tau)[G(\tau)u(\tau) + L(\tau)w(\tau)]d\tau\,. \tag{142}$$

Taking the expectation of this expression the mean $m_k \equiv E[x_k]$ propagates according to

$$m_k = \Phi_{k-1}m_{k-1} + \int_{t_{k-1}}^{t_k} \Phi(t_k,\tau)G(\tau)u(\tau)d\tau\,.$$

So the difference $x_k - m_k$ is

$$x_k - m_k = \Phi_{k-1}(x_{k-1} - m_{k-1}) + \int_{t_{k-1}}^{t_k} \Phi(t_k,\tau)L(\tau)w(\tau)d\tau\,.$$

Using this we can "square" this vector and compute its expectation. By defining P_k to be $P_k \equiv E[(x_k - m_k)(x_k - m_k)^T]$ we find

$$P_k = \Phi_{k-1}P_{k-1}\Phi_{k-1}^T + E\left[\int_{t_{k-1}}^{t_k}\int_{t_{k-1}}^{t_k} \Phi(t_k,\tau)L(\tau)w(\tau)w^T(\alpha)L^T(\alpha)\Phi^T(t_k,\alpha)d\alpha d\tau\right]\,. \tag{143}$$

Now using $E[w(\tau)w^T(\alpha)] = Q_C'(\tau)\delta(\tau-\alpha)$ we can evaluate the α integral in the above. We define the resulting expression to be Q_{k-1} and get

$$Q_{k-1} = \int_{t_{k-1}}^{t_k} \Phi(t_k,\tau)L(\tau)Q_C'(\tau)L^T(\tau)\Phi^T(t_k,\tau)d\tau\,. \tag{144}$$

This expression is how we obtain the discrete noise covariance matrix given its continuous system. We now consider a couple of examples.

For the scalar system $\dot{x}(t) = fx(t) + w(t)$ our state-transition matrix is given by $\phi(t,t_0) = e^{f(t-t_0)}$. Then using Equation 144 to compute the discrete time state propagation covariance matrix q_{k-1} we have

$$\begin{aligned} q_{k-1} &= \int_{t_{k-1}}^{t_k} e^{f(t_k-\tau)}\,1\,q_C\,1\,e^{f(t_k-\tau)}d\tau = q_C\int_{t_{k-1}}^{t_k} e^{2f(t_k-\tau)}d\tau = q_C e^{2ft_k}\int_{t_{k-1}}^{t_k} e^{-2f\tau}d\tau \\ &= \frac{q_C}{2f}e^{2ft_k}(e^{-2ft_{k-1}} - e^{-2ft_k}) = \frac{q_C}{2f}(e^{-2f\Delta t}-1)\,. \end{aligned}$$

For the two dimensional system

$$\frac{dx}{dt} = \begin{bmatrix} 0 & 1 \\ a & b \end{bmatrix} x + \begin{bmatrix} 0 \\ w \end{bmatrix},$$

with $E[w(t)^2] = q_C\delta(t-\tau)$. To use Equation 144 to evaluate Q_{k-1} in forming the discrete model from the continuous model we note that $L(\tau) = I$, $Q_C' = \begin{bmatrix} 0 & 0 \\ 0 & q_C \end{bmatrix}$, and because our system matrix $F = \begin{bmatrix} 0 & 1 \\ a & b \end{bmatrix}$ is independent of time that the state-transition matrix $\Phi(t,\tau)$ is given by $\Phi(t,\tau) = e^{F(t-\tau)}$, thus we see that

$$Q_{k-1} = \int_{t_{k-1}}^{t_k} e^{F(t_k-\tau)}\begin{bmatrix} 0 & 0 \\ 0 & q_C \end{bmatrix} e^{F^T(t_k-\tau)}d\tau\,.$$

If we let $\xi = t_k - \tau$ so that $d\xi = -d\tau$ and the above becomes

$$Q_{k-1} = \int_0^{\Delta t} e^{F\xi}\begin{bmatrix} 0 & 0 \\ 0 & q_C \end{bmatrix} e^{F^T\xi}d\xi\,. \tag{145}$$

Note that we could factor q_C out of this integral to make the inner most matrix a zero matrix with only a single 1 at the $(2,2)$ position. To evaluate this expression further we need to be able to compute the matrix exponential e^{Ft} in some way. Since the functional form from F is known to be $\begin{bmatrix} 0 & 1 \\ a & b \end{bmatrix}$ and is independent of t, this can be done in several ways (probably more). These include

- Using the definition of e^{Ft} as n solutions to the original differential equation system with the n initial conditions given by the n columns of the identity matrix.

- Using a Taylor series representation to evaluate e^{Ft}.
- Inverse Laplace transforms of the inverse of $sI_n - F$ in that

$$\mathcal{L}^{-1}((sI_n - F)^{-1}) = e^{Ft} = \Phi(t, 0)\,,$$

  if the inverse Laplace transform of the elements of $(sI_n - F)^{-1}$ are easy to compute.
- Evaluate the expression e^{Ft} numerically for any needed t. In computing $\Phi(t, \tau)$ numerically we could use any of several methods aimed at solving the original differential equation or to make things simple we could simply use the MATLAB function `expm`.

Using the last method in the MATLAB script `sect_2_dup_example_4_2_2.m` we call the function `example_4_2_2_compute_qkm1.m` to compute the value of Q_{k-1} for several numerical values of a, b, and Δt. We specify the value of a, b, and Δt we want to compute Q_{k-1} for so that we can duplicate the table presented in the book. When the above MATLAB script we get the following table of results (these match the book's results quite nicely)

```
     a          b         dt:        q11        q12        q22
-1.0000    -2.0000    0.0100:     0.0000     0.0000     0.0098
-1.0000    -2.0000    0.1000:     0.0003     0.0041     0.0822
-1.0000    -2.0000    1.0000:     0.0808     0.0677     0.2162
-1.0000     2.0000    0.0100:     0.0000     0.0001     0.0102
-1.0000     2.0000    0.1000:     0.0004     0.0061     0.1225
-1.0000     2.0000    1.0000:     1.5975     3.6950     8.9870
 1.0000    -2.0000    0.0100:     0.0000     0.0000     0.0098
 1.0000    -2.0000    0.1000:     0.0003     0.0041     0.0827
 1.0000    -2.0000    1.0000:     0.1122     0.1267     0.2912
 1.0000     2.0000    0.0100:     0.0000     0.0001     0.0102
 1.0000     2.0000    0.1000:     0.0004     0.0061     0.1234
 1.0000     2.0000    1.0000:     2.4975     6.9186    19.5292
```

Notes on Example 4.2-3 the weathervane angle rate uncertainty

Since the system matrix F is of the same type as we have seen earlier (Page 93) we can use the MATLAB code developed earlier in `example_4_2_2_compute_qkm1.m` to duplicate this example. Here we take the matrix L as the identity and the process noise vector $\mathbf{w}(t)$ in terms of the scalar function $w(t)$ as $\mathbf{w}(t) = \begin{bmatrix} 0 \\ \omega_n^2 \end{bmatrix} w(t) = \begin{bmatrix} 0 \\ \omega_n^2 w(t) \end{bmatrix}$. Thus we get that

$$L(\tau)Q_C'(\tau)L(\tau)^T = \begin{bmatrix} 0 & 0 \\ 0 & \omega_n^4 \times 1 \end{bmatrix} = \omega_n^4 \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}\,,$$

which indicates that when we compute $Q_{k-1} = \mathbf{Q}(\Delta t)$ using Equation 144 we can factor out a leading ω_n^4 as discussed around Equation 145 to get

$$Q_{k-1} = \omega_n^4 \int_0^{\Delta t} e^{F\xi} \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} e^{F^T\xi} d\xi\,.$$

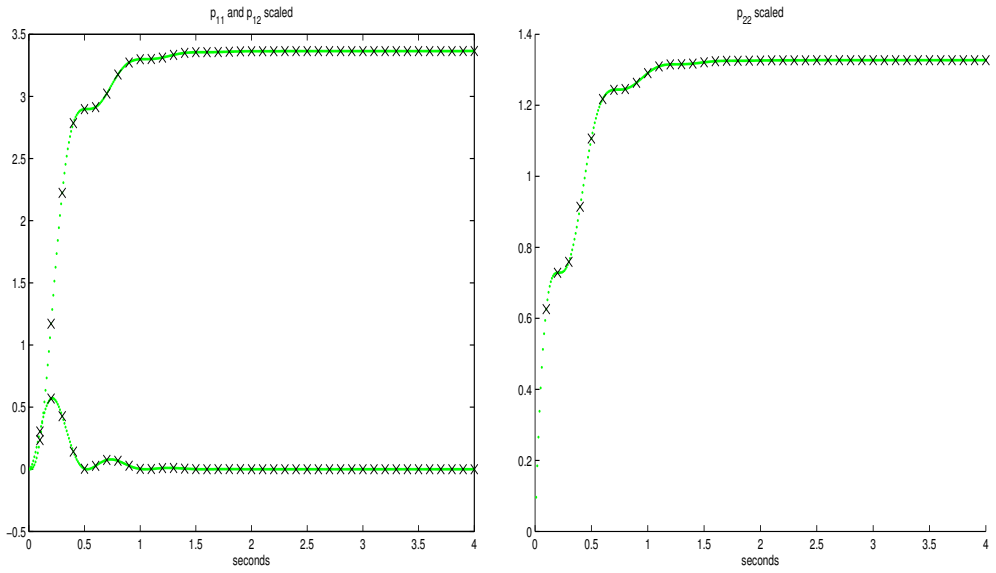

Figure 17: **Left:** Plots of p_{11} and p_{12} for $\Delta t = 0.1$ and $\Delta t = 0.01$. **Right:** Plots of p_{22} for $\Delta t = 0.1$ and $\Delta t = 0.01$. The $\times$ correspond to the larger value for Δt.

We can evaluate this by calling the routine `example_4_2_2_compute_qkm1.m`. In the MATLAB script `sect_2_dup_example_4_2_3.m` we do just that and get

$$Q(0.1) = \begin{bmatrix} 0.0002 & 0.0030 \\ 0.0030 & 0.0626 \end{bmatrix},$$

and $\Phi(0.1)$ is given by

$$\Phi(0.1) = e^{F\,0.1} = \begin{bmatrix} 0.8309 & 0.0780 \\ -3.0747 & 0.5372 \end{bmatrix},$$

these match the results from the book. We can perform the same calculations using a smaller value for Δt say 0.01. When we plot the evolution of p_{11}, p_{12} and p_{22} under the two we obtain the plots given in Figure 17.

Notes on the Kalman filter

Using the symmetric matrix inversion lemma given by Equation 18 we can derive an alternative expression for $P_k(+)$. We have

$$\begin{aligned} P_k(+) &= [P_k(-) + H_k^T R_k^{-1} H_k]^{-1} \\ &= P_k(-) - P_k(-) H_k^T [H_k P_k(-) H_k^T + R_k]^{-1} H_k P_k(-)\,. \end{aligned} \tag{146}$$

But from the expression

$$K_k = P_k(-) H_k^T (H_k P_k(-) H_k^T + R_k)^{-1}\,, \tag{147}$$

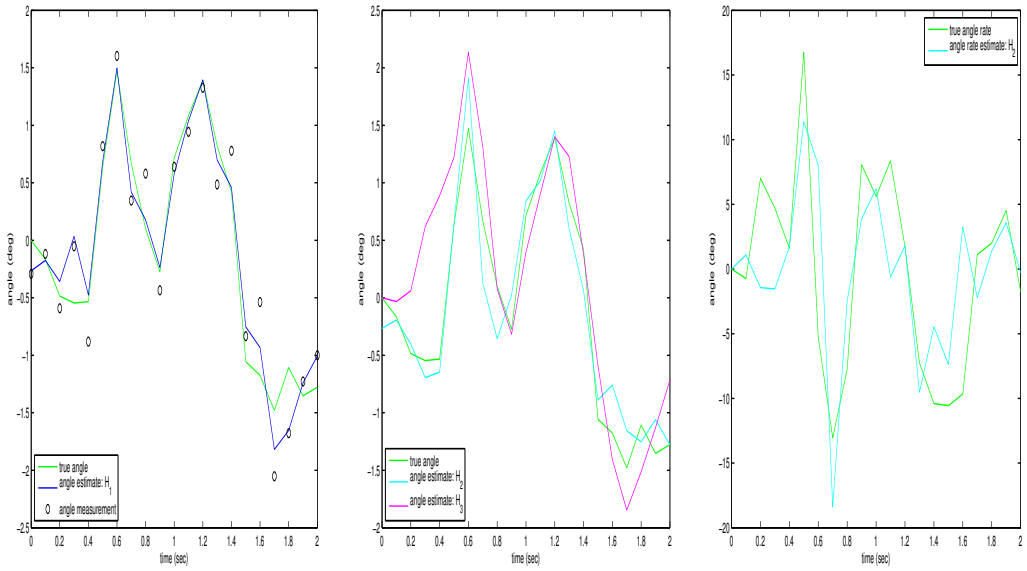

Figure 18: **Left:** Angle estimates with two measurements. **Center:** Angle estimates with one measurement. **Right:** Angular rate measurements with angle measurement. The truth is plotted in green, measurements in black. Estimates with $H = H_1$ are plotted in blue, estimates with $H = H_2$ in cyan, with $H = H_3$ with magenta.

for K_k derived earlier, the second term in the above expression for $P_k(+)$ can be written as

$$P_k(+) = P_k(-) - K_k H_k P_k(-) = (I_n - K_k H_k) P_k(-) \,. \tag{148}$$

From the *iterative* expression for K_k (K_k is on both sides of the equals sign) derived in the book and given by

$$K_k = (I_n - K_k H_k) P_k(-) H_k^T R_k^{-1} \,, \tag{149}$$

we can use this expression by post-multiplying both sides of Equation 148 by $H_k^T R_k^{-1}$ to which the resulting right-hand-side then becomes K_k to obtain

$$K_k = P_k(+) H_k^T R_k^{-1} \,. \tag{150}$$

Notes on Example 4.3-1 the weathervane angle rate estimation

In the MATLAB script `sect_3_dup_example_4_3_1.m` we duplicate the example from this section. Plots of the corresponding images are given in Figure 18. We can also look at the steady-state Kalman estimates of $P(+)$. If we consider the default parameter case we get

```
          p11         p12         p22
H1:   0.066717    0.001547    0.099826
H2:   0.089934    0.725597   61.163986
H3:   1.060471    0.001881    0.099907
```

The book then states if we divide Q by ω_n^4 we get a "low-signal-to-noise" case. Since I found $\omega_n^4 = 1555$ when one divides Q by this the process noise actually *decreases* so I would expect the signal-to-noise to *increase*. In any case for this case we get the following steady-state covariance

```
          p11         p12         p22
H1:    0.001737    0.000347    0.047826
H2:    0.003137    0.000471    0.127672
H3:    0.001806    0.000255    0.048111
```

Both of these numbers agree quite well with the numbers given in the book.

Notes on Alternative forms of the linear-optimal filter

We want to use the measurement update equation given by

$$\hat{x}_k(+) = (I_n - K_k H_k)\hat{x}_k(-) + K_k z_k \,, \tag{151}$$

to derive an alternative estimate of the filtering error covariance $P_k(+)$. We begin by defining

$$\epsilon_k(-) \equiv x_k - \hat{x}_k(-) \tag{152}$$
$$\epsilon_k(+) \equiv x_k - \hat{x}_k(+) \,. \tag{153}$$

Which when put into Equation 151 gives

$$x_k - \epsilon_k(+) = (I_n - K_k H_k)(x_k - \epsilon_k(-)) + K_k z_k \,,$$

or

$$\epsilon_k(+) = K_k H_K x_k + (I_n - K_k H_k)\epsilon_k(-) - K_k z_k \,.$$

But our measurement is given by $z_k = H_k x_k + n_k$ so the above becomes

$$\epsilon_k(+) = K_k H_k x_k + (I_n - K_k H_k)\epsilon_k(-) - K_k H_k x_k - K_k n_k \,,$$

or

$$\epsilon_k(+) = (I_n - K_k H_k)\epsilon_k(-) - K_k n_k \,. \tag{154}$$

Using this expression we can obtain the **Joseph** form for $P_k(+) \equiv E[\epsilon_k(+)\epsilon_k(+)^T]$ given by

$$P_k(+) = (I_n - K_k H_k)P_k(-)(I_n - K_k H_k)^T + K_k R_k K_k^T \,. \tag{155}$$

This expression assumes *uncorrelated* disturbance and measurement noise i.e. $E[w_{k-1} n_k^T] = 0$. See Page 98 for the case where the disturbance and measurement noise are correlated.

Correlation of Disturbance Input and Measurement Noise i.e. $E[w_{k-1}n_k^T] \equiv M_k \neq 0$

In this section of these notes we want to consider the situation where the disturbance noise w_{k-1} in going from t_{k-1} to t_k and the measurement noise n_k at time t_k are cross-correlated. This means that $E[w_{k-1}n_k^T] \equiv M_k \neq 0$. We start with $\epsilon_k(-)$ given by Equation 152 which we write as

$$\begin{aligned}\epsilon_k(-) &= x_k - \hat{x}_k(-) = \Phi_{k-1}x_{k-1} + w_{k-1} - \Phi_{k-1}\hat{x}_{k-1}(+) \\ &= \Phi_{k-1}\epsilon_{k-1}(+) + w_{k-1}\,, \end{aligned} \tag{156}$$

and $\epsilon_k(+)$ given by using Equation 154 which we write as

$$\epsilon_k(+) = \epsilon_k(-) - K_k[H_k\epsilon_k(-) + n_k]\,. \tag{157}$$

Now when we use Equation 157 to compute $P_k(+) = E[\epsilon_k(+)\epsilon_k(+)^T]$ there will be terms $E[\epsilon_k(-)n_k^T]$ which by Equation 156 and the fact that $E[w_{k-1}n_k^T] \neq 0$ *will not vanish*. For example, we find $P_k(+)$ given by

$$\begin{aligned}P_k(+) &\equiv E[(\epsilon_k(-) - K_kH_k\epsilon_k(-) - K_kn_k)(\epsilon_k(-) - K_kH_k\epsilon_k(-) - K_kn_k)^T] \\ &= E[\epsilon_k(-)\epsilon_k(-)^T - \epsilon_k(-)\epsilon_k(-)^TH_k^TK_k^T - \epsilon_k(-)n_k^TK_k^T \\ &- K_kH_k\epsilon_k(-)\epsilon_k(-)^T + K_kH_k\epsilon_k(-)\epsilon_k(-)^TH_k^TK_k^T + K_kH_k\epsilon_k(-)n_k^TK_k^T \\ &- K_kn_k\epsilon_k(-)^T + K_kn_k\epsilon_k(-)^TH_k^TK_k^T + K_kn_kn_k^TK_k^T] \\ &= P_k(-) - P_k(-)H_k^TK_k^T - E[\epsilon_k(-)n_k^T]K_k^T \\ &- K_kH_kP_k(-) + K_kH_kP_k(-)H_k^TK_k^T + K_kH_kE[\epsilon_k(-)n_k^T]K_k^T \\ &- K_kE[n_k\epsilon_k(-)^T] + K_kE[n_k\epsilon_k(-)^T]H_k^TK_k^T + K_kR_kK_k^T\,. \end{aligned}$$

The first, second, fourth, and fifth terms in the above combine to give

$$(I_n - K_kH_k)P_k(-)(I_n - K_kH_k)^T\,,$$

as can be seen by expanding this last expression out. Thus we have for $P_k(+)$ the following

$$\begin{aligned}P_k(+) &= (I_n - K_kH_k)P_k(-)(I_n - K_kH_k)^T + K_kR_kK_k^T \\ &- E[\epsilon_k(-)n_k^T]K_k^T + K_kH_kE[\epsilon_k(-)n_k^T]K_k^T - K_kE[n_k\epsilon_k(-)^T] + K_kE[n_k\epsilon_k(-)^T]H_k^TK_k^T\,. \end{aligned}$$

Now we use the fact that $\epsilon_k(-) = \Phi_{k-1}\epsilon_{k-1}(+) + w_{k-1}$ to evaluate the above. We get

$$E[\epsilon_k(-)n_k^T] = \Phi_{k-1}E[\epsilon_{k-1}(+)n_k^T] + E[w_{k-1}n_k^T] \equiv M_k\,,$$

because $E[\epsilon_{k-1}(+)n_k^T] = 0$ since the new measurement error or n_k, at time step k is independent of any error (a priori or a posteriori) we have in our filtering at the previous time step $k-1$. When we use this fact to take the expectation of these last four terms in $P_k(+)$ we get

$$-M_kK_k^T - K_kM_k^T + K_kH_kM_kK_k^T + K_kM_k^TH_k^TK_k^T = -M_kK_k^T - K_kM_k^T + K_k(H_kM_k + M_k^TH_k^T)K_k^T\,.$$

Thus we finally end with $P_k(+)$ taking the following form

$$\begin{aligned}P_k(+) &= (I_n - K_kH_k)P_k(-)(I_n - K_kH_k)^T + K_kR_kK_k^T \\ &+ K_k(H_kM_k + M_k^TH_k^T)K_k^T - M_kK_k^T - K_kM_k^T\,. \end{aligned} \tag{158}$$

To evaluate the optimal gain K_k, we form the objective function $J_k \equiv \mathrm{Tr}[E[\epsilon_k(+)\epsilon_k(+)^T]] = \mathrm{Tr}[P_k(+)]$ and pick the Kalman gain K_k so that we minimize J_k. To do this we need to recall the following matrix derivatives of traces

$$\frac{\partial}{\partial A}\mathrm{Tr}(ABA^T) = 2AB \quad \text{and} \quad \frac{\partial}{\partial A}\mathrm{Tr}(AB^T) = B \quad \text{and} \quad \frac{\partial}{\partial A}\mathrm{Tr}(BA^T) = B\,. \tag{159}$$

To use these derivatives most easily it will be helpful to write $P_k(+)$ as

$$\begin{aligned} P_k(+) &= P_k(-) + K_k[H_kP_k(-)H_k^T + R_k + H_kM_k + M_k^TH_k^T]K_k^T \\ &\quad - (P_k(-)H_k^T + M_k)K_k^T - K_k(H_kP_k(-) - M_k^T)\,. \end{aligned} \tag{160}$$

Taking the trace of this expression and then the K_k derivative we have

$$\frac{\partial J_k}{\partial K_k} = 2K_k[H_kP_k(-)H_k^T + R_k + H_kM_k + M_k^TH_k^T] - 2P_k(-)H_k^T - 2M_k\,. \tag{161}$$

When we set this equal to zero and then solve for K_k we get

$$K_k = [P_k(-)H_k^T + M_k][H_kP_k(-)H_k^T + R_k + H_kM_k + M_k^TH_k^T]^{-1}\,. \tag{162}$$

For notational simplicity in the next derivation, lets denote the symmetric matrix $H_kP_k(-)H_k^T + R_k + H_kM_k + M_k^TH_k^T$ as A_k. Now using the optimal value for K_k given by Equation 162 in the expression for $P_k(+)$ given by Equation 160 we find

$$\begin{aligned} P_k(+) &= P_k(-) + (P_k(-)H_k^T + M_k)A_k^{-1}A_kA_k^{-1}(H_kP_k(-) + M_k^T) \\ &\quad - (P_k(-)H_k^T + M_k)A_k^{-1}(H_kP_k(-) + M_k^T) - (P_k(-)H_k^T + M_k)A_k^{-1}(H_kP_k(-) + M_k^T) \\ &= P_k(-) - (P_k(-)H_k^T + M_k)A_k^{-1}(H_kP_k(-) + M_k^T) \\ &= P_k(-) - K_k(H_kP_k(-) + M_k^T)\,, \end{aligned} \tag{163}$$

for the a posteriori covariance matrix in the case where we have correlated disturbance and measurement noise.

Notes on time-correlated measurement noise

In this section we derive some of the Kalman filtering expressions presented in the book. We don't derive all of the results presented in the book, since several of the results are simply quoted from various papers. The book states that in the case of time-correlated measurement noise state-augmentation by appending the measurement noise n_k is not a good idea. Instead we frame the problem in terms of the measurement *difference* ζ_{k-1} defined as

$$\zeta_{k-1} \equiv z_k - \Psi_{k-1}z_{k-1}\,, \tag{164}$$

where we assume that our measurement noise n_k, is given by the first order Markov process

$$n_k = \Psi_{k-1}n_{k-1} + \nu_{k-1}\,, \tag{165}$$

with $E[\nu_k \nu_k^T] = Q_{\nu_k}$. Then we can write ζ_{k-1} using Equation 139 assuming no control $\Gamma_{k-1} = 0$ and $\Lambda_{k-1} = I$ as

$$\begin{aligned}
\zeta_{k-1} &= z_k - \Psi_{k-1} z_{k-1} \\
&= H_k x_k + n_k - \Psi_{k-1}[H_{k-1} x_{k-1} + n_{k-1}] \\
&= H_k(\Phi_{k-1} x_{k-1} + w_{k-1}) + (\Psi_{k-1} n_{k-1} + \nu_{k-1}) - \Psi_{k-1}(H_{k-1} x_{k-1} + n_{k-1}) \\
&= (H_k \Phi_{k-1} - \Psi_{k-1} H_{k-1}) x_{k-1} + H_k w_{k-1} + \nu_{k-1} \,, \qquad (166)
\end{aligned}$$

since the terms $\Psi_{k-1} n_{k-1}$ cancel. Motivated by this we define *derived* measurement mapping coefficients D_{k-1} and derived noise $n_{D_{k-1}}$ as

$$D_{k-1} \equiv H_k \Phi_{k-1} - \Psi_{k-1} H_{k-1} \qquad (167)$$

$$n_{D_{k-1}} \equiv H_k w_{k-1} + \nu_{k-1} \,, \qquad (168)$$

so that in terms of these derived variables we have

$$\zeta_{k-1} = D_{k-1} x_{k-1} + n_{D_{k-1}} \,. \qquad (169)$$

It is this derived measurement equation that we would then work with. It is helpful to state the connection to the results from the previous section. To do that we note that this derived measurement noise $n_{D_{k-1}}$ is now cross-correlated with the original process noise w_{k-1}, since

$$\begin{aligned}
E\left[\begin{bmatrix} w_{k-1} \\ n_{D_{k-1}} \end{bmatrix} \begin{bmatrix} w_{k-1}^T & n_{D_{k-1}}^T \end{bmatrix} \right] &= \begin{bmatrix} Q_{k-1} & E[w_{k-1} w_{k-1}^T] H_k^T + E[w_{k-1} \nu_{k-1}^T] \\ H_k Q_{k-1} & E[(H_k w_{k-1} + \nu_{k-1})(H_k w_{k-1} + \nu_{k-1})^T] \end{bmatrix} \\
&= \begin{bmatrix} Q_{k-1} & Q_{k-1} H_k^T \\ H_k Q_{k-1} & H_k Q_{k-1} H_k^T + Q_{\nu_{k-1}} \end{bmatrix} ,
\end{aligned}$$

which has non-zero off diagonal elements since $Q_{k-1} H_k^T \neq 0$. At this point we should take a step back from the derivations that we have been performing to try to look at the bigger picture. By formulating the time-correlated measurement noise problem in terms of the difference $z_k - \Psi_{k-1} z_{k-1}$ we have reduced this problem to the case of cross-correlated disturbance noise which we have already seen how to solve on Page 98. Thus we could now implement a filter designed with cross-correlated disturbance and measurement noise and have the full solution to our problem.

Notes on the continuous Kalman-Bucy filter

In this section of these notes we start with the discrete Kalman filtering results derived above and then take limits as the sampling interval Δt shrinks to zero to derive *continuous* versions of the Kalman filtering equations. To begin, first note that when $\Delta t \ll 1$ we have that $\Phi_{k-1} \approx I + F_{k-1} \Delta t$ so that the discrete covariance propagation Equation 138 when $\Lambda_{k-1} = I$ becomes

$$P_k(-) = [I + F_{k-1} \Delta t] P_{k-1}(+) [I + F_{k-1} \Delta t]^T + Q_{k-1} \,.$$

Using the a posteriori covariance Equation 148 to replace $P_{k-1}(+)$ in the above we have that

$$\begin{aligned}
P_k(-) &= [I + F_{k-1}\Delta t](I - K_{k-1}H_{k-1})P_{k-1}(-)[I + F_{k-1}\Delta t]^T + Q_{k-1} \\
&= (I - K_{k-1}H_{k-1} + F_{k-1}\Delta t - F_{k-1}K_{k-1}H_{k-1}\Delta t)[P_{k-1}(-) + P_{k-1}(-)F_{k-1}^T\Delta t] + Q_{k-1} \\
&= P_{k-1}(-) + P_{k-1}(-)F_{k-1}^T\Delta t - K_{k-1}H_{k-1}P_{k-1}(-) - K_{k-1}H_{k-1}P_{k-1}(-)F_{k-1}^T\Delta t \\
&+ F_{k-1}P_{k-1}(-)\Delta t + O(\Delta t^2) - F_{k-1}K_{k-1}H_{k-1}P_{k-1}(-)\Delta t - O(\Delta t^2) + Q_{k-1}\,,
\end{aligned}$$

when we expand all products. If we form the first difference in time of $P_k(-)$ we get

$$\begin{aligned}
\frac{P_k(-) - P_{k-1}(-)}{\Delta t} &= F_{k-1}P_{k-1}(-) + P_{k-1}(-)F_{k-1}^T + \frac{Q_{k-1}}{\Delta t} \\
&- \frac{K_{k-1}}{\Delta t}H_{k-1}P_{k-1}(-) - F_{k-1}K_{k-1}H_{k-1}P_{k-1}(-) \\
&- K_{k-1}H_{k-1}P_{k-1}(-)F_{k-1}^T + O(\Delta t)\,. \qquad (170)
\end{aligned}$$

We now need to evaluate the how the *discrete* variables Q_{k-1} and K_{k-1} are related to the continuous spectral densities $Q_C'(t)$ and $R_C(t)$ as Δt shrinks to zero.

From Equation 144 (and dropping the prime on Q_C found there) we can write

$$Q_{k-1} \approx L(t_{k-1})Q_C(t_{k-1})L^T(t_{k-1})\Delta t\,,$$

and when we have a constant error spectral density R_C in the interval $t_{k-1} < t < t_k$ we have

$$R_{k-1} = \frac{R_C(t_{k-1})}{\Delta t}\,, \qquad (171)$$

so $R_{k-1}^{-1} = R_C^{-1}(t_{k-1})\Delta t$. The optimal expression for the discrete Kalman gain K_{k-1} is given by Equation 150 and gives

$$K_{k-1} \approx P_{k-1}(+)H_{k-1}^T R_C^{-1}(t_{k-1})\Delta t\,.$$

Lets define the continuous Kalman gain $K_C(t)$ as

$$K_C(t_{k-1}) \equiv \frac{K_{k-1}}{\Delta t} = P(t_{k-1})H_{k-1}^T R_C^{-1}(t_{k-1})\,.$$

Now as $\Delta t \to 0$ we have $P_{k-1}(-) \approx P_{k-1}(+)$ then the fifth $F_{k-1}K_{k-1}H_{k-1}P_{k-1}(-)$ and sixths terms $K_{k-1}H_{k-1}P_{k-1}(-)F_{k-1}^T$ on the right-hand-side of Equation 170 vanish and we are left with

$$\begin{aligned}
\dot{P}(t) &= F(t)P(t) + P(t)F(t)^T + L(t)Q_C(t)L(t)^T - K_C(t)H(t)P(t) \\
&= F(t)P(t) + P(t)F(t)^T + L(t)Q_C(t)L(t)^T - P(t)H(t)^T R_C(t)^{-1}H(t)P(t)\,, \qquad (172)
\end{aligned}$$

with the initial condition $P(0) = P_0$ for the continuous covariance estimation equation. Note we have used the definition of the continuous Kalman gain $K_C(t)$ given by

$$K_C(t) = P(t)H^T(t)R_C^{-1}(t)\,. \qquad (173)$$

To derive the continuous state estimation equation we start with

$$\hat{x}_k(+) = (\Phi_{k-1}\hat{x}_{k-1}(+) + \Gamma_{k-1}u_{k-1}) + K_k\left\{z_k - H_k[\Phi_{k-1}\hat{x}_{k-1}(+) + \Gamma_{k-1}u_{k-1}]\right\}\,. \qquad (174)$$

Then for small Δt we have $\Phi_{k-1} \approx I_n + F_{k-1}\Delta t$ so the above becomes

$$\begin{aligned}\hat{x}_k(+) &\approx \hat{x}_{k-1}(+) + \Delta t F_{k-1}\hat{x}_{k-1}(+) + \Gamma_{k-1}u_{k-1} \\ &\quad + K_k\left\{z_k - H_k\hat{x}_{k-1}(+) - \Delta t H_k F_{k-1}\hat{x}_{k-1}(+) - H_k\Gamma_{k-1}u_{k-1}\right\} .\end{aligned}$$

Recall that in going from the continuous system to the discrete system the product $\Gamma_{k-1}u_{k-1}$ can be approximated as

$$\begin{aligned}\Gamma_{k-1}u_{k-1} &\equiv \int_{t_{k-1}}^{t_k} \Phi(t_k,\tau)G(\tau)u(\tau)d\tau \approx \Phi(t_k,t_{k-1})G(t_{k-1})u(t_{k-1})\Delta t \\ &\approx (I + F_{k-1}\Delta t)G(t_{k-1})u(t_{k-1})\Delta t = (I + F_{k-1}\Delta t)G_{k-1}u_{k-1}\Delta t .\end{aligned}$$

So that just the matrix Γ_{k-1} can be represented with

$$\Gamma_{k-1} = (I + F_{k-1}\Delta t)G_{k-1}\Delta t = G_{k-1}\Delta t + O(\Delta t^2) .$$

Using this in the expression we have for $\hat{x}_k(+)$ we get

$$\begin{aligned}\hat{x}_k(+) &= \hat{x}_{k-1}(+) + [F_{k-1}\hat{x}_{k-1}(+) + G_{k-1}u_{k-1}]\Delta t \\ &\quad + K_k\left\{z_k - H_k[\hat{x}_{k-1}(+) + (F_{k-1}\hat{x}_{k-1}(+) + G_{k-1}u_{k-1})\Delta t]\right\} .\end{aligned} \tag{175}$$

Thus

$$\frac{\hat{x}_k(+) - \hat{x}_{k-1}(+)}{\Delta t} = F_{k-1}\hat{x}_{k-1}(+) + G_{k-1}u_{k-1} + \frac{K_k}{\Delta t}\left[z_k - \cdots\right] .$$

As $\Delta t \to 0$ we get

$$\frac{d\hat{x}(t)}{dt} = F(t)\hat{x}(t) + G(t)u(t) + P(t)H^T(t)R^{-1}(t)[z(t) - H(t)\hat{x}(t)] , \tag{176}$$

with $\hat{x}(0) = \hat{x}_0$.

Notes on alternative forms of the linear-optimal filter

In this section we will derive an alternative way to compute $K_C(t)$ rather than using Equations 172 and 173. We start with a decomposition of $P(t)$ in that we can write it in terms of two other matrices $\Lambda(t)$ and $X(t)$ as

$$P(t) = \Lambda(t)X^{-1}(t) . \tag{177}$$

This of means that

$$\Lambda(t) = P(t)X(t) . \tag{178}$$

Writing Equation 172 but expressed in terms of the matrices $\Lambda(t)$ and $X(t)$ we get

$$\dot{P} = F\Lambda X^{-1} + \Lambda X^{-1}F^T + LQ_CL^T - \Lambda X^{-1}H^TR_C^{-1}H\Lambda X^{-1} . \tag{179}$$

From Equation 177 the time derivative of $P(t)$ is given by

$$\dot{P} = \dot{\Lambda}X^{-1} - \Lambda X^{-1}\dot{X}X^{-1} .$$

so post-multiplying both sides by X gives

$$\dot{P}X = \dot{\Lambda} - \Lambda X^{-1}\dot{X} \tag{180}$$

We can also post-multiply Equation 179 by X to get another expression for $\dot{P}X$ namely

$$\dot{P}X = F\Lambda + \Lambda X^{-1}F^T X + LQ_C L^T X - \Lambda X^{-1}H^T R_C^{-1} H\Lambda \,.$$

We set this equal to the right-hand-side of Equation 180 we get

$$\dot{\Lambda} - \Lambda X^{-1}\dot{X} = F\Lambda + \Lambda X^{-1}F^T X + LQ_C L^T X - \Lambda X^{-1}H^T R_C^{-1} H\Lambda \,.$$

One way the above equation can hold true if we set $\dot{\Lambda}$ equal to the *first* and *third* terms on the right-hand-side or

$$\dot{\Lambda} = F\Lambda + LQ_C L^T X \,. \tag{181}$$

If this equation is to be true then we must set the second term $-\Lambda X^{-1}\dot{X}$ equal the *second* and *fourth* terms or

$$-\Lambda X^{-1}\dot{X} = \Lambda X^{-1}F^T X - \Lambda X^{-1}H^T R_C^{-1} H\Lambda \,.$$

If we pre-multiply by $-\Lambda^{-1}X$ we get

$$\dot{X} = -F^T X + H^T R_C^{-1} H\Lambda \,, \tag{182}$$

Combining Equations 181 and 182 into one system we have

$$\begin{bmatrix} \dot{\Lambda} \\ \dot{X} \end{bmatrix} = \begin{bmatrix} F & LQ_C L^T \\ H^T R_C^{-1} H & -F^T \end{bmatrix} \begin{bmatrix} \Lambda \\ X \end{bmatrix} \equiv A(t) \begin{bmatrix} \Lambda \\ X \end{bmatrix} . \tag{183}$$

Where we have defined the matrix A in the above expression. To determine initial conditions for this larger system, since $\Lambda(t) = P(t)X(t)$ when $t = 0$ we have $\Lambda(0) = P(0)X(0)$ so if we take the initial conditions for X to be $X(0) = I_n$ then we must have $\Lambda(0) = P(0)$. By solving this larger system for the vector $\begin{bmatrix} \Lambda \\ X \end{bmatrix}$ we can compute $P(t)$ for all time.

If all of the matrices F, L, Q_C, H, R_C found in Equation 183 are *constant* then we can compute the state transition matrix $\Theta(\Delta t)$ as

$$\Theta(\Delta t) = e^{A\Delta t} = \begin{bmatrix} \Theta_{11} & \Theta_{12} \\ \Theta_{21} & \Theta_{22} \end{bmatrix} .$$

The state transition matrix, $e^{A\Delta t}$, by definition satisfies

$$\begin{bmatrix} \Lambda(t+\Delta t) \\ X(t+\Delta t) \end{bmatrix} = \begin{bmatrix} \Theta_{11} & \Theta_{12} \\ \Theta_{21} & \Theta_{22} \end{bmatrix} \begin{bmatrix} \Lambda(t) \\ X(t) \end{bmatrix} .$$

or in component form

$$\Lambda(t+\Delta t) = \Theta_{11}\Lambda(t) + \Theta_{12}X(t) \tag{184}$$

$$X(t+\Delta t) = \Theta_{21}\Lambda(t) + \Theta_{22}X(t) \,. \tag{185}$$

Since all of the system matrices F, L, Q_C, H, R_C time independent the matrices Θ_{ij} are also time independent (constant). Using Equation 178 evaluated at $t + \Delta t$ or $\Lambda(t + \Delta t) = P(t + \Delta t)X(t + \Delta t)$ in the left-hand-side of Equation 184 we get

$$P(t + \Delta t)X(t + \Delta t) = \Theta_{11}\Lambda(t) + \Theta_{12}X(t)\,.$$

Then putting Equation 185 into $X(t + \Delta t)$ on the left-hand-side and $\Lambda(t) = P(t)X(t)$ into the right-hand-side gives

$$P(t + \Delta t)\left[\Theta_{21}\Lambda(t) + \Theta_{22}X(t)\right] = \Theta_{11}P(t)X(t) + \Theta_{12}X(t)\,.$$

Post-multiply this expression by $X(t)^{-1}$ to get

$$P(t + \Delta t)\left[\Theta_{21}\Lambda(t)X(t)^{-1} + \Theta_{22}\right] = \Theta_{11}P(t) + \Theta_{12}\,.$$

or solving for $P(t + \Delta t)$ we get

$$P(t + \Delta t) = \left[\Theta_{11}P(t) + \Theta_{12}\right]\left[\Theta_{21}P(t) + \Theta_{22}\right]^{-1}\,. \tag{186}$$

Which is a recursive relationship one could use to compute $P(t)$ starting at $P(0) = P_0$.

Notes on the Chandrasekhar Type Algorithms

This sequence of algorithms provides an alternative to finding the expression for $K_C(t)$ that can be used when all of the system matrices are time independent. If we assume there is no control $u(t)$ then the continuous state propagation equation is given by

$$\dot{\hat{x}} = F\hat{x}(t) + K_C(t)[z(t) - H\hat{x}(t)] = [F - K_C(t)H]\hat{x}(t) + K_C(t)z(t)\,, \tag{187}$$

with $\hat{x}(0) = x_0$ as the initial condition. This means that the state propagation matrix $\Phi(t, 0)$ for our estimate must satisfy

$$\dot{\Phi}(t, 0) = [F - K_C(t)H]\Phi(t, 0) \quad \text{with} \quad \Phi(0, 0) = I_n\,. \tag{188}$$

This result will be used below. Taking the time derivative of Equation 172 and using the fact that F, L, Q_C, H, R_C are constant we have

$$\ddot{P} = F\dot{P} + \dot{P}F^T - \dot{P}H^T R^{-1}HP - PH^T R^{-1}H\dot{P}\,, \tag{189}$$

Since $K_C(t) = P(t)H^T R_C^{-1}$ the above is

$$\begin{aligned}\ddot{P} &= F\dot{P} + \dot{P}F^T - \dot{P}H^T K_C(t)^T - K_C(t)H\dot{P} \\ &= [F - K_C(t)H]\dot{P} + \dot{P}[F - K_C(t)]^T\,.\end{aligned} \tag{190}$$

In the same way that $x(t) = \Phi(t, 0)x(0)$ the value of the time derivative of the covariance matrix $\dot{P}(t)$ can be computed from its initial condition $\dot{P}(0)$ using the state transition matrix $\Phi(t, 0)$ as follows

$$\dot{P}(t) = \Phi(t, 0)\dot{P}(0)\Phi(t, 0)^T\,. \tag{191}$$

Next using Equation 172 evaluated at $t = 0$ we have

$$\dot{P}(0) = FP(0) + P(0)F^T + LQ_CL^T - P(0)H^TR_C^{-1}HP(0) \equiv D\,,$$

where we have defined the symmetric matrix D. Using the LDU decomposition we can write D as a product like

$$D = [M_1 M_2]S[M_1^T M_2^T] = M_1M_1^T - M_2M_2^T\,,$$

where S is a special diagonal matrix, with elements ± 1, as

$$S = \text{Diag}[1, 1, \cdots, 1, -1, -1, \cdots, -1]\,.$$

Let the number of positive ones initially be denoted as β which also equals the number of positive eigenvalues of D. The number of negative ones is then $\alpha - \beta$ where α is the rank of D. With this special difference for $\dot{P}(0)$ using Equation 191 we see that $\dot{P}(t)$ is given by

$$\begin{aligned}\dot{P}(t) &= \Phi(t,0)\dot{P}(0)\Phi(t,0)^T = \Phi(t,0)[M_1M_1^T - M_2M_2^T]\Phi(t,0)^T \\ &= \Phi(t,0)M_1M_1^T\Phi(t,0)^T - \Phi(t,0)M_2M_2^T\Phi(t,0)^T \\ &\equiv Y_1(t)Y_1(t) - Y_2(t)Y_2(t)\,, \end{aligned} \tag{192}$$

where we have defined $Y_1(t)$ and $Y_2(t)$ as

$$\begin{aligned} Y_1(t) &\equiv \Phi(t,0)M_1 = \Phi(t,0)Y_1(0) \\ Y_2(t) &\equiv \Phi(t,0)M_2 = \Phi(t,0)Y_2(0)\,. \end{aligned}$$

Taking the derivative of these two expressions and using Equation 188 for $\dot{\Phi}(t,0)$ gives

$$\dot{Y}_1(t) = [F - K_C(t)H]\Phi(t,0)Y_1(0) = [F - K_C(t)H]Y_1(t) \quad \text{with} \quad Y_1(0) = M_1 \tag{193}$$

$$\dot{Y}_2(t) = [F - K_C(t)H]\Phi(t,0)Y_2(0) = [F - K_C(t)H]Y_2(t) \quad \text{with} \quad Y_1(0) = M_2\,. \tag{194}$$

Taking the time derivative of $K_C(t) = P(t)HR_C^{-1}$ and using Equation 192 gives

$$\dot{K}_C(t) = \dot{P}(t)H^TR_C^{-1} = [Y_1(t)Y_1(t)^T - Y_2(t)Y_2(t)^T]H^TR_C^{-1}\,. \tag{195}$$

with the initial condition $K_C(0) = P(0)HR_C^{-1}$. This is a differential equation for $K_C(t)$ which depends on $Y_1(t)$ and $Y_2(t)$ both of which have differential equations of their own (given by Equation 193 and 194). A Chandrasekhar type algorithm is then to solve the differential Equations 193, 194 and 195 with their various initial conditions.

Notes on the square-root formulation

In the case when the state covariance matrix $P(t)$ is ill-conditioned (has eigenvalues of very different magnitudes) we can introduce "square-root filtering" which is a more numerically stable way to compute $P(t)$. We start with the Cholesky decomposition of $P(t)$ as $P(t) = S(t)S(t)^T$ where $S(t)$ is an upper triangular matrix. Then the time-derivative of P is given by

$$\dot{P} = \dot{S}S^T + S\dot{S}^T\,.$$

Using Equation 172 and our expression for P in terms of S we get

$$\dot{S}S^T + S\dot{S}^T = FSS^T + SS^TF^T + LQ_CL^T - SS^TH^TR_C^{-1}HSS^T\,. \tag{196}$$

Now premultiply by S^{-1} and postmultiply by S^{-T} this expression to get

$$S^{-1}\dot{S} + \dot{S}^TS^{-T} = S^{-1}FS + S^TF^TS^{-T} + S^{-1}LQ_CL^TS^{-T} - S^TH^TR_C^{-1}HS\,. \tag{197}$$

Since S is upper triangular so are S^{-1} and $S^{-1}\dot{S}$. In the same way $\dot{S}^TS^{-T}$ is lower triangular. Thus the left-hand-side is the sum of two matrices the first one is upper triangular and the second one is lower triangular. If we do the same thing with the right-hand-side of the above expression in that after we sum everything into one matrix RHS $= M(t)$, we split this matrix up into its upper and a lower triangular parts as $M(t) = M_{\rm LT}(t) + M_{\rm UT}(t)$ or

$$(m_{ij})_{\rm UT} = \begin{cases} m_{ij} & i < j \\ \frac{1}{2}m_{ij} & i = j \\ 0 & i > j \end{cases}\,.$$

We can then equate the upper triangular parts of both sides to solve

$$S^{-1}\dot{S} = M_{\rm UT} \quad \text{so} \quad \dot{S} = M_{\rm UT}S\,.$$

This last differential equation would have to be solved for $S(t)$ with the initial conditions on S related to those of P in $S(0)S^T(0) = P(0)$. Once one has $S(t)$ one can compute $P(t)$ or $K_C(t) = S(t)S(t)^TH(t)^TR_C(t)^{-1}$ as needed.

Notes on the correlation in disturbance inputs and measurement noise

In this section of these notes we extend the Kalman-Bucy filter to include cases where we have correlations in disturbance inputs and measurement noise. We derive the continuous time results by studying the corresponding discrete time results obtained earlier. We begin by considering the case where we have cross-correlated disturbance inputs and measurement noise. First consider Equation 163 evaluated at $k \to k-1$ which is

$$P_{k-1}(+) = P_{k-1}(-) - K_{k-1}[M_{k-1}^T + H_{k-1}P_{k-1}(-)]\,. \tag{198}$$

Next Equation 138 with Λ the identity matrix where we have

$$\begin{aligned} P_k(-) &= \Phi_{k-1}P_{k-1}(+)\Phi_{k-1}^T + Q_{k-1} \\ &\approx (I + F_{k-1}\Delta t)P_{k-1}(+)(I + F_{k-1}\Delta t)^T + Q_{k-1}\,. \end{aligned} \tag{199}$$

Using Equation 198 into the right-hand-side of Equation 199 to get for $P_k(-)$

$$\begin{aligned} &= (I + F_{k-1}\Delta t)[P_{k-1}(-) - K_{k-1}(M_{k-1}^T + H_{k-1}P_{k-1}(-))](I + F_{k-1}\Delta t)^T + Q_{k-1} \\ &= [P_{k-1}(-) - K_{k-1}(M_{k-1}^T + H_{k-1}P_{k-1}(-)) + \Delta tF_{k-1}P_{k-1}(-) - \Delta tF_{k-1}K_{k-1}(M_{k-1}^T + H_{k-1}P_{k-1}(-))] \\ &\times (I + F_{k-1}\Delta t)^T + Q_{k-1} \\ &= P_{k-1}(-) - K_{k-1}(M_{k-1}^T + H_{k-1}P_{k-1}(-)) + \Delta tF_{k-1}P_{k-1}(-) - \Delta tF_{k-1}K_{k-1}(M_{k-1}^T + H_{k-1}P_{k-1}(-)) \\ &+ P_{k-1}(-)F_{k-1}^T\Delta t - K_{k-1}(M_{k-1}^T + H_{k-1}P_{k-1}(-))F_{k-1}^T\Delta t \\ &+ \Delta t^2F_{k-1}P_{k-1}(-)F_{k-1}^T - \Delta t^2F_{k-1}K_{k-1}(M_{k-1}^T + H_{k-1}P_{k-1}(-))F_{k-1}^T + Q_{k-1}\,. \end{aligned}$$

From this the first difference of $P_k(-)$ is given by

$$\begin{aligned}\frac{P_k(-)-P_{k-1}(-)}{\Delta t} &= F_{k-1}P_{k-1}(-)+P_k(-)F_{k-1}^T+\frac{1}{\Delta t}Q_{k-1}\\ &-\frac{K_{k-1}}{\Delta t}(M_{k-1}^T+H_{k-1}P_{k-1}(-))-F_{k-1}K_{k-1}(M_{k-1}^T+H_{k-1}P_{k-1}(-))\\ &-K_{k-1}(M_{k-1}^T+H_{k-1}P_{k-1}(-))F_{k-1}^T\\ &+\Delta t F_{k-1}P_{k-1}(-)F_{k-1}^T-\Delta t F_{k-1}K_{k-1}(M_{k-1}^T+H_{k-1}P_{k-1}(-))F_{k-1}^T\end{aligned}$$

Changing the order of some of the terms above gives

$$\begin{aligned}&= F_{k-1}P_{k-1}(-)+P_k(-)F_{k-1}^T+F_{k-1}P_{k-1}(-)F_{k-1}^T\Delta t+\frac{Q_{k-1}}{\Delta t}\\ &-\frac{K_{k-1}}{\Delta t}M_{k-1}^T-K_{k-1}M_{k-1}^TF_{k-1}^T-\frac{K_{k-1}}{\Delta t}H_{k-1}P_{k-1}(-)-K_{k-1}H_{k-1}P_{k-1}(-)F_{k-1}^T\\ &-F_{k-1}K_{k-1}M_{k-1}^T-F_{k-1}K_{k-1}M_{k-1}^TF_{k-1}^T\Delta t\\ &-F_{k-1}K_{k-1}H_{k-1}P_{k-1}(-)-F_{k-1}K_{k-1}H_{k-1}P_{k-1}(-)F_{k-1}^T\Delta t\,. \qquad (200)\end{aligned}$$

We will now evaluate various terms in the above in the limit as $\Delta t\to 0$.

To start we will evaluate the expression M_{k-1} in this limit. Note that in the continuous filtering problem we are considering here

$$\dot{x}(t)=F(t)x(t)+G(t)u(t)+L(t)w(t)\,,$$

where we have included an $L(t)$ matrix, when we turn this into the model used in the section on *discrete* cross-correlated disturbance inputs and measurement noise

$$x_k=\Phi_{k-1}x_{k-1}+\Gamma_{k-1}u_{3k-1}+w_{k-1}\,,$$

note that here there is no Λ_{k-1} coefficient in front of the discrete process noise w_{k-1} or $\Lambda_{k-1}=I$. Using Equation 142 we have the discrete process noise in terms of the continuous system given by

$$w_{k-1}\approx\int_{t_{k-1}}^{t_k}\Phi(t_k,\tau)L(\tau)w(\tau)d\tau\approx\Phi(t_k,t_{k-1})L(t_{k-1})w(t_{k-1})\Delta t=L(t_{k-1})w(t_{k-1})\Delta t+O(\Delta t^2)\,.$$

Using this and Equation 171 to argue that under squared expectation $n_{k-1}\approx\frac{R_C(t_{k-1})}{\Delta t}$ we are now ready to derive the limit of M_{k-1} as $\Delta t\to 0$. In this case as $M_{k-1}=E[w_{k-2}n_{k-1}^T]$ and using the above two components we find

$$\lim_{\Delta t\to 0}M_{k-1}\approx E\left[L(t_{k-2})w(t_{k-2})\Delta t\left(\frac{R_C(t_{k-1})}{\Delta t}\right)^T\right]=L(t)E[w(t)n(t)^T]=L(t)M_C(t)\,,$$

a limit independent of Δt.

We will now evaluate the limit of K_{k-1} as $\Delta t\to 0$. To do this recall Equation 162 for K_k now evaluated at $k-1$ which gives

$$K_{k-1}=[P_{k-1}(-)H_{k-1}^T+M_{k-1}][H_{k-1}P_{k-1}(-)H_{k-1}^T+H_{k-1}M_{k-1}+M_{k-1}^TH_{k-1}^T+R_{k-1}]^{-1}\,.$$

Using Equation 171 the second factor in the expression for K_{k-1} is given by

$$\left[H_{k-1}P_{k-1}(-)H_{k-1}^T + H_{k-1}M_{k-1} + M_{k-1}^T H_{k-1} + \frac{1}{\Delta t}R(t_{k-1})\right]^{-1},$$

Factoring out Δt we get

$$\Delta t\left[(H_{k-1}P_{k-1}(-)H_{k-1}^T + H_{k-1}M_{k-1} + M_{k-1}^T H_{k-1})\Delta t + R(t_{k-1})\right]^{-1},$$

which as $\Delta t \to 0$ gives $\Delta t R_C(t_{k-1})^{-1}$ to first order. Combining what we have seen thus far we have shown that

$$K_C(t) \equiv \lim_{\Delta t\to 0}\frac{K_{k-1}}{\Delta t} = [P(t)H^T(t) + L(t)M_C(t)]R_C^{-1}(t)\,. \tag{201}$$

Thus for the differential equation for $P(t)$ from Equation 200 we find

$$\begin{aligned}\dot{P}(t) &= FP + PF^T + LQ_CL^T - K_C(t)[M_C(t)^T L(t)^T + H(t)P(t)] \\ &= FP + PF^T + LQ_CL^T \\ &- [P(t)H^T(t) + L(t)M_C(t)]R_C^{-1}[M_C(t)^T L(t)^T + H(t)P(t)]\,.\end{aligned} \tag{202}$$

The desired expression.

Notes on continuous time-correlated measurement error

In this section we treat the problem where we have time-correlated measurement errors. We begin by assuming a model for their time correlation

$$\dot{n}(t) = N(t)n(t) + \nu(t)\,, \tag{203}$$

and then introduce the derived measurement $\zeta(t)$

$$\zeta(t) = \dot{z}(t) - N(t)z(t)\,. \tag{204}$$

From the measurement equation $z(t) = H(t)x(t) + n(t)$ we find its time derivative given by

$$\dot{z}(t) = \dot{H}(t)x(t) + H(t)\dot{x}(t) + \dot{n}(t)\,,$$

thus using this and the model for the measurement noise Equation 203 $\zeta(t)$ can be written

$$\begin{aligned}\zeta(t) &= \dot{H}x + H\dot{x} + \dot{n} - N(Hx + n) = \dot{H}x + H(Fx + Lw) + \dot{n} - NHx - Nn \\ &= (\dot{H} + HF - NH)x + HLw + \nu(t)\,,\end{aligned}$$

since $-Nn + \dot{n} = \nu(t)$ and $\dot{x} = Fx + Gw$. Lets introduce the functions $D(t)$ and $\eta(t)$ in the mapping of the state $x(t)$ into the measurement $\zeta(t)$ (the measurement equation) as

$$\begin{aligned}D(t) &= \dot{H}(t) + H(t)F(t) - N(t)H(t) \\ \eta(t) &= H(t)L(t)w(t) + \nu(t)\,.\end{aligned}$$

Then this system with state $x(t)$ and measurement $\zeta(t)$ has *cross-correlated* measurement and disturbance noise. Thus the results just derived can be used to filter this signal. The cross-correlated component is given by

$$\begin{aligned} M_{CC}(t) \equiv E[w(t)\eta(\tau)^T] &= E[w(t)(H(\tau)L(\tau)w(\tau)+\nu(\tau))^T] \\ &= E[w(t)w(\tau)]L^T(\tau)H^T(\tau) + E[w(t)\nu(\tau)^T] \\ &= Q_C(t)L^T(t)H^T(t)\delta(t-\tau)\,. \end{aligned}$$

since we assume that $E[w(t)\nu(\tau)^T] = 0$. This new systems measurement spectral density $R_{CC}(t)$ is given by

$$\begin{aligned} R_{CC}(t) \equiv E[\eta(t)\eta(\tau)^T] &= H(t)L(t)E[w(t)w(\tau)^T]L^T(\tau)H^T(\tau) \\ &\quad + H(t)L(t)E[w(t)\nu(\tau)^T] + E[\nu(t)w(\tau)^T]L^T(\tau)H^T(\tau) \\ &\quad + E[\nu(t)\nu(\tau)^T] \\ &= [H(t)L(t)Q_C(t)L^T(t)H^T(t) + V_C(t)]\delta(t-\tau)\,. \end{aligned} \tag{205}$$

With these definitions for M_{CC}, R_{CC} and taking $Q_{CC} = Q_C$ (the typical process noise spectral density function) we can compute $K_C(t)$ and $P(t)$ for this problem by using Equations 201 and 202 but with the substitutions

$$M_C \to M_{CC}\,, \quad R_C \to R_{CC}\,, \quad Q_C \to Q_{CC}\,, \quad \text{and} \quad H \to D\,.$$

To compute the estimate of the state $\hat{x}(t)$ we integrate the ordinary differential equation

$$\begin{aligned} \dot{\hat{x}}(t) &= F(t)\hat{x}(t) + K_C(t)[\zeta(t) - D(t)\hat{x}(t)] \\ &= F(t)\hat{x}(t) + K_C(t)[\dot{z}(t) - N(t)z(t) - D(t)\hat{x}(t)]\,. \end{aligned} \tag{206}$$

As seen in Equation 204 the measurement $\zeta(t)$ depends on $\dot{z}$ which might not be desirable to compute directly from $z(t)$ since $z(t)$ contains noise which might make the expression for $\dot{z}$ relatively unstable. Note that we can get an expression with $\dot{z}(t)$ as in Equation 206 as

$$K_C(t)\dot{z}(t) = \frac{d}{dt}[K_C(t)z(t)] - \dot{K}_C(t)z(t)\,. \tag{207}$$

If we put this into the second term on the right-hand-side of Equation 206 and then bring $\frac{d}{dt}[K_C(t)z(t)]$ over to the left-hand-side we get

$$\frac{d}{dt}\hat{x} - \frac{d}{dt}[K_C(t)z(t)] = F(t)\hat{x}(t) - \dot{K}_C(t)z(t) - K_C(t)[N(t)z(t) + D(t)\hat{x}(t)]\,,$$

or

$$\frac{d}{dt}[\hat{x}(t) - K_C(t)z(t)] = [F(t) - K_C(t)D(t)]\hat{x}(t) - [K_C(t)N(t) + \dot{K}_C(t)]z(t)\,. \tag{208}$$

Introduce $\hat{\chi}(t)$ so that

$$\hat{\chi}(t) \equiv \hat{x}(t) - K_C(t)z(t)\,,$$

which has the initial conditions $\hat{\chi}(0) = \hat{x}(0) - K_C(0)z(0)$. Thus for state estimation we will integrate Equation 208 which has $\dot{K}_C(t)$ rather than $\dot{z}(t)$. The the expression $\dot{K}_C(t)$ is found from the expression obtained via integration for $K_C(t)$ which hopefully is more stable than taking the derivative of $z(t)$. Now $\dot{K}_C(t)$ needed above is easy to calculate if H and R_C are constant for then

$$\dot{K}_C(t) = \dot{P}(t)H^T R_C^{-1}\,,$$

and $\dot{P}(t)$ is given by the right-hand-side of Equation 202.

Notes on Quasilinear Filter

We want to minimize the distance between $f(x)$ and $a_0 + a_1(x - x_0) = a_0 + a_1\tilde{x}$, where $\tilde{x} \equiv x - x_0$. To do this we consider the cost function $J(a_0, a_1)$ given by

$$J(a_0, a_1) = E\{[f(x) - a_0 - a_1\tilde{x}]^2\}\,.$$

Then the needed derivatives of J become

$$\frac{\partial J}{\partial a_0} = E\{2[f(x) - a_0 - a_1\tilde{x}](-1)\} = 0\,,$$

or

$$a_0 = E[f(x)] - a_1 E[\tilde{x}]\,. \tag{209}$$

and

$$\frac{\partial J}{\partial a_1} = E[2(f(x) - a_0 - a_1\tilde{x})\tilde{x}(-1)] = 0\,,$$

or

$$E[f(x)\tilde{x}] - a_0 E[\tilde{x}] - a_1 E[\tilde{x}^2] = 0\,.$$

As a system these two equation are

$$\begin{aligned} a_0 + E[\tilde{x}]a_1 &= E[f(x)] \\ E[\tilde{x}]a_0 + E[\tilde{x}^2]a_1 &= E[f(x)\tilde{x}]\,. \end{aligned}$$

We can solve for a_0 and a_1 using Crammers rule. We need

$$D = \begin{vmatrix} 1 & E[\tilde{x}] \\ E[\tilde{x}] & E[\tilde{x}^2] \end{vmatrix} = E[\tilde{x}^2] - E[\tilde{x}]^2\,.$$

Then we have

$$\begin{aligned} a_0 &= \frac{1}{D}\begin{vmatrix} E[f] & E[\tilde{x}] \\ E[f\tilde{x}] & E[\tilde{x}^2] \end{vmatrix} = \frac{E[f]E[\tilde{x}^2] - E[\tilde{x}]E[f\tilde{x}]}{E[\tilde{x}]^2 - E[\tilde{x}^2]} \\ a_1 &= \frac{1}{D}\begin{vmatrix} 1 & E[f] \\ E[\tilde{x}] & E[f\tilde{x}] \end{vmatrix} = \frac{E[f\tilde{x}] - E[\tilde{x}]E[f]}{E[\tilde{x}]^2 - E[\tilde{x}^2]}\,. \end{aligned}$$

Warning: This solution is somewhat different than the what the book presents, but I don't see any problems with the above derivation. If anyone sees anything wrong with what I have done please contact me. If $E[\tilde{x}] = 0$ and $E[f] = 0$ then we get

$$a_0 = 0 \quad \text{and} \quad a_1 = \frac{E[f(x)\tilde{x}]}{E[\tilde{x}^2]}\,.$$

We will now consider the multidimensional generalization of the above approximation. We will approximate $\mathbf{f}(\mathbf{x})$ using the Taylor series like approximation given by

$$\mathbf{f}(\mathbf{x}) \approx b + D(x - \hat{x})\,,$$

where $\hat{x}$ is some "centering value" (often taken to be the mean of x). If we assume that $\mathbf{f}$ is a $r \times 1$ vector and $\mathbf{x}$ is a $n \times 1$ vector then b is an $r \times 1$ bias vector and D is an $r \times n$ matrix. With this approximation we form the objective function J we want to minimize given by

$$J = E[(f(x) - b - D(x - \hat{x}))^T (f(x) - b - D(x - \hat{x}))] \,,$$

and we need to find the minimum of the above expression as a function of b and D. Taking the partial derivative of this expression with respect to b we get

$$\frac{\partial J}{\partial b} = 2E[f(x) - b - D\tilde{x}] = 0 \,,$$

or

$$E[f(x)] - b - DE[\tilde{x}] = 0 \,.$$

If we take $\hat{x}$ to be the mean of x then $E[\tilde{x}] = 0$ and we have

$$b = E[f(x)] \,. \tag{210}$$

With b specified we now minimize J with respect to D. To do this we need to compute $\frac{\partial J}{\partial D}$. Taking the D derivative of the above expression is made easier if we write J as

$$\begin{aligned} J &= E[(f-b)^T(f-b)] - E[(f-b)^T D(x-\hat{x})] \\ &- E[(x-\hat{x})^T D^T (f-b)] + E[(x-\hat{x})^T D^T D(f-b)] \,. \end{aligned}$$

To evaluate the matrix derivatives we recall

$$\frac{\partial}{\partial \mathbf{X}}(\mathbf{a}^T \mathbf{X} \mathbf{b}) = \mathbf{a}\mathbf{b}^T \,, \tag{211}$$

and

$$\frac{\partial}{\partial \mathbf{X}}(\mathbf{a}^T \mathbf{X}^T \mathbf{b}) = \mathbf{b}\mathbf{a}^T \,. \tag{212}$$

To evaluate the fourth term which has a product of $D^T D$ we use the product rule with the above identities. First taking $a = (x - \hat{x})$ and $b = D(x - \hat{x})$ and the taking $a = (x - \hat{x})D^T$ and $b = (x - \hat{x})$. With these we find

$$\begin{aligned} \frac{\partial J}{\partial D} &= -E[(f-b)(x-\hat{x})^T] - E[(f-b)(x-\hat{x})^T] \\ &+ E[D(x-\hat{x})(x-\hat{x})^T] + E[D(x-\hat{x})(x-\hat{x})^T] \\ &= -2E[(f-b)(x-\hat{x})^T] + 2DE[(x-\hat{x})(x-\hat{x})^T] \,. \end{aligned}$$

When we set this last expression equal to zero and solve for D we find

$$D = E[(f-b)(x-\hat{x})^T]E[(x-\hat{x})(x-\hat{x})^T]^{-1} \,. \tag{213}$$

Introducing $P = E[(x - \hat{x})(x - \hat{x})^T]$ and noting that

$$\begin{aligned} E[(f-b)(x-\hat{x})^T] &= E[fx^T] - E[f\hat{x}^T] - E[bx^T] + E[b\hat{x}^T] \\ &= E[fx^T] - b\hat{x}^T - b\hat{x}^T + b\hat{x}^T \\ &= E[fx^T] - bE[x]^T \,, \end{aligned}$$

so when we replace $b = E[f]$ we get

$$D = (E[fx^T] - E[f]E[x]^T)P^{-1} \,. \tag{214}$$

This result is some what different than that presented in the book but it matches that found in [1] so I believe it is correct.

Example 4.7-1 the extended Kalman filter to estimate a random constant

For this example our *true* system state is given by the weathervane example where the dynamics are

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\omega_n^2 & -2\zeta\omega_n \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ \omega_n^2 \end{bmatrix} w \,. \tag{215}$$

We specify *truth* parameters $\zeta = 0.1$ and $\omega_n = 2$ radians/second and take a measurement equation given by

$$\begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} n_1 \\ n_2 \end{bmatrix} . \tag{216}$$

We simulated a system like this for Example 4.3-1 on Page 96.

Let $Q_C = E[w^2] = q = 1000$. We assume that the estimate of ω_n is *unknown* and thus introduce an additional state component denoted a and defined by

$$a = -\omega_n^2 \,.$$

Since the weathervane dynamics depends on the parameter $-2\zeta\omega_n$ we technically don't know this value either. For the ease of this example we will *assume* that we know its value and take

$$b = -2\zeta\omega_n = -2(0.1)(2) = -0.4 \,.$$

Then since we would like to estimate a, a constant, its dynamics are $\dot{a} = 0$. The fully augmented (with the additional state element a) system dynamics looks like

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{a} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ a & b & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ a \end{bmatrix} + \begin{bmatrix} 0 \\ -a \\ 0 \end{bmatrix} w \,. \tag{217}$$

To "solve" this, we will use the "Hybrid Extended Kalman" filter. To duplicate the simulation presented in the book we will generate data and measurements from Equation 215 and Equation 216. Then starting at $t_0 = 0$ and for 20 seconds with a fixed time step we will start with given estimates of the state components ($\hat{x}_1(0)$, $\hat{x}_2(0)$, $\hat{a}(0)$) and estimates of their uncertainties ($p_{11}(0)$, $p_{22}(0)$, $p_{33}(0)$). This information corresponds to what we know at the time step index $k = 0$. Then we iterate for $k = 1, 2, 3, \cdots$ the following

- First, state estimation propagation to get the state $\hat{x}[t_k(-)]$. We do this by numerically solving the coupled Equations 217 (with no noise $w = 0$) to the time $t_k(-)$ starting with $x(0) = \hat{x}[t_{k-1}(+)]$. Second, state covariance propagation to get the matrix $P[t_k(-)]$. We do this by numerically solving the Riccati equation

  $$\dot{P}(t) = F_A P + P F_A^T + L_A Q_A L_A^T - P H_A^T R_A^{-1} H_A P \,,$$

  starting with $P(0) = P[t_{k-1}(+)]$ to the time $t_k(-)$. Here we have

  $$F_A = \begin{bmatrix} 0 & 1 & 0 \\ \hat{a} & b & \hat{x}_1 \\ 0 & 0 & 0 \end{bmatrix}, \; H_A = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}, \; L_A = \begin{bmatrix} 0 \\ -\hat{a} \\ 0 \end{bmatrix}, \; Q_A = 1000, \; R_A = \begin{bmatrix} 10 & 0 \\ 0 & 10 \end{bmatrix} .$$

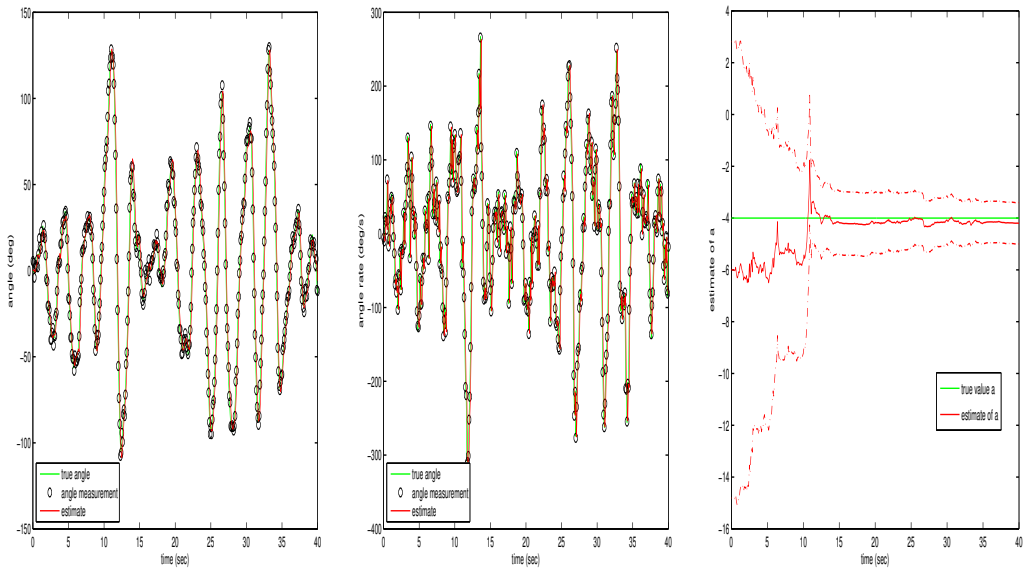

Figure 19: The Hybrid Extended Kalman filter state estimate for Example 4.7-1. **Left:** Estimates of x_1. **Center:** Estimates of x_2. **Right:** Estimates of $a = -\omega_n^2$ each with 95% error bounds.

- Given these new values of the state and covariance at the time $t_k(-)$ we perform the filter gain calculation

$$K(t_k) = P[t_k(-)]H^T(t_k(-))\left\{H(t_k(-))P[t_k(-)]H(t_k(-))^T + R(t_k(-))\right\}^{-1},$$

- Using this filter gain we update the state estimate

$$\hat{x}[t_k(+)] = \hat{x}[t_k(-)] + K(t_k)\left\{z_k - h[\hat{x}[t_k(-)], t_k(-)]\right\},$$

- and update the covariance estimation update

$$P[t_k(+)] = (I_n - K(t_k)H(t_k))P(t_k(-)),$$

  to get the information needed to process the next iteration.

Since we are assuming that we don't know the true value of the constant a we start with the incorrect value of $a(0) = -6.0$ (the true value is -4.0). It is amazing how well the system works at computing the correct value of a one should try to run the following codes with different (and incorrect) starting values of $\hat{a}(0)$ and observe the convergence of $\hat{a}$ to the correct value.

See the MATLAB codes `sect_7_dup_example_4_7_1.m`, `example_4_7_1_extended_kf.m`, and `example_4_7_1_extended_kf_fn.m` for an implementation of this problem. When these are run they product the plots given in Figure 19. The same qualitative comments the book makes apply here also.

Example 4.7-2 estimation of a constant parameter using parallel filters

In this section we duplicate the multiple model estimation results presented in the book. To that end we implement a function `weathervane_system.m` to compute Φ and Q for the weathervane example. For all values of ω_n^2 suggested by the book this routine gives the same Φ as presented in the book. For the value of $\omega_n^2 = 4.0$ this routine gives the same value of Q presented in the book. For the different values of ω_n^2 ($\neq 4$) and suggested in this section this routine gives a different expression for Q than presented in the book. For example, when we take $\omega_n = 4.4$ the routine `weathervane_system.m` gives

$$\Phi = \begin{bmatrix} 0.9784 & 0.0972 \\ -0.4277 & 0.9376 \end{bmatrix} \quad \text{and} \quad Q = \begin{bmatrix} 6.2 & 91.5 \\ 91.5 & 1830.3 \end{bmatrix}.$$

This makes me wonder if there is a typo in the presentation of Q_2 and Q_3 in the book. Proceeding by using the above MATLAB routine we implemented the multiple model application discussed in the book. I choose to consider *four* hypothesis for the value of ω_n^2 rather than three. These correspond to

$$\omega_n \in \{\sqrt{3.6}\,, 2\,, \sqrt{4.4}\,, \sqrt{4.8}\}\,.$$

The multiple mode is implemented in the MATLAB code `sect_7_dup_example_4_7_2.m`. For one random seed the running of the previous code gives the results shown in Figure 20.

Problem Solutions

Section 4.1 Problem 1 (least squares estimates to fit polynomial models)

This problem is worked in the MATLAB script `sect_1_1.m`

Section 4.1 Problem 2 (polynomial models with different variances)

This problem is worked in the MATLAB script `sect_1_2.m`

Section 4.1 Problem 3 (fitting polynomial models with one more data point)

This problem is worked in the MATLAB script `sect_1_3.m`

Section 4.1 Problem 5 (least square mapping of an input/output relationship)

For this problem we stack the individual matrices H "on top of each other" and then use the left pseudo inverse to produced the globally best estimate. This problem is worked in

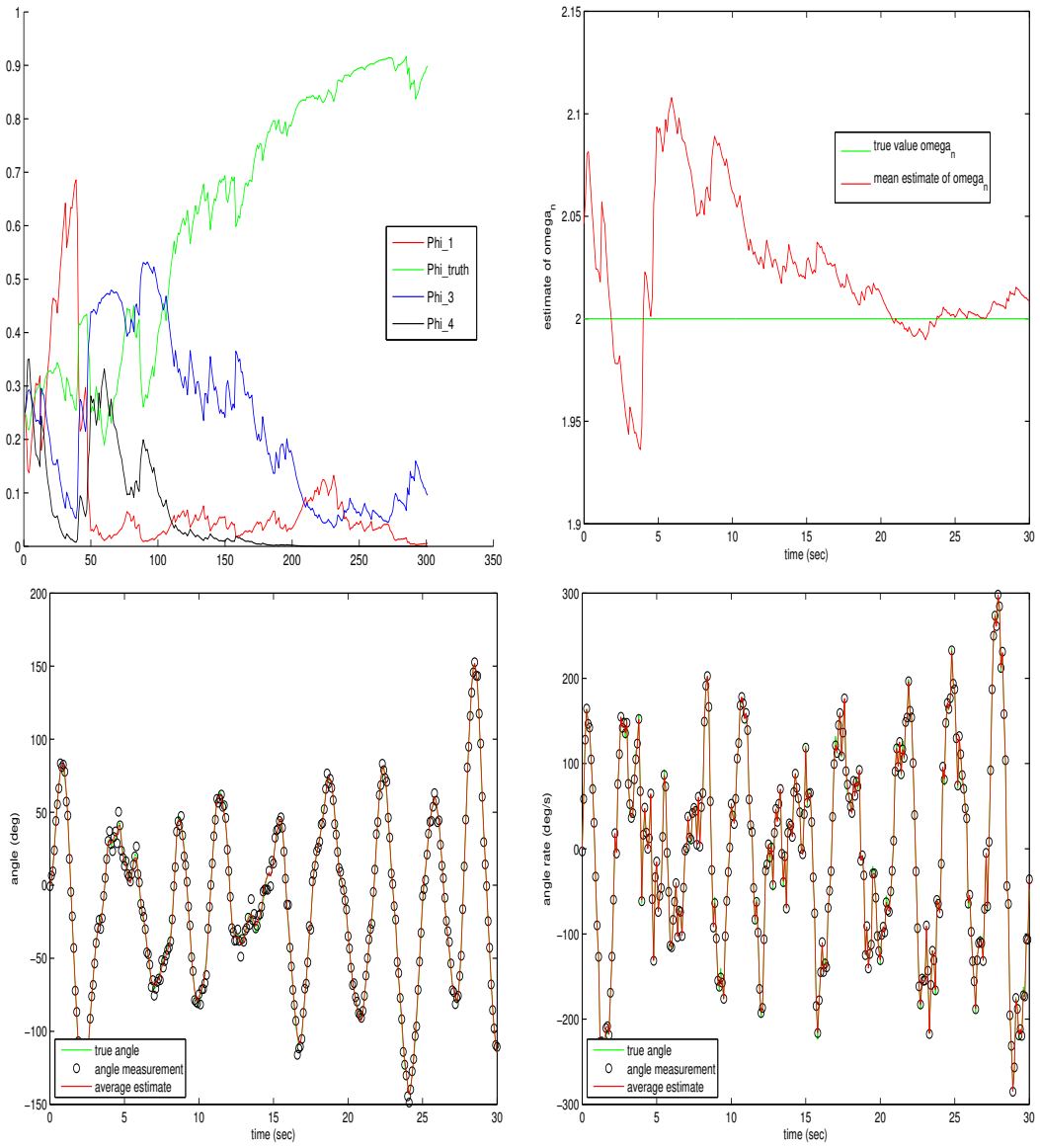

Figure 20: **Top Left:** Estimates of the probability of each model. The correct model (the green line) eventually asymptotes to 1. **Top Right:** Average estimates of the true value of ω_n using probabilities from each model. Again $\omega_n \to 2$ the truth value as we process more measurements. **Bottom:** Plots of x_1 and x_2 the truth and the estimate. One cannot tell the difference between the truth and the estimates.

the MATLAB script `sect_1_5.m`.

Section 4.1 Problem 6 (fitting a nonlinear input/output relationship)

Rather than implement the suggested Newton-Raphson minimization routine for the objective function

$$J(x_1, x_2) = \frac{1}{2}\sum_{i=1}^{10}\left\{(z_{i1} - x_1^2 - x_2 - 5)^2 + (z_{i2} - \frac{1}{4}x_2^3 - 4)^2\right\},$$

we instead use the built-in MATLAB function `fmins`. This problem is worked in the MATLAB script `sect_1_6.m`.

Section 4.2 Problem 1 (when is the expected state a constant)

The expected value of a state takes a constant value when there is no system dynamics $\Phi_{k-1} = 0$ and no process noise $w_{k-1} = 0$. The covariance matrix P_k can in general be a constant when Φ is time independent and non-zero, since constant steady state solutions to the covariance Kalman filtering equations exist in this case.

Section 4.2 Problem 2 (simulating a discrete-time system)

We are given a time-independent discrete dynamical system of the form

$$x_k = \Phi x_{k-1} + \Lambda w_{k-1},$$

with $\Phi = \begin{bmatrix} 1 & 1 \\ -1 & 0.4 \end{bmatrix}$ and $\Lambda = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$. We want to simulate this discrete dynamical system with $x_0 = 0$. We are told to assume that w_k is has a mean of zero and a variance of 1. In this case the state's mean value m_k propagates according to

$$m_k = \Phi m_{k-1}.$$

This equation is very easy to simulate since in the case when $m_0 = E[x_0] = 0$ we have $m_k = 0$ for all k. The states covariance P_k is updated using Equation 138. With this background this problem is worked in the MATLAB script `sect_2_prob_2.m`, where we simulate a true state trajectory (including noise) for x_k for 20 time steps (in green), plot the mean m_k (in red), and plot confidence intervals around the mean (in red). When we run that script we get the results shown in Figure 21.

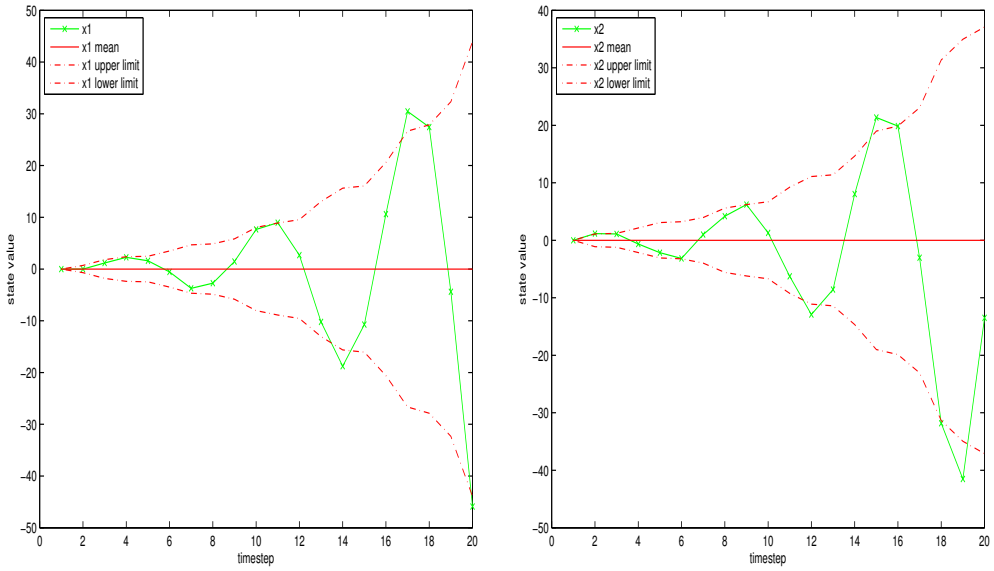

Figure 21: **Left:** The state component x_1, its mean and interval of uncertainty for 20 time steps. **Right:** The state component x_2, its mean and interval of uncertainty for 20 time steps.

Section 4.2 Problem 3 (computing sampled-data disturbance covariance)

For the given *continuous* system

$$\dot{x}(t) = Fx(t) + Lw(t)\,,$$

the *discrete* system is given by

$$x_k = \Phi_{k-1}x_{k-1} + \Lambda_{k-1}w_{k-1}\,,$$

when F and L are constants with Φ_{k-1} given by

$$\Phi_{k-1} = \Phi(\Delta t) = e^{F\Delta t} \approx I_n + \Delta t F = \begin{bmatrix} 1 & 1 \\ -1 & 0.4 \end{bmatrix}\,,$$

to first order and with $\Delta t = 1$. In addition, we have Λ_{k-1} given by

$$\Lambda_{k-1} = \Lambda(\Delta t) = \Phi(\Delta t)[I_n - \Phi^{-1}(\Delta t)]F^{-1}L\,.$$

The book does not say what order this expression is. If we want a first order approximation since $\Phi^{-1}(\Delta t) = e^{-F\Delta t} \approx I - F\Delta t$ we have

$$\Lambda_{k-1} = (I + F\Delta t)[I - (I - F\Delta t)]F^{-1}L = \Delta t(I + F\Delta t)L = \Delta t L\,.$$

If $\Delta t = 1$ we have $\Lambda_{k-1} = L = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$ and our discrete approximation is then

$$x_k = \begin{bmatrix} 1 & 1 \\ -1 & 0.4 \end{bmatrix} x_{k-1} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} w_{k-1}\,.$$

This verifies the discrete dynamical system presented in the previous problem. To compute the sampled-data disturbance matrix Q_{k-1} we use Equation 144 which is discussed on Page 92. We compute this in the MATLAB file `sect_2_prob_3.m`, where we use the routine `example_4_2_2_compute_qkm1.m`. When we run that script we get the sampled-data disturbance covariance Q_{k-1} given by

$$Q_{k-1} = \begin{bmatrix} 0.1801 & 0.2006 \\ 0.2006 & 0.4515 \end{bmatrix} .$$

Section 4.2 Problem 4 (a continuous system)

One way to view this problem would be to note that if we sample this system to first order with $\Delta t = 1$ we have the discrete system given by Problem (2) in this section. This later system is simulated there. As an alternative we could use a higher order Runge-Kutta algorithm to generate $x(t_k)$ for $0 \leq t_k \leq 20$, but the results would be qualitatively similar to that from Problem 2.

Section 4.2 Problem 5 (the Cholesky decomposition)

The Cholesky decomposition of Q is the factorization of Q as SS^T where S is lower triangular. This can be computed with the MATLAB command `chol`.

Section 4.3 Problem 1 (codes to perform discrete time Kalman filtering)

Rather than implement a Kalman filter ourselves we will use the MATLAB implementation provided in the `Bayes Network Toolbox` by Kevin Murphy. This package can be downloaded at `http://code.google.com/p/bnt/` and provides a large number of useful auxiliary routines.

Section 4.3 Problem 2-4 (a third-order continuous-time system)

For this continuous-time problem since $F(t)$, $G(t) = 0$, and $L(t) = I$ are independent of time i.e. the discrete-time matrix coefficients Φ, Γ, and Λ for the discretized system Equation 139 are given by Equations 23, 31, and 32 or

$$\begin{aligned} \Phi &= \Phi(\Delta t) = e^{F\Delta t} \\ \Gamma &= \Phi(\Delta t)[I_n - \Phi^{-1}(\Delta t)]F^{-1}G = 0 \quad \text{since} \quad G = 0 \\ \Lambda &= \Phi(\Delta t)[I_n - \Phi^{-1}(\Delta t)]F^{-1}L\,, \end{aligned}$$

all of which can be computed. Then the discrete-time model then is

$$x_k = \Phi x_{k-1} + \Lambda w_{k-1}\,.$$

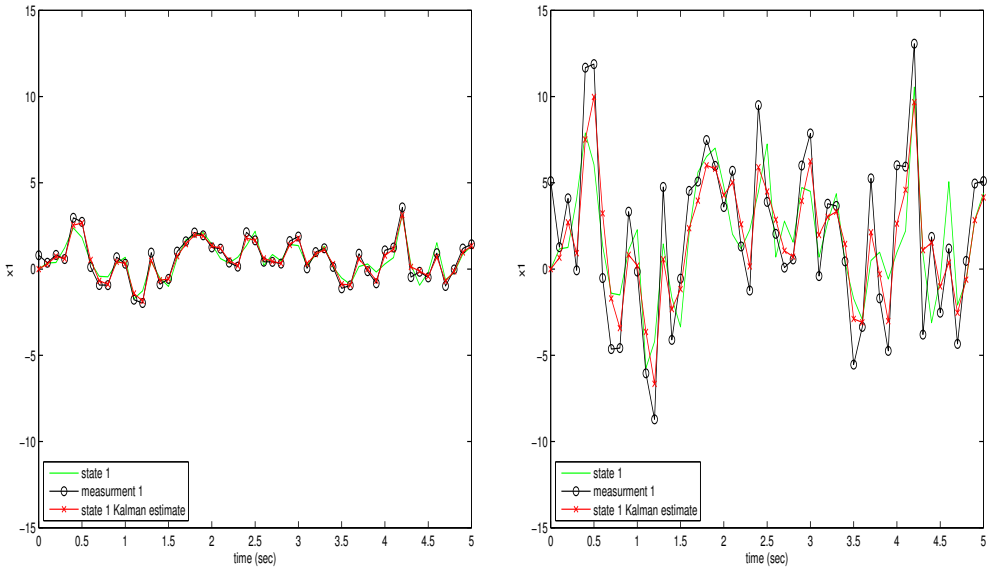

Figure 22: **Left:** The state component x_1 for $t \in [0, 10]$ when $R = 0.25$ and $Q = I$. **Right:** The state component x_2 for $t \in [0, 10]$ when $R = 10.25$ and $Q = 11I$, the addition of 10 to the uncertainties of the previous problem.

We assume that expressions for $E[w_k w_k^T]$ and R given in the text correspond to the *discrete time* noise process. We then formulate the discrete-time Kalman filtering equations and estimate the first state $x_1(t)$ at the discrete times $t_k = 0.1k$. This is done in in the MATLAB function `sect_3_prob_3_N_4.m`. When we run that function we obtain the results given in Figure 22 (left). Then if we increase the the process and measurement noise we get the plot given in Figure 22 (right).

Section 4.3 Problem 5 (the UD decomposition)

The UD decomposition of a matrix P is writing P as $P = UDU^T$ and is equivalent to an eigendecomposition of the matrix P.

Section 4.3 Problem 6 (the matrix condition number)

Assume that $\sigma_{\max}$ is supposed to represent the variable $\lambda_{\max}$ introduced in the text and which represents the maximum eigenvalue of PP^T (the same comment holds for the variable $\sigma_{\min}$). Then the **condition number** of the matrix P $\kappa(P)$ is defined as

$$\kappa(P) \equiv \left(\frac{\lambda_{\max}}{\lambda_{\min}} \right)^{1/2} . \tag{218}$$

This expression can be easily computed with MATLAB.

Section 4.3 Problem 7 (sequential processing to compute the Kalman gain)

I'll assume that for this problem we are asked to estimate $P(+)$ given $P(-)$ using the sequential measurement processing algorithm. We can use this algorithm directly since we have uncorrelated measurements. If we didn't have uncorrelated measurements we would have to find a transformation C, that makes R diagonal i.e. $CRC^T = D$. In the case given, for each row of H we compute a column vector that we store in a matrix K and then using this vector we can compute an *update* to $P(+)$, where $P(+)$ starts initialized as $P(-)$. Each update incorporates another measurement. It should be noted that the final matrix K computed by storing each K_{i_k} column as the algorithm progress is *not* the same matrix as when we compute the Kalman gain "all at once" (see the next problem). We print the final K matrix obtained by saving each column in the manner described below. We implement this procedure in the MATLAB function `sect_3_prob_7.m` and find when finished that

$$K = \begin{bmatrix} 0.9524 & 0.2128 \\ 0.4762 & 0.5532 \\ 0.1429 & 0.2553 \end{bmatrix},$$

and

$$P(+) = \begin{bmatrix} 0.8511 & 0.2128 & 0.0213 \\ 0.2128 & 0.5532 & 0.2553 \\ 0.0213 & 0.2553 & 0.4255 \end{bmatrix}.$$

Section 4.3 Problem 8 (the Joseph form of the a posteriori covariance update)

For this problem we that the book means to use the Joseph form for the covariance update in computing $P(+)$ which is given by Equation 155 with K given by Equation 147. In the MATLAB function `sect_3_prob_8.m` we implement this procedure and find when finished that

$$K = \begin{bmatrix} 0.8511 & 0.2128 \\ 0.2128 & 0.5532 \\ 0.0213 & 0.2553 \end{bmatrix},$$

and

$$P(+) = \begin{bmatrix} 0.8511 & 0.2128 & 0.0213 \\ 0.2128 & 0.5532 & 0.2553 \\ 0.0213 & 0.2553 & 0.4255 \end{bmatrix}.$$

Notice that the expressions for K in this problem and the previous one are *different*, while the expressions for $P(+)$ are the *same* as they should be.

Section 4.4 Problem 1 (correlated disturbance and measurement errors)

As suggested, we consider the discrete-time second order dynamic and measurement equations given by

$$\begin{aligned} x_k &= ax_{k-1} + bx_{k-2} + w_{k-1} \quad \text{for} \quad k \geq 1 \quad \text{and} \quad x_0\,, p_0 \quad \text{given and} \\ z_k &= cx_k + n_k \quad \text{for} \quad k \geq 1\,, \end{aligned}$$

where everything in these expressions is a scalar. We will form a matrix system for these equations by defining the vector state, $\mathbf{x}_k$, as

$$\mathbf{x}_k = \begin{bmatrix} x_k \\ x_{k-1} \end{bmatrix}.$$

Then the vector $\mathbf{x}_k$ state satisfied the dynamic equation

$$\begin{bmatrix} x_k \\ x_{k-1} \end{bmatrix} = \begin{bmatrix} ax_{k-1} + bx_{k-2} + w_{k-1} \\ x_{k-1} \end{bmatrix} = \begin{bmatrix} a & b \\ 1 & 0 \end{bmatrix} \begin{bmatrix} x_{k-1} \\ x_{k-2} \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} w_{k-1}\,.$$

In the discrete-time framework of

$$x_k = \Phi_{k-1}x_{k-1} + \Gamma_{k-1}u_{k-1} + \Lambda_{k-1}w_{k-1}\,,$$

we have matrices for this problem given by $\Phi_{k-1} = \begin{bmatrix} a & b \\ 1 & 0 \end{bmatrix}$, $\Gamma_{k-1} = 0$, $\Lambda_{k-1} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$, and $H_k = \begin{bmatrix} c & 0 \end{bmatrix}$. Let the vector disturbance noise be given by $\mathbf{w}_{k-1} = \begin{bmatrix} w_{k-1} \\ 0 \end{bmatrix}$ so that $Q_{k-1} = \begin{bmatrix} q & 0 \\ 0 & 0 \end{bmatrix}$. Here q is the variance of the scalar process noise w_{k-1}. As suggested for this problem we assume that the disturbance noise and measurement noise are correlated so that

$$E\left\{ \begin{bmatrix} \mathbf{w}_{k-1} \\ \mathbf{n}_k \end{bmatrix} \begin{bmatrix} \mathbf{w}_{k-1}^T & \mathbf{n}_k^T \end{bmatrix} \right\} = \begin{bmatrix} Q_{k-1} & M_k \\ M_k^T & R_k \end{bmatrix},$$

where $Q_{k-1} = \begin{bmatrix} q & 0 \\ 0 & 0 \end{bmatrix}$, $R_k = r$ and

$$M_k = E[\mathbf{w}_{k-1}\mathbf{n}_k^T] = E\left[\begin{bmatrix} w_{k-1} \\ 0 \end{bmatrix} n_k \right] = \begin{bmatrix} E[w_{k-1}n_k] \\ 0 \end{bmatrix} = \begin{bmatrix} m \\ 0 \end{bmatrix},$$

is not the zero vector. The filtering equations for this type of problem are discussed on Page 98 of these notes. We have a state propagation step when there is no control $\Gamma_{k-1} = 0$ given by

$$\hat{\mathbf{x}}_k(-) = \Phi_{k-1}\hat{\mathbf{x}}_{k-1}(+)\,,$$

or in component form this becomes

$$\begin{bmatrix} \hat{x}_{k,1}(-) \\ \hat{x}_{k,2}(-) \end{bmatrix} = \begin{bmatrix} a & b \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \hat{x}_{k-1,1}(+) \\ \hat{x}_{k-1,2}(+) \end{bmatrix},$$

and a state measurement update step given by $\hat{\mathbf{x}}_k(+) = \hat{\mathbf{x}}_k(-) + K_k[z_k - H_k\hat{\mathbf{x}}_k(-)]$ or in component form

$$\begin{bmatrix} \hat{x}_{k,1}(+) \\ \hat{x}_{k,2}(+) \end{bmatrix} = \begin{bmatrix} \hat{x}_{k,1}(-) \\ \hat{x}_{k,2}(-) \end{bmatrix} + K_k[z_k - c\hat{x}_{k,1}(-)] \,.$$

These updates depend on the Kalman gain matrix K_k which is computed from the covariance estimates $P_k(\pm)$. We start with the state propagation covariance matrix

$$\begin{aligned} P_k(-) &= \Phi_{k-1}P_{k-1}(+)\Phi_{k-1}^T + Q_{k-1} \quad \text{for} \quad k \geq 1 \\ &= \begin{bmatrix} a & b \\ 1 & 0 \end{bmatrix} P_{k-1}(+) \begin{bmatrix} a & 1 \\ b & 0 \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} , \end{aligned}$$

with $P_0(+)$ given. The Kalman gain K_k is given by Equation 162 where since $H_k = \begin{bmatrix} c & 0 \end{bmatrix}$ we get

$$\begin{aligned} K_k &= \left(P_k(-) \begin{bmatrix} c \\ 0 \end{bmatrix} + \begin{bmatrix} m \\ 0 \end{bmatrix} \right) \left(\begin{bmatrix} c & 0 \end{bmatrix} P_k(-) \begin{bmatrix} c \\ 0 \end{bmatrix} + \begin{bmatrix} c & 0 \end{bmatrix} \begin{bmatrix} m \\ 0 \end{bmatrix} + \begin{bmatrix} m & 0 \end{bmatrix} \begin{bmatrix} c \\ 0 \end{bmatrix} + r \right)^{-1} \\ &= \left(P_k(-) \begin{bmatrix} c \\ 0 \end{bmatrix} + \begin{bmatrix} m \\ 0 \end{bmatrix} \right) \left(\begin{bmatrix} c & 0 \end{bmatrix} P_k(-) \begin{bmatrix} c \\ 0 \end{bmatrix} + 2cm + r \right)^{-1} . \end{aligned}$$

Note that K_k is of dimension 2×1. Once we have computed K_k the a posteriori measurement covariance matrix $P_k(+)$ is given by Equation 163 which in this case is given by

$$P_k(+) = P_k(-) - K_k \left(\begin{bmatrix} c & 0 \end{bmatrix} P_k(-) - \begin{bmatrix} m & 0 \end{bmatrix} \right) .$$

Section 4.4 Problem 2 (cross-correlated disturbance and measurement noise)

Consider the first-order system

$$\begin{aligned} x_k &= 0.8x_{k-1} + w_{k-1} \quad \text{for} \quad k \geq 1 \quad \text{and} \quad x_0 \,, p_0 \quad \text{given and} \\ z_k &= x_k + n_k \quad \text{for} \quad k \geq 1 \,. \end{aligned}$$

Here x_0 is our initial state estimate and p_0 is its variance. Assume we have cross-correlated disturbance noise and measurement noise such that

$$E \left\{ \begin{bmatrix} w_{k-1} \\ n_k \end{bmatrix} \begin{bmatrix} w_{k-1} & n_k \end{bmatrix} \right\} = \begin{bmatrix} Q & M \\ M & R \end{bmatrix} ,$$

where Q, M, and R are known numerical scalars with $M \neq 0$. Our measurement equation specifies that $H_k = 1$. To simulate this system we must now generate w_{k-1} and n_k at the same time with the given cross-correlation matrix given above for $k = 1, 2, \ldots, 50$. Starting with "initial conditions" on the initial state and it uncertainty of $\hat{x}_0(+) = 0$ and $p_0(+) = 0$ for $k = 1, 2, \ldots 50$ we iterate the Kalman filtering equation for cross-correlated disturbance and measurement noise given by

$$\begin{aligned} \hat{x}_k(-) &= \phi\hat{x}_{k-1}(+) \\ p_k(-) &= \phi^2 p_{k-1}(+) + Q \\ K_k &= (p_k(-) + M)(p_k(-) + 2M + R)^{-1} \\ \hat{x}_k(+) &= \hat{x}_k(-) + K_k(z_k - \hat{x}_k(-)) \\ p_k(+) &= p_k(-) - K_k(p_k(-) + M) \,, \end{aligned}$$

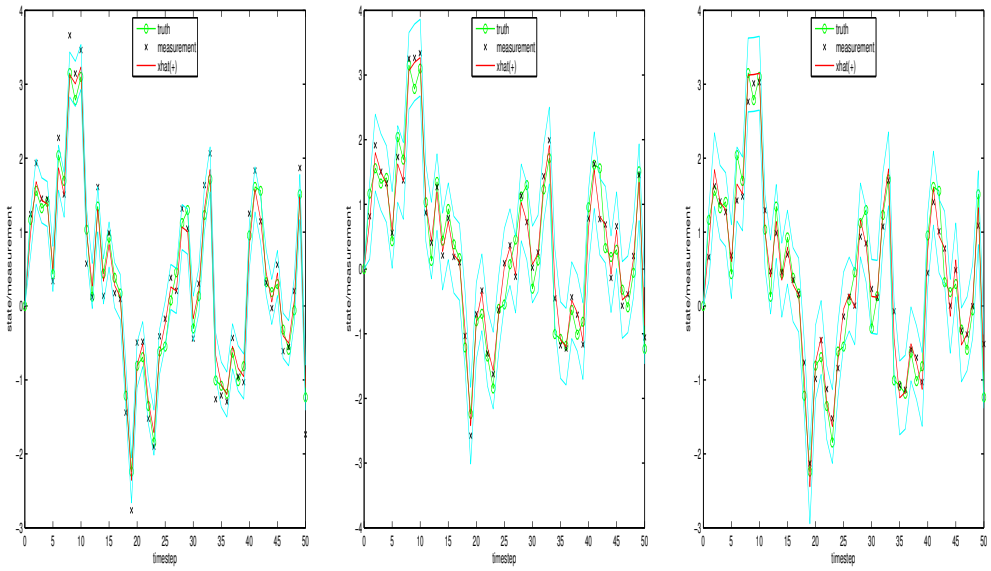

Figure 23: Plots of the state x_k and the a posteriori estimate $\hat{x}_k(+)$ for the cross-correlated process and measurement noise. M measures the cross correlation value/amount. **Left:** $M = 0.25$ **Center:** $M = 0$ **Right:** $M = -0.25$. Notice that the truth (green line) is always between the confidence bands as it should be.

with $\phi = 0.8$, Q, R, and M specified scalars. We implement this procedure in the the MATLAB functions `sect_4_2_gen_xz.m`, `sect_4_2_kfilter_z.m`, and `sect_4_2.m`. When we run these scripts we can change the value of M and observe the affects on the results. We obtain the plots shown in Figure 23. Notice that when implemented correctly, meaning the correlation between measurement and process noise and is taken into account, the numerical value of M does not affect the estimation accuracy.

Section 4.4 Problem 3 (time-correlated measurement noise)

We assume that for this problem we don't have cross-correlated disturbance and measurement noise but instead only have time-correlated measurement noise, given by the dynamics

$$n_k = 0.2n_{k-1} + 0.8\nu_{k-1} = 0.2n_{k-1} + \tilde{\nu}_{k-1}\,, \tag{219}$$

where the scalar $\tilde{\nu}_{k-1}$ has been introduced with statistics

$$\begin{aligned} E[\tilde{\nu}_{k-1}] &= 0 \\ E[\tilde{\nu}^2_{k-1}] &= 0.8^2 E[\nu_{k-1}] = 0.8^2(0.1) = 0.064\,. \end{aligned}$$

Where we have combined the expression $0.8\nu_{k-1}$ into a single noise term (with only one factor) denoted $\tilde{\nu}_{k-1}$ and a different numerical variance value. From this point on we will drop the tilde notation and just assume that our noise term has the statistics given above. Now to fit Equation 219 into the time-correlated measurement noise equation

$$n_k = \Psi_{k-1} n_{k-1} + \nu_{k-1}\,,$$

we take $\Psi_{k-1} = 0.2$ and $Q_{\nu_k} = 0.064$. As in the previous problem $\Phi_{k-1} = 0.8$, $Q = 1$, and $H_k = 1$. To use the optimal filter for this problem we introduced the derived measurement ζ_{k-1}, defined as

$$\zeta_{k-1} = z_k - \Psi_{k-1} z_{k-1} \,,$$

which can be shown equivalent (see Equation 166) to

$$\zeta_{k-1} = D_{k-1} x_{k-1} + n_{D_{k-1}} \,,$$

with D_{k-1} given in this case by

$$D_{k-1} = H_k \Phi_{k-1} - \Psi_{k-1} H_{k-1} = 0.8 - 0.2(1) = 0.6 \,.$$

Then with this derived measurement ζ_{k-1} the process noise w_{k-1} and the derived measurement noise $n_{D_{k-1}}$ are now cross-correlated

$$\begin{aligned} E\left\{ \begin{bmatrix} w_{k-1} \\ n_{D_{k-1}} \end{bmatrix} \begin{bmatrix} w_{k-1} & n_{D_{k-1}} \end{bmatrix} \right\} &= \begin{bmatrix} Q_{k-1} & Q_{k-1} H_k^T \\ H_k^T Q_{k-1} & H_k Q_{k-1} H_k^T + Q_{\nu_{k-1}} \end{bmatrix} \\ &= \begin{bmatrix} 1 & 1 \\ 1 & 1 + 0.064 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & 1.064 \end{bmatrix} . \end{aligned}$$

These components are assigned *new* values of Q_{k-1}, M_{k-1} and R_{k-1} for which we can use the cross-correlated Kalman algorithm to compute our estimates $\hat{x}_k$ and P. This leads to the following algorithm.

Kalman filtering with time-correlated measurement noise

We start our Kalman filtering iterations with $\hat{x}_1(-)$ and $P_1(-)$ *given* and the first two measurements z_1 *and* z_2 measured and are then working on computing the update at x_1. Based on the algorithm presented in the book for $k = 2, 3, \ldots, N$ we need to repeatedly iterate

$$\begin{aligned} \zeta_{k-1} &= z_k - \Psi_{k-1} z_{k-1} \\ D_{k-1} &= H_k \Phi_{k-1} - \Psi_{k-1} H_{k-1} \\ K_{k-1} &= P_{k-1}(-) D_{k-1}^T [D_{k-1} P_{k-1}(-) D_{k-1}^T + R_{k-1}]^{-1} \\ C_{k-1} &= M_{k-1} [D_{k-1} P_{k-1}(-) D_{k-1}^T + R_{k-1}]^{-1} \\ \hat{x}_{k-1}(+) &= \hat{x}_{k-1}(-) + K_{k-1}[\zeta_{k-1} - D_{k-1} \hat{x}_{k-1}(-)] \\ \hat{x}_k(-) &= \Phi_{k-1} \hat{x}_{k-1}(+) + C_{k-1}[\zeta_{k-1} - D_{k-1} \hat{x}_{k-1}(+)] \\ P_{k-1}(+) &= (I - K_{k-1} D_{k-1}) P_{k-1}(-) (I - K_{k-1} D_{k-1})^T + K_{k-1} R_{k-1} K_{k-1}^T \\ P_k(-) &= \Phi_{k-1} P_{k-1}(+) \Phi_{k-1}^T + Q_{k-1} - C_{k-1} M_{k-1}^T - \Phi_{k-1} K_{k-1} M_{k-1} - M_{k-1}^T K_{k-1}^T \Phi_{k-1}^T \,. \end{aligned}$$

The steps above are roughly: given $\hat{x}_k(-)$ and $P_k(-)$ and an infinite sequence of measurements z_k we first compute $\hat{x}_k(+)$ and $P_k(+)$ and then from these compute $\hat{x}_{k+1}(-)$ and $P_{k+1}(-)$. We implement this procedure in the the MATLAB functions `sect_4_3_gen_xz.m`, `sect_4_3_kfilter_z.m`, and `sect_4_3.m`. When we run these scripts we obtain the plot shown in Figure 24.

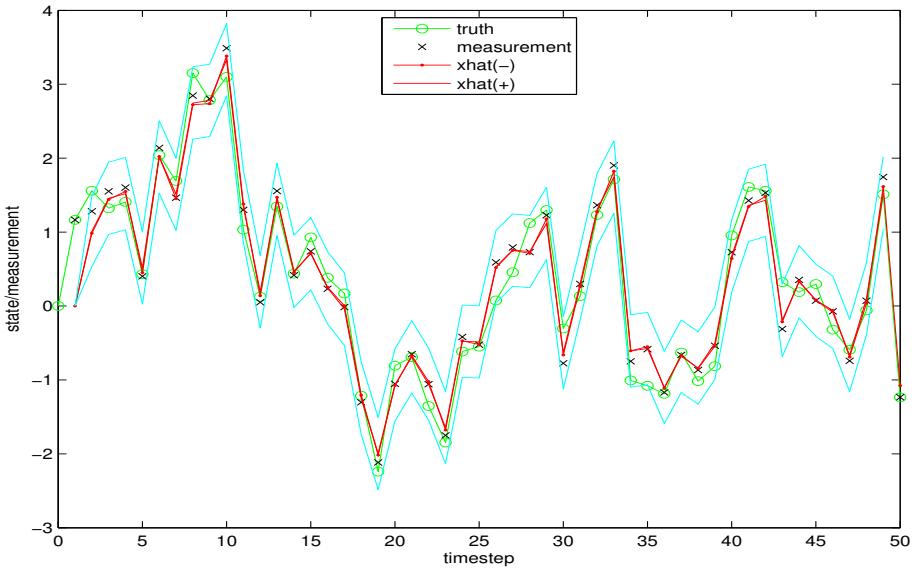

Figure 24: Kalman filtering a scalar problem with correlated measurement noise.

Section 4.4 Problem 4 (more time-correlated measurement errors)

This problem has to deal with time-correlated measurement noise and thus we want to write our system function as

$$\mathbf{x}_k = \Phi_{k-1}\mathbf{x}_{k-1} + \tilde{w}_{k-1}\,,$$

where the process noise $\tilde{w}_{k-1}$ has known second order statistics. From the expressions given we see that $E[\tilde{w}_{k-1}] = 0$ and

$$E[\tilde{w}_{k-1}\tilde{w}_{k-1}^T] = \begin{bmatrix} 0.15 \\ 0.21 \end{bmatrix} (1) \begin{bmatrix} 0.15 & 0.21 \end{bmatrix} = \begin{bmatrix} 0.0225 & 0.0315 \\ 0.0315 & 0.0441 \end{bmatrix}.$$

This last matrix we take to be Q_{k-1} the process noise covariance. We also require our measurement noise dynamics to be given by $n_k = \Psi_{k-1}n_{k-1} + \nu_{k-1}$. From the form of the equations given in the book we have that $\Psi_{k-1} = 0.5I$ and $Q_{\nu_{k-1}} = 0.05I$. We implement this procedure in the the MATLAB functions `sect_4_4_gen_xz.m`, `sect_4_4_kfilter_z.m`, and `sect_4_4.m`. When we run these scripts we obtain the plot shown in Figure 25.

Section 4.4 Problem 5 (time correlated process noise)

For this problem we have time-correlated process noise with independent measurement noise. To filter measurements for this problem we append to the original state vector $\mathbf{x}$ the value

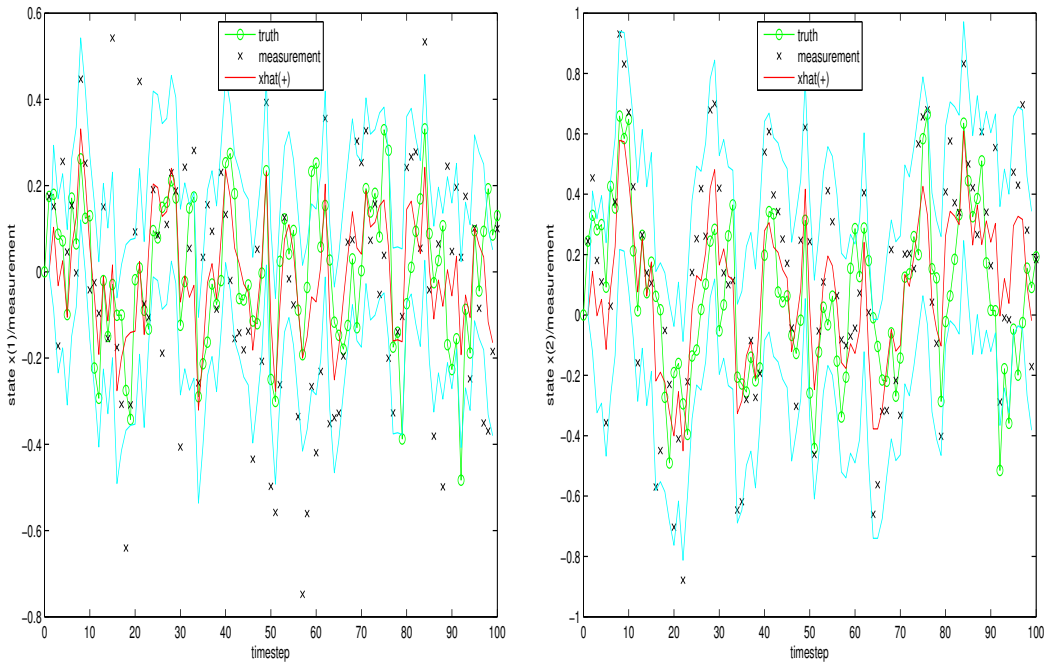

Figure 25: Kalman filtering a 2×2 problem with correlated measurement noise. **Left:** Plots of $x_{k,1}$ the truth in green, $z_{k,1}$ the measurements as a black x and $\hat{x}_{k,1}(+)$ our approximation in red. **Right:** Plots of $x_{k,2}$ the truth in green, $z_{k,2}$ the measurements as a black x and $\hat{x}_{k,2}(+)$ our approximation in red.

of the process noise w_k at time t_k to get the *augmented* state vector $\tilde{\mathbf{x}}_k$ given by

$$\begin{aligned}\tilde{\mathbf{x}}_k &= \begin{bmatrix} x_{k,1} \\ x_{2,k} \\ w_k \end{bmatrix} = \begin{bmatrix} 0.7x_{k-1,1} - 0.15x_{k-1,2} + 0.15w_{k-1} \\ 0.03x_{k-1,1} + 0.79x_{k-1,2} + 0.21w_{k-1} \\ 0.5w_{k-1} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0.5v_{k-1} \end{bmatrix} \\ &= \begin{bmatrix} 0.7 & -0.15 & 0.15 \\ 0.03 & 0.79 & 0.21 \\ 0 & 0 & 0.5 \end{bmatrix} \begin{bmatrix} x_{k-1,1} \\ x_{k-1,2} \\ w_{k-1} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0.5 \end{bmatrix} v_{k-1} \,.\end{aligned}$$

This expression defines new system matrix $\tilde{\Phi}_{k-1}$ and process noise term $\Lambda_{k-1}v_{k-1}$ for the augmented system. Note that the process noise for this augmented system is given by

$$Q_{k-1} = \begin{bmatrix} 0 \\ 0 \\ 0.5 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0.5 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0.25 \end{bmatrix} .$$

In this formulation our problem is now in the normal Kalman filtering framework. Dropping the tilde notation for simplicity, we assume we are *given* x_0 the initial mean of our state and its initial uncertainty in the matrix P_0. We use these to initialize our initial estimates $\hat{x}_0(+)$

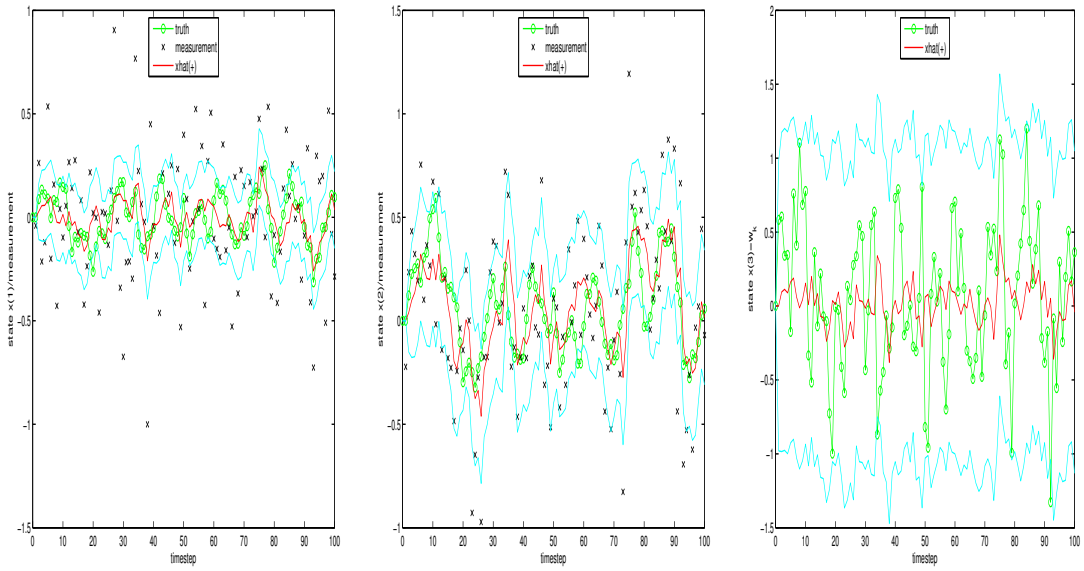

Figure 26: Kalman filtering a 2×2 problem with correlated process noise. This problem can be solved with state augmentation. **Left:** Plots of $x_{k,1}$ the truth in green, $z_{k,1}$ the measurements as a black x and $\hat{x}_{k,1}(+)$ our approximation in red. **Center:** Plots of $x_{k,2}$ the truth in green, $z_{k,2}$ the measurements as a black x and $\hat{x}_{k,2}(+)$ our approximation in red. **Right:** Plots of the true value of $x_{k,3}$ in green (note that this is the same as the value of the process noise w_k) and our estimate of it $\hat{x}_{k,2}(+)$ in red. Notice that since the state x_3 is not observable the error in our estimate of it is significant. The error boundaries (in cyan) emphasis this.

and $P_0(+)$. Then for $k = 1, 2, \cdots$ we iterate

$$\begin{aligned}
\hat{x}_k(-) &= \Phi_{k-1}\hat{x}_{k-1}(+) \\
P_k(-) &= \Phi_{k-1}P_{k-1}(+)\Phi_{k-1}^T + Q_{k-1} \\
K_k &= P_k(-)H_k^T[H_kP_k(-)H_k^T + R_k]^{-1} \\
\hat{x}_k(+) &= \hat{x}_k(-) + K_k[z_k - H_k\hat{x}_k(-)] \\
P_k(+) &= (I_n - K_kH_k)P_k(-)(I_n - K_kH_k)^T + K_kR_kK_k^T \,.
\end{aligned}$$

This is implemented in the MATLAB functions `sect_4_5_gen_xz.m`, `sect_4_5_kfilter_z.m`, and `sect_4_5.m`. When we run these scripts we obtain the plot shown in Figure 26.

Section 4.5 Problem 1 (filtering robot arm measurements)

I'll assume that the dynamics for this problem are similar to the weathervane example (and Problem 3 Section 3.4) and take the form

$$\begin{aligned}
\dot{x}_1 &= x_2 \\
\dot{x}_2 &= -\omega_n^2 x_1 - 2\zeta\omega_n x_2 + w \,,
\end{aligned}$$

Here x_1 is the arm position and x_2 is the arm velocity. The variable ω_n is the systems natural frequency of 10 Hz. The variable ζ is the systems natural damping of 0.3. Fitting this system into the general continuous time framework of

$$\dot{x}(t) = F(t)x(t) + G(t)u(t) + L(t)w(t)\,,$$

We see that $F(t) = \begin{bmatrix} 0 & 1 \\ -\omega_n^2 & -2\zeta\omega_n \end{bmatrix}$, $G(t) = 0$, and $L(t) = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$, and $E[w(t)w(\tau)] = Q_C(t)\delta(t-\tau) = q\delta(t-\tau) = 10\delta(t-\tau)$. The continuous time measurements for this system take the form

$$z(t) = H(t)x(t) + n(t)\,,$$

where $H(t)$ is the time-independent matrix $\begin{bmatrix} 0 & 1 \end{bmatrix}$, $\begin{bmatrix} 1 & 0 \end{bmatrix}$, and $\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$, for Parts a, b, and c respectively. In the first two cases we have $n(t)$ a scalar and measurement noise statistics given by

$$E[n(t)n(\tau)] = R_C(t)\delta(t-\tau) = R\delta(t-\tau) = \delta(t-\tau)\,,$$

since $R = 1$, while in the second case $n(t)$ is a 2×1 vector and we have

$$E[n(t)n(\tau)^T] = R_C(t)\delta(t-\tau) = \begin{bmatrix} r_{11} & 0 \\ 0 & r_{22} \end{bmatrix}\delta(t-\tau) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\delta(t-\tau)\,.$$

With this background for this problem we want to solve Equation 172 with $P(0) = 0$ from the time range $t = 0$ until $t = \Delta t = 0.2$. Once we have $P(t)$ we can use Equation 173 to compute $K_C(t)$. Notice that in all three cases to consider the first three terms on the right-hand-side of Equation 172 are the same. They are given by

$$\begin{aligned} FP + PF^T + LQ_CL^T &= \begin{bmatrix} 0 & 1 \\ -\omega_n^2 & -2\zeta\omega_n \end{bmatrix}\begin{bmatrix} p_{11} & p_{12} \\ p_{12} & p_{22} \end{bmatrix} + \begin{bmatrix} p_{11} & p_{12} \\ p_{12} & p_{22} \end{bmatrix}\begin{bmatrix} 0 & -\omega_n^2 \\ 1 & -2\zeta\omega_n \end{bmatrix} + 10\begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} 2p_{12} & -\omega_n^2 p_{11} - 2\zeta\omega_n p_{12} + p_{22} \\ -\omega_n^2 p_{11} - 2\zeta\omega_n p_{12} + p_{22} & -2\omega_n^2 p_{12} - 4\zeta\omega_n p_{22} + 10 \end{bmatrix}. \end{aligned}$$

The fourth term $-PH^TR_C^{-1}HP$ is different depending on what H is. For each of the above parts a-c above we have that $H^TR_C^{-1}H$ given by

$$\begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}, \quad \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, \quad \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

When we compute $-PH^TR_C^{-1}HP$ from these and get

$$-\begin{bmatrix} p_{12}^2 & p_{12}p_{22} \\ p_{12}p_{22} & p_{22}^2 \end{bmatrix}, \quad -\begin{bmatrix} p_{11}^2 & p_{11}p_{12} \\ p_{11}p_{12} & p_{12}^2 \end{bmatrix}, \quad -\begin{bmatrix} p_{12}^2 + p_{11}^2 & p_{12}p_{22} + p_{11}p_{12} \\ p_{12}p_{22} + p_{11}p_{12} & p_{22}^2 + p_{12}^2 \end{bmatrix}.$$

This problem is further worked in the MATLAB function `sect_5_1.m` where the function `ode45` is used to integrate the above ordinary differential equation for $p_{11}(t)$, $p_{12}(t)$, and $p_{22}(t)$. The `ode45` function needs a function to compute $\dot{x}$ given (t, x). The common part of the ordinary differential equation for $P(t)$ is implemented in the function `sect_5_1_ode_fn.m`. The specific parts are implemented in the three functions:

`sect_5_1_ode_fn_part_a.m`, `sect_5_1_ode_fn_part_b.m`, and `sect_5_1_ode_fn_part_c.m`

which call this common function. When the above script is run it produces the plots shown in Figure 27.

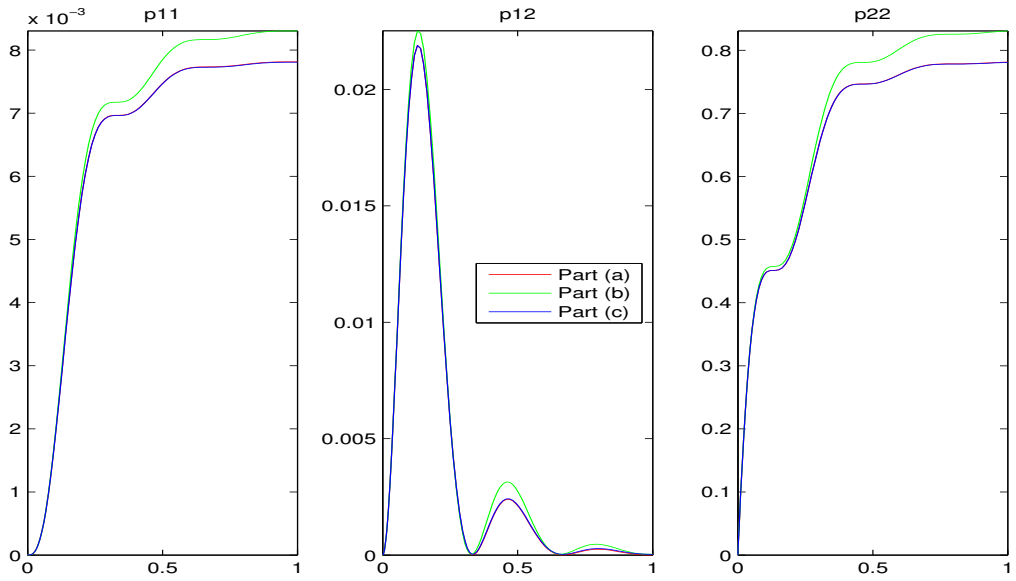

Figure 27: The evolution of elements of the covariance for Problem 1 Section 5. The plots from left-to-right are of $p_{11}(t)$, $p_{12}(t)$, and $p_{22}(t)$ respectively. The elements of the covariance matrix for Part a are plotted in red, for Part b in green and for Part c are in blue.

Section 4.5 Problem 2 (the prediction covariance)

In this case we integrate the equation for $P(t)$ up until the time $t_{\text{calc}} = 0.2$ and then using this value $P(t_{\text{calc}})$ as the initial condition solve Equation 172 but without the $-PH^T R_C^{-1} HP$ term. This means that we integrate the following differential equation

$$\dot{P} = FP + PF^T + LQ_C L^T \,,$$

from t_{calc} until the desired prediction time $t_{\text{est}} = 0.4$. This problem is worked in the MATLAB script `sect_5_2.m` and uses many of the same functions as in the previous problem. When this script is run it produces the plots shown in Figure 28. See the caption for additional details.

Section 4.5 Problem 3 (using a Chandrasekhar-type algorithm)

Since $P(0) = 0$ we can use the special case discussed in the book at the end of the section on the Chandrasekhar-type algorithms where

$$D \equiv FP(0) + P(0)F^T + LQ_C L^T - P(0)H^T R_C^{-1} HP(0) = LQ_C L^T = \begin{bmatrix} 0 & 0 \\ 0 & 10 \end{bmatrix} .$$

Note that D is positive semidefinite with rank $\alpha = 1$, and the number of positive eigenvalues is $\beta = 1$. We have to write the matrix D above as $M_1 M_1^T$ where M_1 is of dimension $n \times \alpha = 2 \times 1$. We can do this if we take $M_1 = \begin{bmatrix} 0 \\ \sqrt{10} \end{bmatrix}$. Then $Y_2(t) \equiv 0$ and the system we

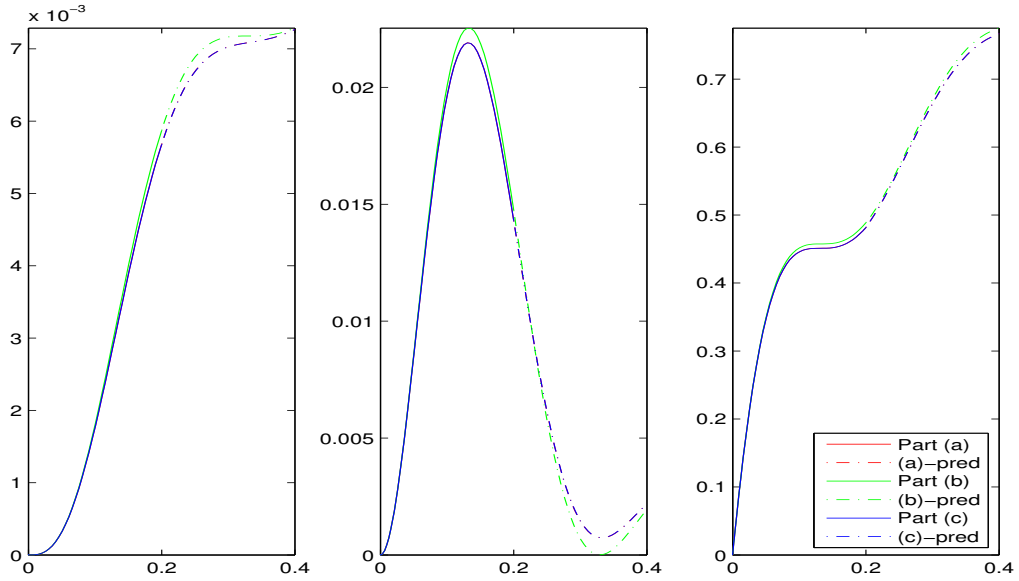

Figure 28: The evolution of elements of the covariance for Problem 2 Section 5. The plots from left-to-right are of $p_{11}(t)$, $p_{12}(t)$, and $p_{22}(t)$ respectively. The elements of the covariance matrix for Part a are plotted in red, for Part b in green and for Part c are in blue. We have measurements up until the time $t_{\text{calc}} = 0.2$ and then have no measurements from this point onwards. The curves for times greater than t_{calc} are drawn as dashed lines. Notice that without measurements our state estimation errors grow as would be expected.

need to solve to compute the continuous Kalman gain $K_C(t)$ is

$$\begin{aligned} \dot{Y}_1(t) &= [F - K_C(t)H]Y_1(t) \quad \text{with} \quad Y_1(0) = M_1 \\ \dot{K}_C(t) &= Y_1(t)Y_1(t)^T H^T R_C^{-1} \quad \text{with} \quad K_C(0) = P(0)H^T R_C^{-1} . \end{aligned}$$

When we write our unknowns as vectors say $Y_1(t) = \begin{bmatrix} y_1(t) \\ y_2(t) \end{bmatrix}$ and $K_C(t) = \begin{bmatrix} k_1(t) \\ k_2(t) \end{bmatrix}$ then the various expressions needed are given by

Part (a): Where $H = \begin{bmatrix} 0 & 1 \end{bmatrix}$ so

$$F - K_C(t)H = \begin{bmatrix} 0 & 1 - k_1(t) \\ -\omega_n^2 & -2\zeta\omega_n - k_2(t) \end{bmatrix} ,$$

and

$$Y_1(t)Y_1(t)^T H^T R_C^{-1} = \begin{bmatrix} y_1(t) \\ y_2(t) \end{bmatrix} \begin{bmatrix} y_1(t) & y_2(t) \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} y_1(t)y_2(t) \\ y_2(t)^2 \end{bmatrix} .$$

Thus the Chandrasekhar-type ordinary differential equation system is given by

$$\begin{aligned} \frac{d}{dt}\begin{bmatrix} y_1(t) \\ y_2(t) \end{bmatrix} &= \begin{bmatrix} (1 - k_1(t))y_2(t) \\ -\omega_n^2 y_1(t) + (-2\zeta\omega_n - k_2(t))y_2(t) \end{bmatrix} \quad \text{with} \quad \begin{bmatrix} y_1(0) \\ y_2(0) \end{bmatrix} = \begin{bmatrix} 0 \\ \sqrt{10} \end{bmatrix} \\ \frac{d}{dt}\begin{bmatrix} k_1(t) \\ k_2(t) \end{bmatrix} &= \begin{bmatrix} y_1(t)y_2(t) \\ y_2(t)^2 \end{bmatrix} \quad \text{with} \quad \begin{bmatrix} k_1(0) \\ k_2(0) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} . \end{aligned}$$

We can of course lump these two systems together and integrate a single system of dimension 4×1. This ordinary differential equation is computed in the MATLAB function `sect_5_3_ode_fn_part_a.m`.

Part (b): Where $H = \begin{bmatrix} 1 & 0 \end{bmatrix}$ so

$$F - K_C(t)H = \begin{bmatrix} -k_1(t) & 1 \\ -\omega_n^2 - k_2(t) & -2\zeta\omega_n \end{bmatrix},$$

and

$$Y_1(t)Y_1(t)^T H^T R_C^{-1} = \begin{bmatrix} y_1(t) \\ y_2(t) \end{bmatrix} \begin{bmatrix} y_1(t) & y_2(t) \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} y_1(t)^2 \\ y_1(t)y_2(t) \end{bmatrix}.$$

Thus the Chandrasekhar-type ordinary differential equation system is given by

$$\begin{aligned} \frac{d}{dt}\begin{bmatrix} y_1(t) \\ y_2(t) \end{bmatrix} &= \begin{bmatrix} -k_1(t)y_1(t) + y_2(t) \\ (-\omega_n^2 - k_2(t))y_1(t) - 2\zeta\omega_n y_2(t) \end{bmatrix} \quad \text{with} \quad \begin{bmatrix} y_1(0) \\ y_2(0) \end{bmatrix} = \begin{bmatrix} 0 \\ \sqrt{10} \end{bmatrix} \\ \frac{d}{dt}\begin{bmatrix} k_1(t) \\ k_2(t) \end{bmatrix} &= \begin{bmatrix} y_1(t)^2 \\ y_1(t)y_2(t) \end{bmatrix} \quad \text{with} \quad \begin{bmatrix} k_1(0) \\ k_2(0) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}. \end{aligned}$$

This ordinary differential equation is computed in the MATLAB function `sect_5_3_ode_fn_part_b.m`.

Part (c): When $H = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ the matrix $K_C(t)$ is now 2×2 and we find

$$F - K_C(t)H = \begin{bmatrix} -k_{11}(t) & 1 - k_{12}(t) \\ -\omega_n^2 - k_{21}(t) & -2\zeta\omega_n - k_{22}(t) \end{bmatrix},$$

and

$$Y_1(t)Y_1(t)^T H^T R_C^{-1} = \begin{bmatrix} y_1(t) \\ y_2(t) \end{bmatrix} \begin{bmatrix} y_1(t) & y_2(t) \end{bmatrix} = \begin{bmatrix} y_1(t)^2 & y_1(t)y_2(t) \\ y_1(t)y_2(t) & y_2(t)^2 \end{bmatrix}.$$

Thus the Chandrasekhar-type ordinary differential equation system is given by

$$\begin{aligned} \frac{d}{dt}\begin{bmatrix} y_1(t) \\ y_2(t) \end{bmatrix} &= \begin{bmatrix} -k_{11}(t)y_1(t) + (1 - k_{12}(t))y_2(t) \\ (-\omega_n^2 - k_{21}(t))y_1(t) + (-2\zeta\omega_n - k_{22}(t))y_2(t) \end{bmatrix} \\ \frac{d}{dt}\begin{bmatrix} k_{11}(t) & k_{12}(t) \\ k_{21}(t) & k_{22}(t) \end{bmatrix} &= \begin{bmatrix} y_1(t)^2 & y_1(t)y_2(t) \\ y_1(t)y_2(t) & y_2(t)^2 \end{bmatrix}, \end{aligned}$$

with initial conditions given by

$$\begin{bmatrix} y_1(0) \\ y_2(0) \end{bmatrix} = \begin{bmatrix} 0 \\ \sqrt{10} \end{bmatrix} \quad \text{with} \quad \begin{bmatrix} k_1(0) \\ k_2(0) \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}.$$

This ordinary differential equation is computed in the MATLAB function `sect_5_3_ode_fn_part_c.m`.

When the MATLAB script `sect_5_3.m` is run we numerically integrate each of the above systems and then compare the corresponding results with $K_C(t)$ found from Problem 1 in this section. In each case the between $K_C(t)$ computed in two different ways is quite good.

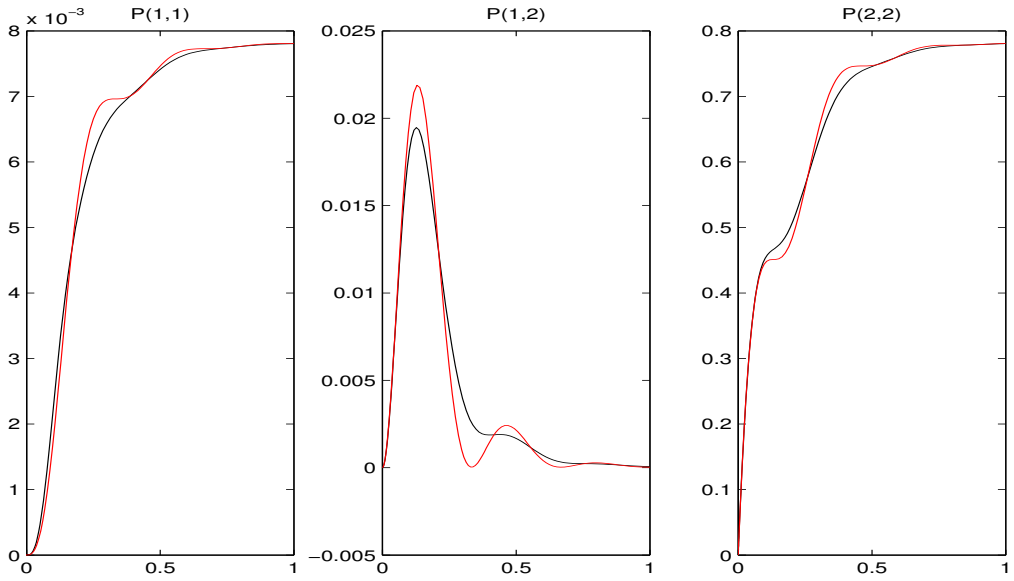

Figure 29: The evolution of elements of the covariance for Problem 4 Section 5. The plots from left-to-right are of $p_{11}(t)$, $p_{12}(t)$, and $p_{22}(t)$ respectively. The black curves are computed using the square root algorithm, while the red curves are computed using the Kalman-Bucy filter.

Section 4.5 Problem 4 (using a square-root algorithm)

In this formulation we must solve the ordinary differential equation

$$\dot{S}(t) = S(t)M_{\rm UT}(t) \quad \text{with} \quad S(0)S(0)^T = P(0)\,,$$

and S is upper triangular $S(t) = \begin{bmatrix} s_{11}(t) & s_{12}(t) \\ 0 & s_{22}(t) \end{bmatrix}$ and $M_{\rm UT}(t)$ given by the upper triangular part of the right-hand-side of Equation 197. In this expression when only H changes three terms are the same

$$S^{-1}FS + S^TF^TS^{-T} + S^{-1}LQ_CL^TS^{-T}\,,$$

and only one term is different $-S^TH^TR_C^{-1}HS$. Recall that the matrix product $LQ_CL^T = \begin{bmatrix} 0 & 0 \\ 0 & 10 \end{bmatrix}$. Then with the upper triangular form of S given above, in the Mathematica file `evaluate_MUT.nb` we compute the three common terms above needed to compute $M_{\rm UT}(t)$, $S(t)M_{\rm UT}(t)$ for the three common terms and then $-SS^TH^TR_C^{-1}HS$. The three common terms are coded in the MATLAB function `sect_5_4_ode_fn.m`, while each of the the specific matrix forms for H in the parts above are coded in function with names that denote the part (a,b, or c) in which they are derived. For example `sect_5_4_ode_fn_part_a.m`. The driver script for this problem is `sect_4_5.m` and when that is run we produce plots of the elements of the covariance matrix $p_{11}(t)$, $p_{12}(t)$ and $p_{22}(t)$. An example plot generated for Part (c) is shown in Figure 29. The black curves are computed using the square root algorithm, while the red curves are computed using the Kalman-Bucy filter. While the curves don't agree exactly this difference may be due to the fact that in the square root formulation we are expecting to start with a positive definite matrix $P(0)$ to decompose as $S(0)S(0)^T$. Since our

$P(0)$ in this problem is 0 and is there for not positive definite we might expect the numerical approximation to be suboptimal.

Section 4.5 Problem 5 (estimation of drug content in the blood)

The dynamic system and measurement equations for this problem are given by

$$\begin{aligned} \dot{x}_1 &= -k_1 x_1 + u \\ \dot{x}_2 &= k_1 x_1 - k_2 x_2 \\ z &= x_2 + n \,. \end{aligned}$$

Where the constants $k_1 > 0$ and $k_2 > 0$. Here x_1 is the mass of drug in the gastrointestinal track and must be less than or equal to another constant $x_{1,\max}$, x_2 is the mass of the drug in the blood stream, while u is our control. We desire to measure x_2. For this problem we see that this

$$F = \begin{bmatrix} -k_1 & 0 \\ k_1 & -k_2 \end{bmatrix}, \quad G = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad L = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \quad \text{and} \quad H = \begin{bmatrix} 0 & 1 \end{bmatrix}.$$

with $R_C(t) = r$ a scalar constant. In this formulation of the problem there is no process noise. We will modify this system to add scalar noise terms w_i to each equation. Then in that case $L = I$ and $Q = \text{Diag}\begin{bmatrix} q_1 & q_2 \end{bmatrix}$. Then to use the Kalman-Bucy filter we need to first solve for $P(t)$ the following ordinary differential equation

$$\dot{P} = FP + PF^T + Q - P \begin{bmatrix} 0 & 0 \\ 0 & r^{-1} \end{bmatrix} P \,,$$

with a given initial condition on $P(t)$, say $P(0) = 0$ if the initial state is known with certainty. Once we have solved this system for $P(t)$ the continuous Kalman gain $K_C(t)$ is given by $K_C(t) = P(t)H^T R_C^{-1} = P(t) \begin{bmatrix} 0 \\ r^{-1} \end{bmatrix}$.

Section 4.6 Problem 1 (oven temperature)

We are told to consider the dynamic system

$$\dot{x}(t) = -k_1 x(t) - k_2 x(t)^4 + k_3 u + k_4 \,,$$

with measurements given by

$$z(t) = x(t) + n(t) \,.$$

Now since the process noise k_4 has a *nonzero* mean (it has the mean value of 1) we will remove that mean value from k_4 and write $k_4 = 1 + \tilde{k}_4$, where $\tilde{k}_4$ has the same variance as k_4 but with a mean of 0.0. When we do this our nonlinear dynamical system becomes

$$\dot{x}(t) = -k_1 x(t) - k_2 x(t)^4 + k_3 u + 1 + \tilde{k}_4 \,.$$

from which we see that our nonlinear system $\dot{x} = f(\cdot)$ has $f = -k_1 x(t) - k_2 x(t)^4 + k_3 u + 1$, and the new "control" term is $k_3 u + 1$.

Part (a): The book has discussed how to convert a *linear* continuous time system into a discrete system. Since the given system is nonlinear the techniques presented in the book do not seem to be directly applicable for this problem. Thus the way I choose to simulate this system is using a method similar to the *hybrid* extended Kalman filter discussed in the book. We first break up the interval of simulation into segments where the measurements occur $t_0 < t_1 < \cdots < t_{k-1} < t_k < \cdots < t_N = t_{\text{final}}$. Assume we know the state at the beginning of this interval $x(t_{k-1})$. Then over an internal interval say $t_{k-1} < t < t_k$ we integrate the nonlinear equation $\dot{x}(t) = f[x, u, w, p, t]$ (with zero noise) over that time

$$x(t_k) = x(t_{k-1}) + \int_{t_{k-1}}^{t_k} f[x(\tau), u(\tau), 0, p(\tau), \tau] d\tau \,,$$

using a general purpose integration routine. At the time t_k we then will apply a "jump" in the state x due to the process noise which was neglected in the above integration. This process noise will be a random variable drawn from a Gaussian with zero mean and a variance given by what we would compute for the *linearized* problem which in this case is $\dot{x}(t) = -k_1 x(t) + \tilde{k}_4$. Note we drop the control terms $k_3 u + 1$ here. Based on the results derived on Page 93 for the same linear system we have that the covariance of the discrete-time process noise for this system should be given by

$$q = \frac{q_C}{2(-k_1)}(e^{2(-k_1)\Delta t} - 1) = \frac{q_C}{2k_1}(1 - e^{-2k_1 \Delta t}) \,.$$

Where $\Delta t = t_k - t_{k-1}$. Then with the process noise added to the state we then generate a measurement using $z(t_k) = x(t_k) + n$ where n is a random draw from a Gaussian distribution with mean 0 and variance R.

The above procedure is implemented in the MATLAB code `sect_6_1_gen_xz.m`.

Warning: As this system is in fact nonlinear, the above approach may not in fact be the correct way to simulate from this system. If anyone sees anything wrong with this approach and knows of a better method please contact me.

Part (b): For this part we drop the $k_2 x^4$ term from the dynamic equation which gives

$$\begin{aligned} \dot{x}(t) &= -k_1 x(t) + k_3 u + 1 + \tilde{k}_4 \\ z(t) &= x(t) + n(t) \,. \end{aligned}$$

We could solve this in the continuous Kalman-Bucy framework if desired. To that end note that the state equation has $F = -k_1$, $Gu = k_3 u$, $Lw = k_4$. To compute the filter gain $K_C(t)$ using the Kalman-Bucy filter we must first solve

$$\dot{P} = FP + PF^T + LQ_C L^T - PH^T R_C^{-1} HP \,,$$

which in this case becomes

$$\dot{p} = -2p + 1 - p^2 \quad \text{with} \quad p(0) = 0 \,.$$

Note that everything is a scalar. Once we have $p(t)$ then $K_C(t) = P(t)H^T R_C^{-1} = p(t)$ in this case.

Since we will be comparing these linearized results with the results when we use the extended Kalman filter we will instead discretize the above system and the perform Kalman filtering on that system. For the time independent continuous system like we have above

$$\dot{x}(t) = Fx(t) + Gu(t) + Lw(t)\,,$$

the discrete-time formulation of the mean propagation looks like

$$x_k = \Phi x_{k-1} + \Gamma(k_3 u + 1) \quad \text{for} \quad k \geq 0\,,$$

with

$$\begin{aligned}\Phi(\Delta t) &= e^{F\Delta t} = e^{-k_1 \Delta t} \\ \Gamma &= \Phi(\Delta t)[I_n - \Phi^{-1}(\Delta t)]F^{-1}G = \frac{1}{k_1}(1 - e^{-k_1\Delta t})\,.\end{aligned}$$

The uncertainty added at each time step is given by the variance of the discrete-time process noise which is q as calculated above. See the notes around Page 10. With this formulation and since $H_k = 1$ and the state dimension is 1 the discrete-time Kalman equations are

$$\begin{aligned}\hat{x}_k(-) &= \Phi\hat{x}_{k-1}(+) + \Gamma(k_3 u + 1) \\ P_k(-) &= \Phi P_{k-1}(+)\Phi^T + q \\ K_k &= P_k(-)(P_k(-) + R)^{-1} \\ \hat{x}_k(+) &= \hat{x}_k(-) + K_k[z_k - \hat{x}_k(-)] \\ P_k(+) &= (1 - K_k)P_k(-)\,.\end{aligned}$$

This is implemented in the MATLAB code `sect_6_1_linear_kf.m`.

Part (c): To use the hybrid Extended Kalman-Bucy Filter for each interval $t_{k-1} < t < t_k$ we let the estimates of the mean state and its uncertainty at the start of the interval t_{k-1} be given by $\hat{x}[t_{k-1}(+)]$ and $P[t_{k-1}(+)]$ we then integrate the coupled nonlinear system

$$\begin{aligned}\dot{x}(t) &= -k_1 x(t) - k_2 x^4(t) + k_3 u + 1 \\ \dot{P}(t) &= F(t)P(t) + P(t)F(t)^T + q_C \\ &= 2(-k_1 - 4k_2 x(t)^3)P(t) + q_C\,,\end{aligned}$$

to the time $t = t_k$ starting with the initial conditions on x and P of $\hat{x}[t_{k-1}(+)]$ and $P[t_{k-1}(+)]$. This gives the new state estimate and uncertainty $\hat{x}[t_k(-)]$ and $P[t_k(-)]$. Once we have these two values we compute the filter gain (since $H \equiv 1$)

$$K(t_k) = P[t_k(-)](P[t_k(-)] + R)^{-1}\,,$$

and then the state and covariance measurement update according to

$$\hat{x}[t_k(+)] = \hat{x}[t_k(-)] + K(t_k)\{z_k - \hat{x}[t_k(-)]\} \tag{220}$$

$$P[t_k(+)] = (1 - K(t_k))P[t_k(-)]\,. \tag{221}$$

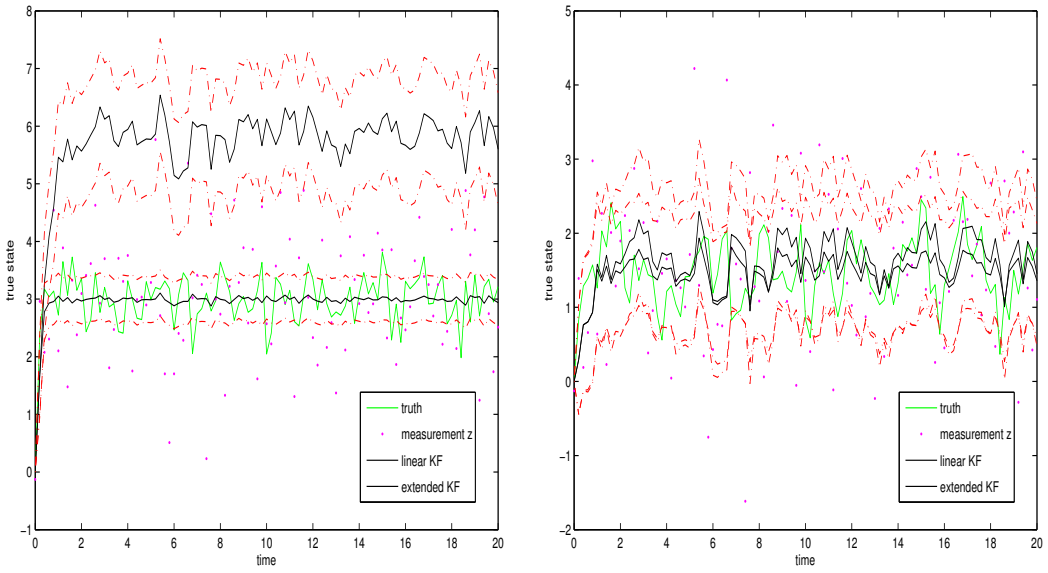

Figure 30: Plots of the true state x_k (in green) and two Kalman filtered estimates of its value $\hat{x}_k$ (in black). The first Kalman filter estimate corresponds to dropping the x^4 term (denoted as "linear") and the second corresponds to filtering with the extended Kalman filter (denoted as "extended"). In addition, around each state estimate we plot its 95% confidence interval (in red). Measurements are shown in cyan (purple). **Left:** When the forcing term is "large" i.e. $k_3 u = 10$. **Right:** When the forcing term is "large" i.e. $k_3 u = 1$. See the text for more details.

Then this process repeats over the next time interval $t_k < t < t_{k+1}$. This is implemented in the MATLAB code `sect_6_1_extended_kf.m`.

The above routines are driven using the MATLAB script `sect_6_1.m`. We first performed the numerical computations as suggested with $k_3 u = 10$. The resulting plot is given in Figure 30 (left). We next performed the numerical computations but $k_3 u = 1$. The resulting plot is given in Figure 30 (right). Notice that for both values of $k_3 u$ extended Kalman filter does a good job at representing the true state. When the value of $k_3 u$ is small the limiting value of x is not that large and thus the linear Kalman filter does a good job filtering. When the value of $k_3 u$ is large (say 10) the linear Kalman filter does *not* do a good job representing the true state. The nonlinearity in this case appears to the user as a bias to the true state. This shows what type of trouble that filtering with the wrong differential equation for the physical process can appear. This is often called the "model mismatch" problem.

As an example of how the term $k_2 x^4$ can affect the final solution x if we take $\dot{x} = 0$ to consider the steady-state solution (denoted x^*) we get

$$0 = -k_1 x^* - k_2 x^{*4} + k_3 u + 1 \,.$$

For the case where we don't include x^{*4} and the two values of $k_3 u$ of 10 and 1 we find

$$x^* = \frac{k_3 u + 1}{k_1} = 11 \,, \text{ and } \quad 2 \,.$$

While when we do include x^{*4} using a root finding algorithm we find

$$x^* = 2.991492\,,\ \text{and}\quad 1.497336\,.$$

This simple calculation shows how different the results can be when we drop the x^4 term or not. These computations are done in the `R` code `sect_6_1_roots.R`.

Section 4.6 Problem 2 (a nonlinear measurement mapping)

For the given system given to match the continuous time dynamic process model

$$\dot{x}(t) = F(t)x(t) + G(t)u(t) + L(t)w(t)\,,$$

we have a state $\mathbf{x}$ given by $\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}$ and system matrices

$$F(t) = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -1/\tau & 0 \\ 0 & 0 & 0 & -1/\tau \end{bmatrix}, \quad G(t)u(t) = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1/\tau & 0 \\ 0 & 1/\tau \end{bmatrix} \begin{bmatrix} u_3 \\ u_4 \end{bmatrix}, \quad L(t)w(t) = \begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ w_4 \end{bmatrix},$$

with $\tau = 30$ seconds. In the book the vector $L(t)w(t)$ seems to have *two* components equal to w_4. I'm going to assume this is a typo. Coupled with this linear system we have the *nonlinear* measurement mapping

$$z(t) = h[x(t), t] + n(t)\,,$$

where the functional form for $h[x(t), t]$ is given in the book. To implement the extended Kalman filter for this problem we will need to linearize this measurement equation. To do this we will need to evaluate $\frac{\partial h}{\partial x}$. To this end recall that

$$\frac{d}{dx}\sin^{-1}(x) = \frac{1}{\sqrt{1-x^2}}\,.$$

If we define the variable $\Theta = \Theta(x_3, x_4)$ as

$$\Theta(x_3, x_4) = \frac{x_4}{(x_3^2 + x_4^2)^{1/2}}\,,$$

then we have that

$$\begin{aligned} \frac{\partial \Theta}{\partial x_3} &= -\frac{x_3 x_4}{(x_3^2 + x_4^2)^{3/2}} \\ \frac{\partial \Theta}{\partial x_4} &= \frac{1}{(x_3^2 + x_4^2)^{1/2}} - \frac{x_4^2}{(x_3^2 + x_4^2)^{3/2}}\,, \end{aligned}$$

both of which are functions of the current state $\mathbf{x}$. For the extended Kalman filter we will need to compute the linearization of the measurement equation or

$$
\begin{aligned}
H_k &= \left.\frac{\partial h[x,t]}{\partial x(t)}\right|_{x=\hat{x}_k(-),t=t_k(-)} \\
&= \left.\begin{bmatrix} 2(x_1-r_{n_1}) & 2(x_2-r_{e_1}) & 0 & 0 \\ 2(x_1-r_{n_2}) & 2(x_2-r_{e_2}) & 0 & 0 \\ 0 & 0 & \frac{x_3}{(x_3^2+x_4^2)^{1/2}} & \frac{x_4}{(x_3^2+x_4^2)^{1/2}} \\ 0 & 0 & \frac{1}{\sqrt{1-\Theta^2}}\frac{\partial \Theta}{\partial x_3} & \frac{1}{\sqrt{1-\Theta^2}}\frac{\partial \Theta}{\partial x_4} \end{bmatrix}\right|_{x=\hat{x}_k(-),t=t_k(-)},
\end{aligned}
$$

where the reference points are given by $(r_{n_1}, r_{e_1}) = (0,0)$ and $(r_{n_2}, r_{e_2}) = (10^5, 0)$.

Part (a): A time-independent continuous system (like we have here)

$$\dot{x}(t) = Fx(t) + Gu(t) + w(t)\,,$$

transforms into the discrete-time formulation

$$x_k = \Phi x_{k-1} + \Gamma u_{k-1} + w_{k-1}\,, \tag{222}$$

with

$$
\begin{aligned}
\Phi(\Delta t) &= e^{F\Delta t} \\
\Gamma(\Delta t) &= \Phi(\Delta t)[I_n - \Phi^{-1}(\Delta t)]F^{-1}G\,,
\end{aligned}
$$

or since in this case F is not invertible one must use the expression that uses the Taylor series approximation to $\Gamma(\Delta t)$ given in Equation 35. We assume that the discrete-time noise w_k has $E[w_{k-1}w_{k-1}^T] = Q$, with Q given in the problem. We are assuming we are given Q in the *discrete-time* formulation. With this background starting at an initial state $\hat{x}_0(+)$ and uncertainty $P_0(+)$ the extended Kalman filtering equations are

$$
\begin{aligned}
\hat{x}_k(-) &= \Phi\hat{x}_{k-1}(+) + \Gamma u_{k-1} \\
P_k(-) &= \Phi P_{k-1}(+)\Phi^T + Q \\
H_k &= \left.\frac{\partial}{\partial x(t)}\right|_{x=\hat{x}_k(-),t=t_k(-)} = \text{using the above expression} \\
K_k &= P_k(-)H_k^T\{H_kP_k(-)H_k^T + R\}^{-1} \\
\hat{x}_k(+) &= \hat{x}_k(-) + K_k\{z(t_k) - h[\hat{x}_k(-), t_k]\} \\
P_k(+) &= [I_n - K_kH_k]P_k(-)\,.
\end{aligned}
$$

Part (b-d): For this part of the problem, we compute a truth trajectory using Equation 222 and iterate the extended Kalman filtering equations above. This procedure requires several codes all of which are called from the main MATLAB script `sect_6_2.m`. When this main driver script is run it produces the plots for each of the states given in Figure 31.

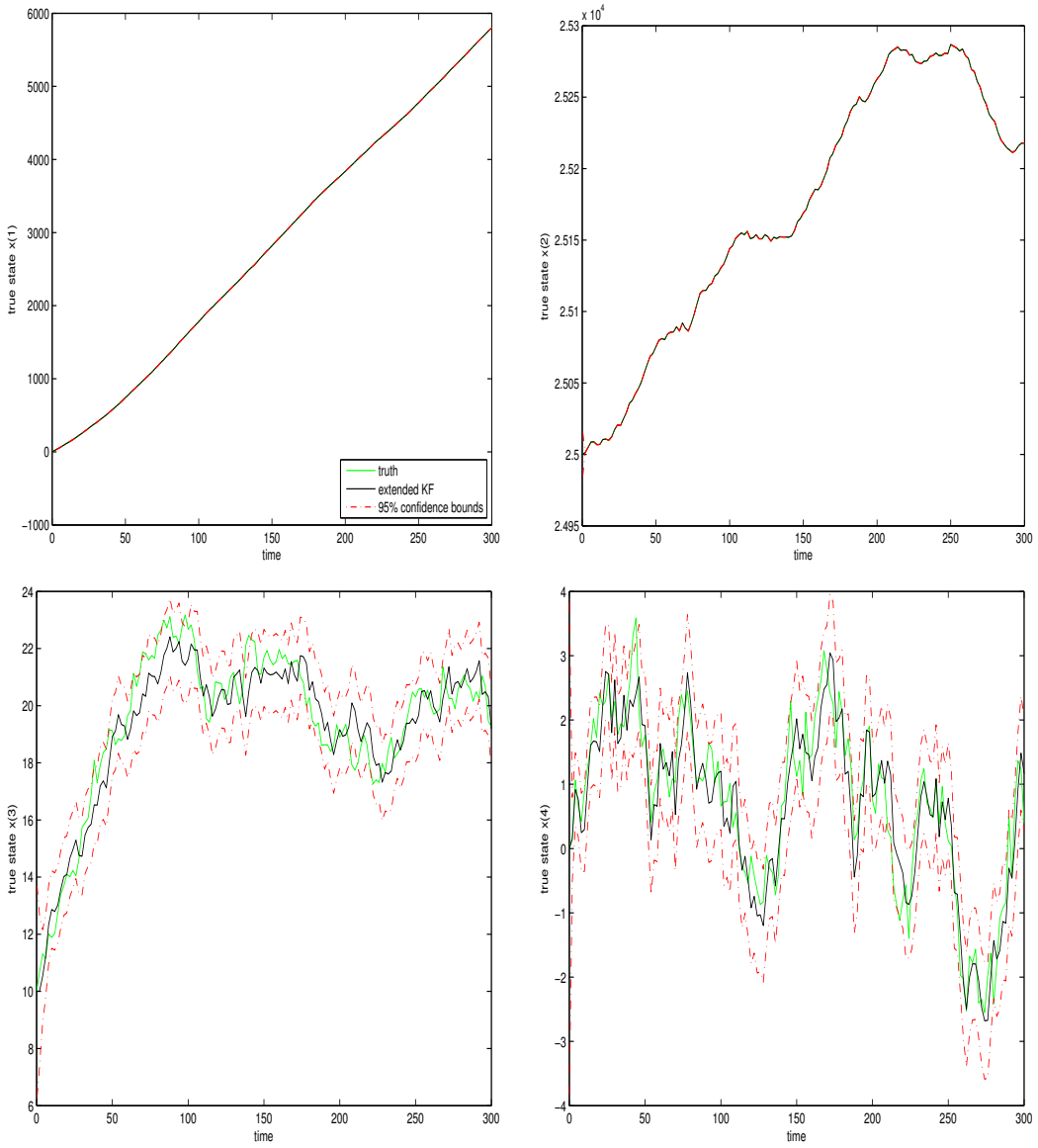

Figure 31: Plots of the true state x_k for $k = 1, 2, 3, 4$ (in green) and the extended Kalman filtered estimate $\hat{x}_k$ (in black). In addition, around each state estimate we plot its 95% confidence interval (in red). Since the measurements are nonlinear mappings of the state they are not shown. **Top Left:** The state x_1. **Top Right:** The state x_2. **Bottom Left:** The state x_3. **Bottom Right:** The state x_4. See the text for more details. Notice that the estimation is so good with the states $x_1(t)$ and $x_2(t)$ that the estimates and the truth coincide.

Section 4.6 Problem 3 (another ships initial condition)

For this problem we used the same MATLAB script `sect_6_2.m` but with modified initial conditions and a forcing of $u_3 = 0$ and $u_4 = 20$ rather than $u_3 = 20$ and $u_4 = 0$. When we run with these initial conditions we get the plots given in Figure 32.

Section 4.6 Problem 4 (increasing the sizes of Q and R)

For this problem we used the same MATLAB script `sect_6_2.m` but with modified versions of Q and R. When we increase each term of Q by 10. When we do that, we expect our system state to possibly possess more movement in each time step since the random vector drawn now comes from a distribution with a larger variance and can therefore itself be of a larger magnitude. As the original Q matrix was

$$Q = \text{diag}(4, 4, 0.25, 0.25)\,,$$

we expect that this increase to affect the states x_3 and x_4 most since 10 is a larger percentage of 0.25. Plotting $x_3(t)$ in this case in Figure 33 (left) demonstrates this.

When we increase each term of Q by 10 we get the result for $x_3(t)$ shown in Figure 33 (right). In this case the change in the truth or state estimate is hard to observe.

Section 4.6 Problem 5 (a describing function)

For the nonlinear function $f(x)$ defined by

$$f(x) = \begin{cases} -D & x < -a \\ 0 & -a \le x \le a \\ D & x > a \end{cases}\,,$$

we want to compute the describing function. Note the book considers the case where $D = 1$. Note that $E[f(x)] = 0$, thus $a_0 = 0$, $E[\tilde{x}] = \sigma^2$, and so

$$a_1 = \frac{1}{\sigma^2} E[f(x)\tilde{x}] = \frac{1}{\sigma^2} E[f(x)x]\,,$$

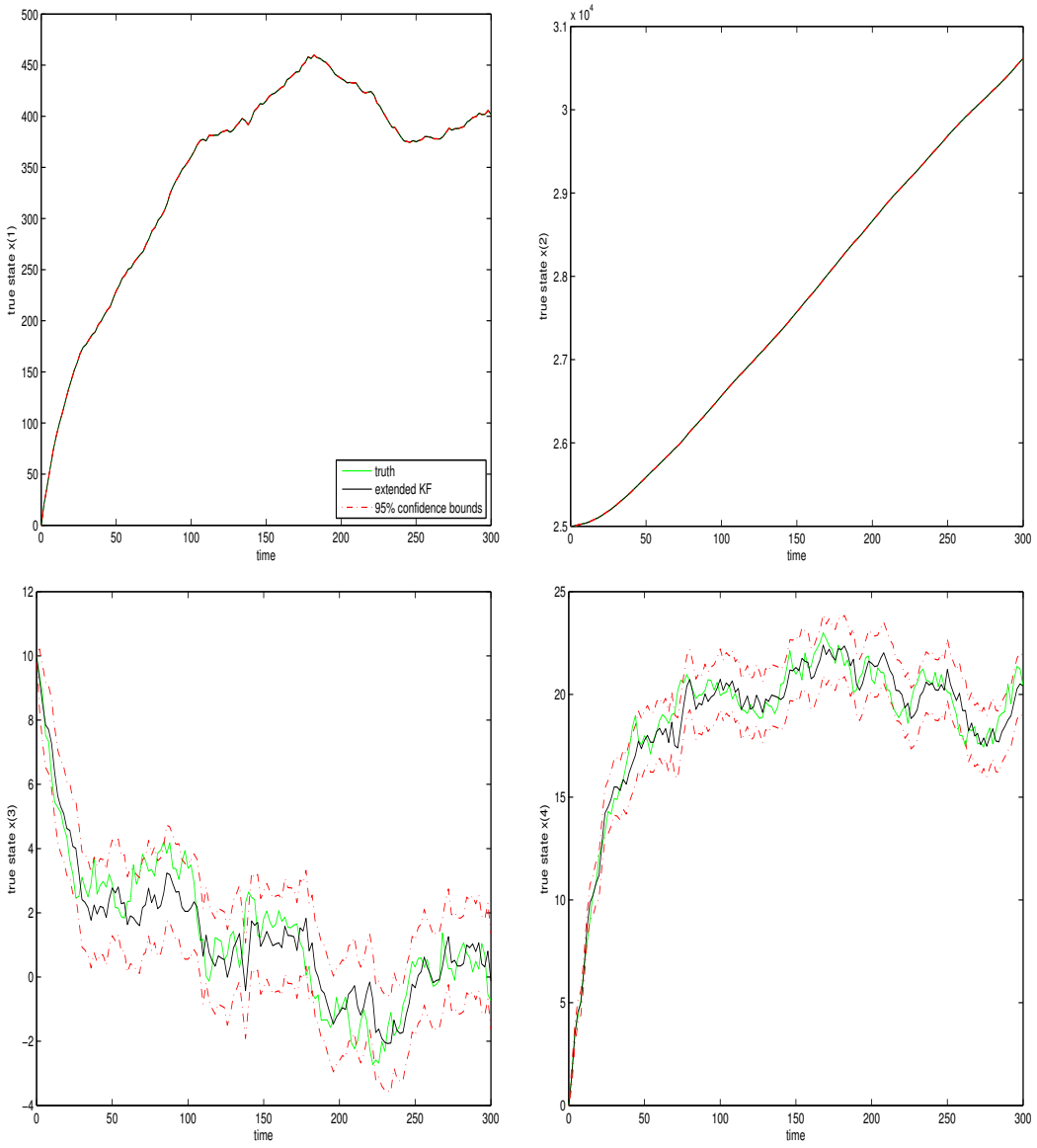

Figure 32: The same four states as in Problem 2 and given in Figure 31 but with a different initial condition and forcing. Note that the qualitative behavior of x_1 and x_2 and x_3 and x_4 seems to have switched between these two runs.

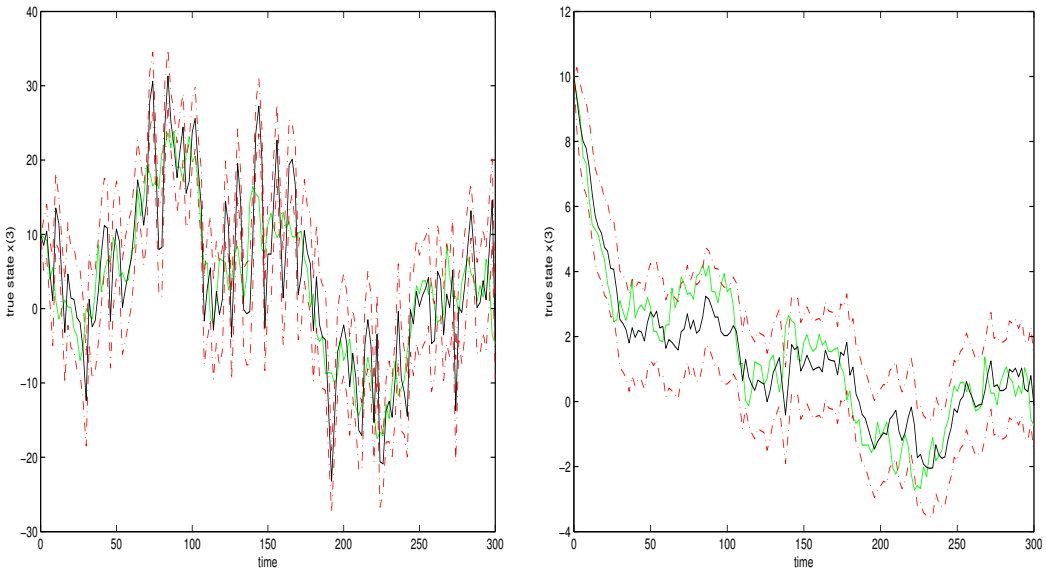

Figure 33: The state $x_3(t)$ as in Problem 3 and given in Figure 32 but with modified Q (left) and R (right) matrices. We see that modifying Q gives a more volatile time series while there seems to be less of an effect on the state and its estimate when we modify R.

when we assume that $E[x]$ is zero. For a zero mean Gaussian density recall that $p(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}}$ and we compute

$$\begin{aligned} a_1 &= \frac{1}{\sigma^2}\int x f(x) p(x) dx \\ &= -\frac{D}{\sigma^2}\int_{-\infty}^{-a} x \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}} dx + \frac{D}{\sigma^2}\int_a^{\infty} x \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}} dx \\ &= -\frac{D}{\sigma^2}\frac{1}{\sqrt{2\pi}\sigma}(-\sigma^2)\, e^{-\frac{x^2}{2\sigma^2}}\Big|_{-\infty}^{-a} + \frac{D}{\sigma^2}\frac{1}{\sqrt{2\pi}\sigma}(-\sigma^2)\, e^{-\frac{x^2}{2\sigma^2}}\Big|_{a}^{\infty} \\ &= \frac{D}{\sigma\sqrt{2\pi}}(e^{-\frac{a^2}{2\sigma^2}} - 0) - \frac{D}{\sigma\sqrt{2\pi}}(0 - e^{-\frac{a^2}{2\sigma^2}}) = \sqrt{\frac{2}{\pi}}\frac{D}{\sigma} e^{-\frac{a^2}{2\sigma^2}} . \end{aligned}$$

Section 4.7 Problem 1 (the parameter-adaptive filter)

Part (a-b): See the MATLAB script `sect_7_prob_1.m` where we work this problem. When we run that script we find that the Kalman estimate *with* the error in the $(2,1)$ element of Φ seems to be a very good approximation to the true state.

Part (c): For this part of the problem we augment the original state $\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$ with the value of the $(2,1)$ element of Φ. If we denote this element as a, then our augmented dynamical

system is thus given by

$$\left[\begin{array}{c} x_1 \\ x_2 \\ a \end{array}\right]\Bigg|_{k+1} = \left[\begin{array}{ccc} 0.9 & 1 & 0 \\ a & 0.8 & 0 \\ 0 & 0 & 1 \end{array}\right] \left[\begin{array}{c} x_1 \\ x_2 \\ a \end{array}\right]\Bigg|_{k} + \left[\begin{array}{cc} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{array}\right] \left[\begin{array}{c} w_1 \\ w_2 \end{array}\right].$$

These dynamic equations are coupled with a measurement equation given by

$$z_k = \left[\begin{array}{ccc} 1 & 0 & 0 \\ 0 & 1 & 0 \end{array}\right] \left[\begin{array}{c} x_1 \\ x_2 \\ a \end{array}\right]\Bigg|_{k} + n_k \,.$$

Our system above is nonlinear as it has an ax_1 term. We must use the extended Kalman filter to perform inference on this system. The state propagation is done using the above nonlinear mapping of the augmented state. The covariance propagation step needs the linearized coefficient matrix F_A. For the above system we find that

$$F_A = \left[\begin{array}{ccc} 0.9 & 1 & 0 \\ a & 0.8 & x_1 \\ 0 & 0 & 1 \end{array}\right].$$

Using these we can formulate the extended Kalman filter for this problem. This is implemented in the the MATLAB script `sect_7_prob_1.m`, and the associated subroutines. When we run that script we get the plots given in Figure 34.

Section 4.7 Problem 2 (tests for whiteness)

This problem is implemented in the `R` function `sect_7_prob_2.R`. When that script is run it produces a plot like that seen in Figure 35.

Section 4.7 Problem 3 (a simple example of noise adaptive filtering)

Warning: These results do not look like what I would expect. I'm not sure that what I have done is wrong, I have tried techniques like this in other context and have not had much luck getting them to work. I'm not sure where the discrepancy lies between theory and practice. If anyone sees anything wrong with what I have done please contact me.

The true system has $w_k \sim N(0, 10)$ and $n_k \sim N(0, 1)$ and we will be changing the values of R and Q used in filtering to experiment with noise adaptive filtering. We first discuss how to obtain a better approximation of R the noise covariance matrix. At each step in the filtering we compute the **measurement residual** r_k given by

$$r_k = z_k - H\hat{x}_k(-) \,. \tag{223}$$

Then if our filtering is optimal $E[\hat{x}_k n_k^T] = E[x_k n_k^T] = 0$ and we have

$$E[r_k r_k^T] = H P_k(-) H^T + R \,. \tag{224}$$

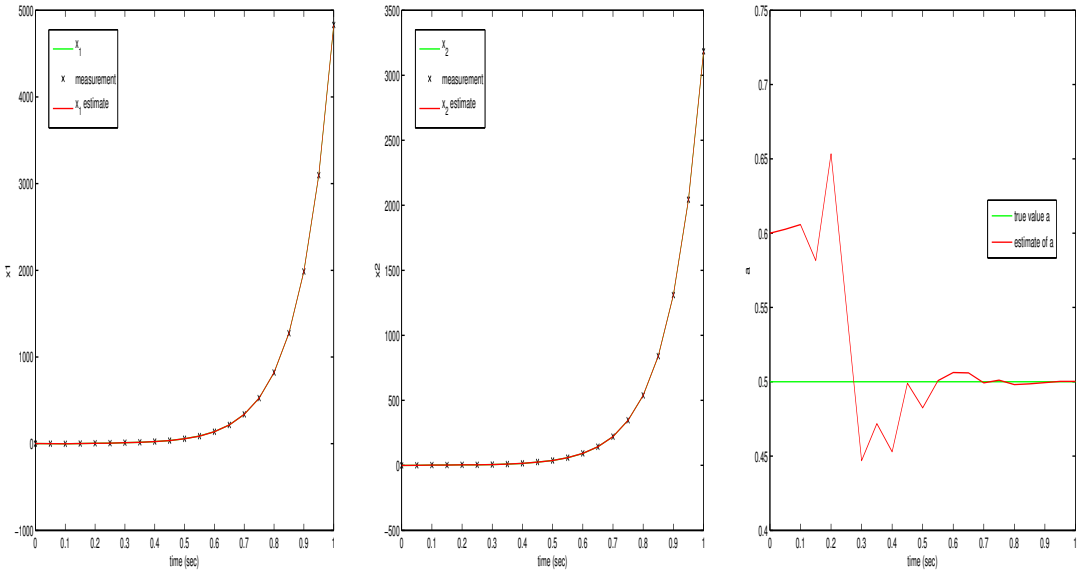

Figure 34: Estimation of a parameter using extended Kalman filtering and state augmentation. **Left:** Plots of $x_{k,1}$ the truth in green, $z_{k,1}$ the measurements as a black x and $\hat{x}_{k,1}(+)$ our approximation in red. **Center:** Plots of $x_{k,2}$ the truth in green, $z_{k,2}$ the measurements as a black x and $\hat{x}_{k,2}(+)$ our approximation in red. **Right:** Plots of the true value of the $(2,1)$ denoted as a in green and our estimate over time of it $\hat{x}_{k,3}(+)$ in red. Notice that our estimate of the $(2,1)$ component of Φ starts at the value of 0.6 and converges to 0.5. See the text for additional details.

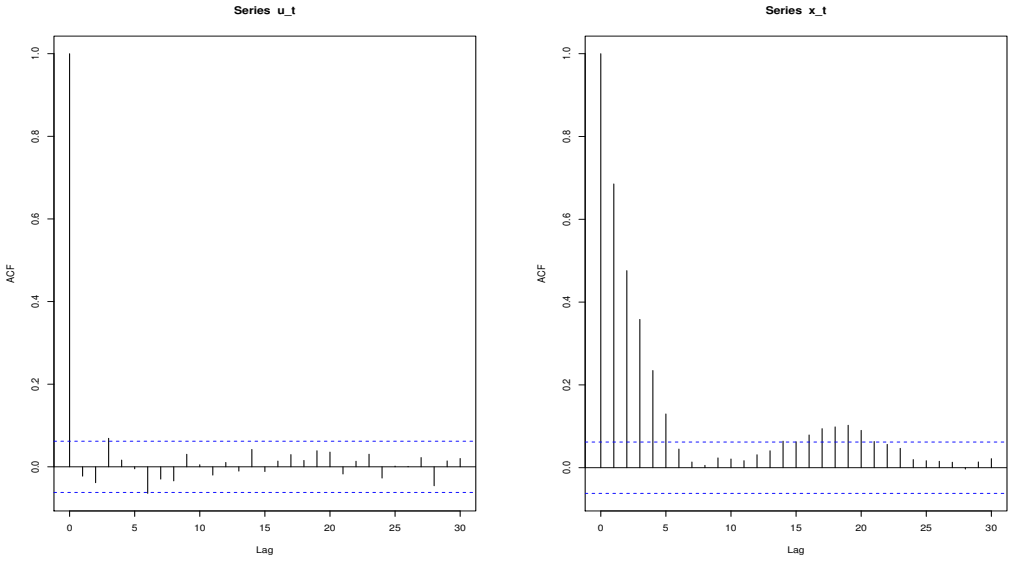

Figure 35: **Left:** Plots of the autocorrelation function for u_k (Gaussian white noise). Note that the autocovariance function at all lags are insignificant. This sequence is most certainly white. **Right:** Plots of the autocorrelation function for the sequence $z_k = 0.7z_{k-1} + u_k$ a first order AR model. Note that the autocovariance function has an exponential type decay starting at the lag of $k = 0$. This is a characteristics of AR time series. This sequence is most not certainly white.

We assume that we have processed N measurements and can compute the sample mean of r_k in the normal way

$$\bar{r} = \frac{1}{N}\sum_{k=1}^{N} r_k \,. \tag{225}$$

Under optimal filtering r_k is zero mean but if we filter with incorrect estimates of measurement and noise covariance Q and R to compute an estimate of $E[r_k r_k^T]$ we must subtract the sample mean. This using Equation 224 to get an approximation of R (denoted $\hat{R}$) as

$$\frac{1}{N-1}\sum_{k=1}^{N}(r_k - \bar{r})(r_k - \bar{r})^T = \frac{1}{N}\sum_{k=1}^{N} HP_k(-)H^T + \hat{R} \,.$$

Our estimate of R depends on N the number of processed measurements. Solving for $\hat{R}$ we can write it under one summation as

$$\hat{R} = \frac{1}{N-1}\left\{\sum_{k=1}^{N}(r_k - \bar{r})(r_k - \bar{r})^T - \frac{N-1}{N}HP_k(-)H^T\right\} \,, \tag{226}$$

or as two summations as

$$\hat{R} = \frac{1}{N-1}\sum_{k=1}^{N}(r_k - \bar{r})(r_k - \bar{r})^T - \frac{1}{N}\sum_{k=1}^{N} HP_k(-)H^T \,. \tag{227}$$

If our filter has been operating for a long time and has reached steady-state $P_k(-)$ has become constant denoted by $P_{SS}(-)$ and the above estimate of R becomes

$$\hat{R} = \hat{S} - HP_{\text{SS}}(-)H^T \,. \tag{228}$$

We note that in the above we are neglecting state-noise correlation i.e. we assume that $E[\hat{x}_k n_k^T] = 0$. Since when we have *incorrect* values for Q or R this will not necessarily be true. In that case its not clear how good of an approximation the above will be. One approach would be to use the initial value of R for a while and once non-optimal steady-state filtering has been achieved update it using the approximation $\hat{R}$ given by Equation 228. This could be performed several times as needed.

We now discuss how to obtain a better approximation of the process noise covariance matrix Q. Unlike the measurement residual $r_k = z_k - \hat{x}_k(-)$, where we have observed or computed every variable in that expression, in the **process residual** q_k given by

$$q_k = x_{k+1} - \Phi\hat{x}_k(+) \,, \tag{229}$$

we don't have access to the hidden variable x_{k+1}. In process noise covariance filtering we have to approximate it with $\hat{x}_{k+1}(+)$ thus we will take for q_k the approximation

$$q_k \approx \hat{x}_{k+1}(+) - \Phi\hat{x}_k(+) \,.$$

The theoretical forcing residual q_k can be shown to equal

$$q_k = \Phi[x_k - \hat{x}_k(+)] + w_k \,,$$

which under optimal steady-state filtering again requires that

$$Q_{S_k} \equiv E[(q_k - \bar{q})(q_k - \bar{q})^T] = \Phi P_k(+)\Phi^T + Q\,, \tag{230}$$

Solving for Q in the above expression and denoting this as $\hat{Q}$ and using a sample based approach to estimate $E[(q_k - \bar{q})(q_k - \bar{q})^T]$ we have

$$\hat{Q} = \frac{1}{N-1}\sum_{k=1}^{N}(q_k - \bar{q})(q_k - \bar{q})^T - \frac{1}{N}\sum_{k=1}^{N}\Phi P_k(+)\Phi^T\,. \tag{231}$$

In steady-state this estimate becomes

$$\hat{Q} = Q_S - \Phi P_{SS}(+)\Phi^T\,. \tag{232}$$

With this background we are ready to implement this problem. This problem is worked in the MATLAB codes `sect_7_prob_3.m` and `sect_7_prob_3_noise_adaptive_filtering.m`. When these scripts are run they generate the plots shown in Figure 36. Notice that both $\bar{r}$ and $\bar{q}$ approach 0 as we take more measurements for all filtering both with and without the correct model parameters. Note that the estimate of R when filtering with the correct model $\hat{R}$ eventually limits to the input measurement noise covariance $R = 1$ which shows consistency of this estimation technique. The estimate of Q when filtering with the correct model does not limit to the correct value of 10 but seems to limit to something less. *Neither* of the estimates of R or Q when filtering with the incorrect model gives estimates of Q or R that are very close to the values used to generate the data $Q = 10$ and $R = 1$. In fact for the incorrect model $(Q, R) = (20, 0.5)$ the estimate of R is negative.

Again I'm not sure where the difficulty with this technique lies. If anyone has any insight into this please let me know.

Section 4.7 Problem 4 (a simple example of multiple-model estimation)

This problem is worked in the MATLAB code `sect_7_prob_4.m`. When that code is run it generates the plots shown in Figure 37.

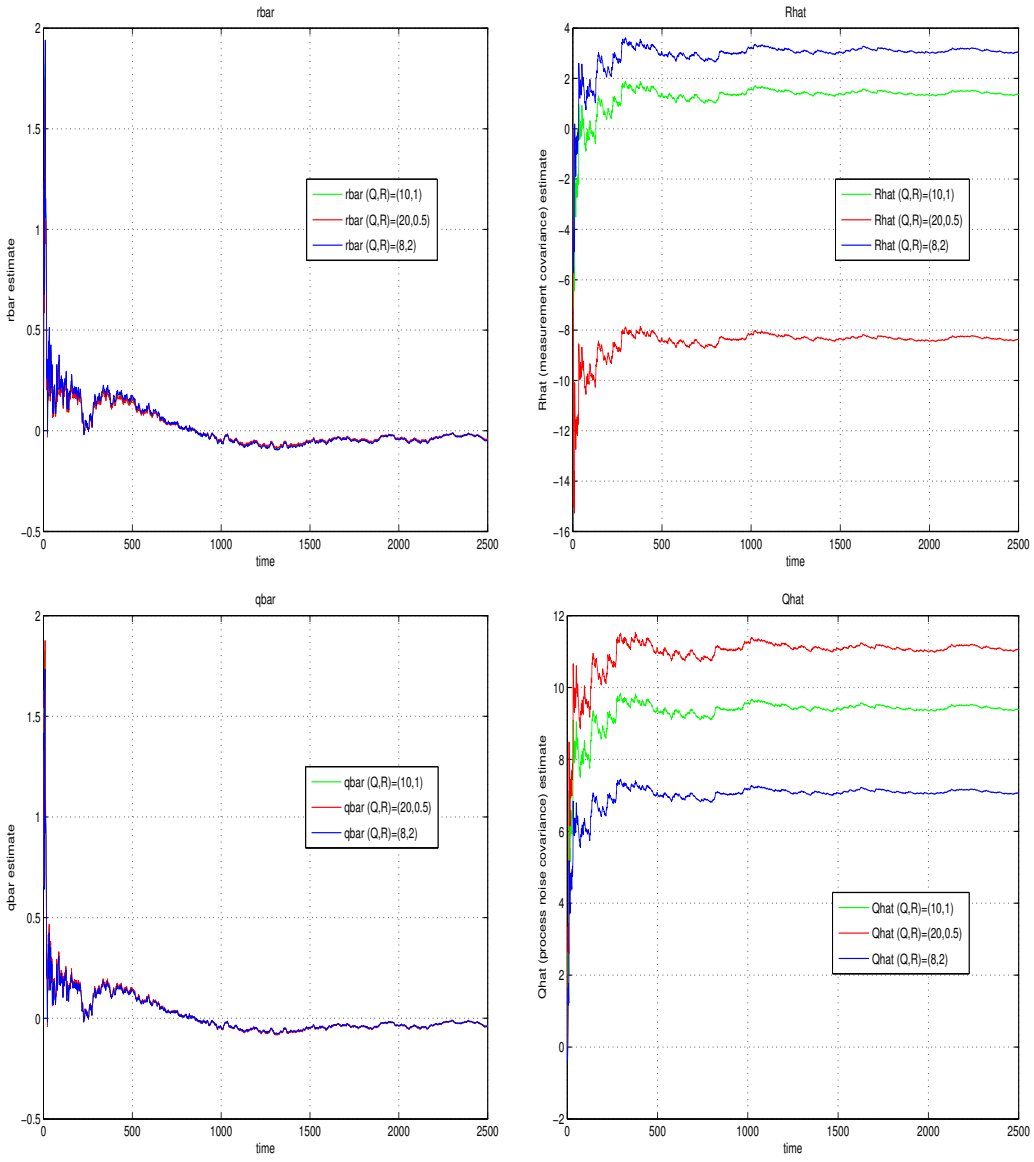

Figure 36: **Top Left:** Plots of $\bar{r}(N)$ the mean measurement residual as a function of time N. **Top Right:** Plot of $\hat{R}(N)$ the estimate of the measurement noise covariance as a function of time N. **Bottom Left:** Plots of $\bar{q}(N)$ the mean process noise residual as a function of time N. **Bottom Right:** Plot of $\hat{Q}(N)$ the estimate of the process noise covariance as a function of time N.

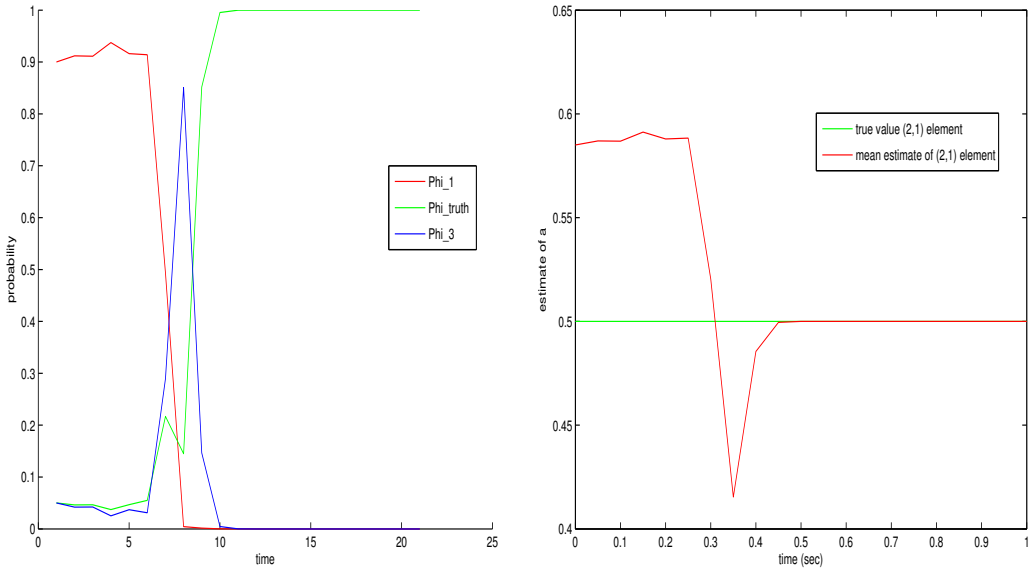

Figure 37: **Left:** Plots of the posteriori probability of various models. Each model is a different value for the $(2,1)$ component of Φ. The correct model wins as time goes on. **Right:** Plots of the probability blended estimate of the $(2,1)$ element of Φ. Eventually the element goes to 0.5 the correct value.

References

[1] A. Gelb. *Applied optimal estimation*. MIT Press, 1974.