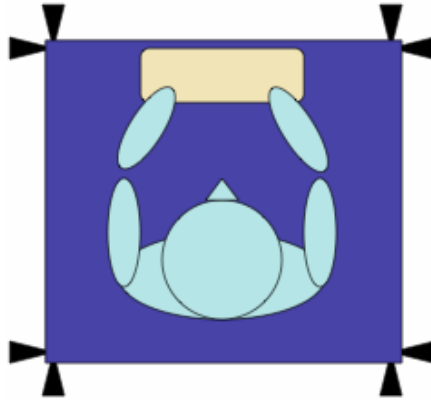


## SBE 2006 Dynamics Homework Problem

### *Preamble*

Imagine you are a test director at Johnson Space Center (JSC). You are interested in testing a training version of SAFER (Simplified Aid For EVA Rescue) to be used on the Precision Air Bearing Floor (PABF).

The training version of the SAFER is designed to work only in one plane (i.e., the plane of motion parallel to the plane of the floor). The SAFER can be controlled in two ways: translation or rotation, depending on which jets are active at any time. Balanced jets in one direction will result in a force applied in the opposite direction. Counteracting jets on either side of the SAFER will result in a torque about the center of mass of the SAFER. The SAFER uses “bang-off-bang” control, meaning each jet is either fully on or fully off. A diagram of the training SAFER can be found below.



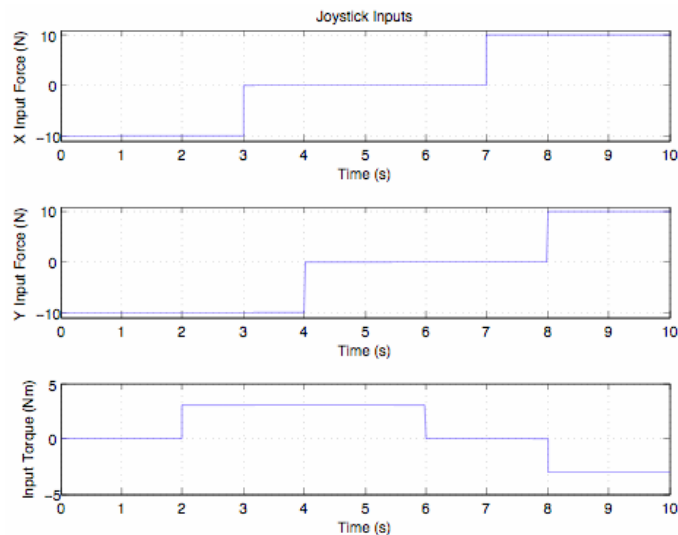
In order to evaluate the performance of this SAFER model, you want to track the motion of a subject wearing the SAFER training unit. To collect data for your test, you have setup a small, motion tracking system. The tracking system uses magnetic markers on the SAFER and provides noisy measurements of the planar X and Y positions as well as a rotation angle,  $\theta$ , of the SAFER.

This problem explores the dynamics of this situation, simulates it and leads you through the development of a simple Kalman filter for tracking the position, velocity, angle and angular rate of a subject using this training version of the SAFER.

## Problems

1. Attach a local reference frame that rotates and translates with the subject on the SAFER (call this the “body frame”). Expressing the forces and torques in the body frame, derive the translational and rotational dynamics that describe the motion **in the inertial reference frame**. (You can assume that all forces and torques are applied through the center of mass of the subject / SAFER system). Justify any assumptions you make regarding what effects to model and which ones you are leaving out.
2. Using Simulink, develop a computational model that simulates the motion of the subject on the training SAFER unit.

A Simulink model has been started for you called `dynamics_prob2.mdl`. This model requires a small Matlab script to be run to initialize some variables called `dynamics_prob2_script.m`. You will notice 3 “From Workspace” blocks in `dynamics_prob2.mdl`. These are the control inputs you should assume. The figure below illustrates the forces that these blocks generate for you (forces are either  $\pm 10$  or 0 Newtons and the torque is either  $\pm 3$  or 0 Newton meters):



The `dynamics_prob2.mdl` model has been set to have a fixed time step of 0.01 seconds and the simulation will run from 0 to 10 seconds (these are set in the “Configuration Parameters” dialog under the “Simulation” menu in Simulink). For your dynamics, assume a mass of the subject / SAFER system to be 125 kg and assume the inertia about the center of mass to be  $10 \text{ kg m}^2$ . Plot the x-y position and the angular displacement of the SAFER that results from the above control inputs. You can assume that the SAFER starts from rest at position (0,0) meters with an initial displacement angle of 0 radians.

3. Copy the dynamics you created in part 2 into `dynamics_prob345.mdl`. Again, hook up the control inputs in the same manner as part 2. In `dynamics_prob345.mdl`, you will notice several random noise blocks. Add the disturbance accelerations into your dynamics in the *inertial frame*. The noise to the x and y accelerations generated by the disturbance accelerations blocks is zero mean, with a variance of  $0.01 \text{ m}^2$ . The noise to the angular acceleration is zero mean, with a variance of  $0.001 \text{ radians}^2$ . These parameters are defined in the `dynamics_prob345_script.m` Matlab script. This script must be run before simulating your Simulink block diagram. This noise represents the “process noise” and will be used later in the Kalman filter you write.
4. Using the random noise input blocks again, create simulated noisy measurements of the x and y position as well as the angular displacement by adding noise to the position outputs of your dynamics. You can use the measurement random noise blocks provided to you in `dynamics_prob345.mdl` (from part 3). These blocks assume that the noise on the x and y position measurements is zero mean, with a variance of  $0.005 \text{ m}^2$ . The noise on the angular displacement measurement is also zero mean, with a variance of  $0.001 \text{ radians}^2$ .
5. Using the “To Workspace” blocks, output the true position and velocity as well as your noisy measurements to the workspace. These blocks have been provided to you in `dynamics_prob345.mdl`. Simply connect your true position and velocity signals and the noisy measurements from your block diagram to these blocks and the variables “” and “” will appear in the Matlab workspace after you run the simulation.

*The next part of this problem set can be done either in Simulink or using regular Matlab scripts and functions. Sometimes, it can be tricky to get Simulink to process events in exactly the order you require. Running a Kalman filter requires that you compute specific quantities at very specific times. As such, Matlab scripts / functions are recommended.*

6. Write a Kalman filter that estimates the position, velocity (both x and y), angle and angular rate of the subject on the SAFER training unit. Use the noisy x position, y position and angular position measurements you created in part 4 as measurements.

To start your simulation, add a random initial estimation error by taking the truth and adding random noise to it (using the `randn` Matlab function). The noise should be zero mean with the following variances:

x-y initial position error variance:  $0.025 \text{ m}^2$   
 x-y initial velocity error variance:  $1 \text{ (m/s)}^2$   
 initial angle error variance:  $0.01 \text{ radians}^2$   
 initial angular rate error variance:  $0.02 \text{ (radians/second)}^2$

Use these initial error variances to form your initial covariance matrix (by placing them on the matrix diagonal).

You can use the script entitled `dynamics_prob6_script.m` to get you started. This will setup all of the variables needed to run the simulation and add noise to your initial estimate (as per the parameters defined above). This script also automatically simulates the model created in parts 2 - 5, assuming your model is called `my_solution.mdl` (you can change this to be whatever you want).

In writing the Kalman filter, you will need to discretize your continuous dynamics (developed in part 1). You can use the function:

```
[Phi, Bk, Qk] = get_discrete_dyn(A, Q, B, time_step)
```

provided for you. The above function accepts the continuous state-space dynamics matrix ( $A$ ), the continuous process noise matrix ( $Q$ ), the continuous state-space input matrix ( $B$ ) and the time step, and outputs the discretized state transition matrix ( $\Phi$ ), the discretized input matrix ( $B_k$ ) and the discretized process noise matrix ( $Q_k$ ). In other words, it takes the system:

$$\dot{X}(t) = A \cdot X(t) + B \cdot u(t)$$

and converts it into:

$$X_{k+1} = \Phi \cdot X_k + B_k \cdot u_k$$

In addition, it provides the required  $Q_k$  matrix for the time update of the Kalman filter. Note that the continuous  $Q$  matrix is computed for you in `dynamics_prob6_script.m`.

7. Plot the estimation error resulting from your Kalman filter with the covariance bounds on either side. The covariance bounds are found by taking  $\pm$  square root of the diagonal elements of the covariance matrix at every time step. How much better is your position estimate versus what you would have had if you used just the measurements alone (*i.e.*, without using a filter at all)?
8. How would you modify the filter above if there were unknown frictional effects on the PABF? What if you knew what the friction was? Would that change how you dealt with it?