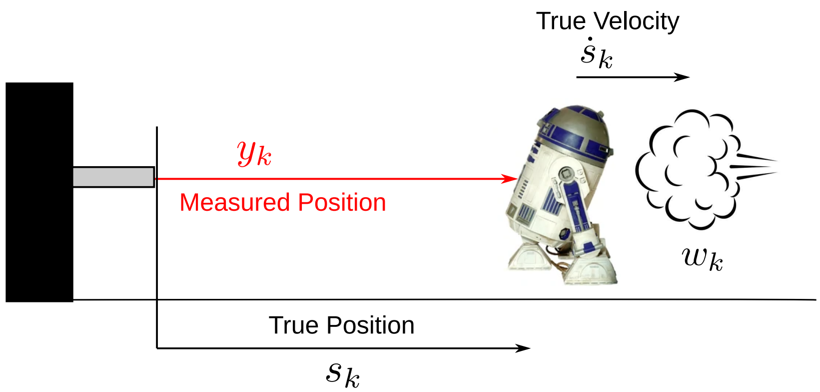
Homework 7 (due at the start of class November 8th, 2022)

The robot R2-D2 is on a mission in the harsh desert world of Tatooine. He leaves his home base to seek out Obi-Wan and his progress is tracked by a laser-based measurement system as shown below.



We assume R2-D2 moves in a straight line and the dynamic model of his motion is derived from Newton's 2nd Law $F = m\ddot{s} = u - b\dot{s}$ where s is the robot's position, $m = 32$ kg is the mass of the robot (source), and $b = 2.5$ N/(m/s) is a linear drag coefficient. Let $T = 0.5$ sec be the sampling time of the laser measurement system. In continuous state-space form we have the dynamics

$$\begin{bmatrix} \dot{s} \\ \ddot{s} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & (-b/m) \end{bmatrix} \begin{bmatrix} s \\ \dot{s} \end{bmatrix} + \begin{bmatrix} 0 \\ (1/m) \end{bmatrix} u \tag{1}$$

which can be discretized as

$$\boldsymbol{F} \approx \mathbf{1} + \boldsymbol{A}T = \begin{bmatrix} 1 & 1+T \\ 0 & 1-(b/m)T \end{bmatrix} \tag{2}$$

$$\boldsymbol{G} \approx \boldsymbol{B}T = \begin{bmatrix} 0 \\ (1/m)T \end{bmatrix} \tag{3}$$

Then the discrete-time dynamics are:

$$\begin{bmatrix} s_k \\ \dot{s}_k \end{bmatrix} = \begin{bmatrix} 1 & 1+T \\ 0 & 1-(b/m)T \end{bmatrix} \begin{bmatrix} s_{k-1} \\ \dot{s}_{k-1} \end{bmatrix} + \begin{bmatrix} 0 \\ (1/m)T \end{bmatrix} u_{k-1} + \boldsymbol{w}_{k-1} \tag{4}$$

$$y_k = [1 \quad 0] \begin{bmatrix} s_k \\ \dot{s}_k \end{bmatrix} + v_k \tag{5}$$

where s_k is the robot's position at time-step k, $\dot{s}_k$ is the robot's velocity at time-step k, and $\boldsymbol{w}_k$ is a model of the random disturbances imparted to the robot by the wind (and other sources of model uncertainty). Note: the winds on this planet combined with low gravity are strong enough to perturb the robot even when it is at rest (i.e., to overcome its static friction). The laser produces a noisy measurement due to inherent sensor limitations and as a result of the sandstorm on the planet. The sensor's noise is modeled as zero-mean, Gaussian, uncorrelated in time, and with variance $E[v_k^2] = 1000$ m ^{2}. The wind disturbance is also modeled as zero-mean, Gaussian, and uncorrelated in time with variance $E[\boldsymbol{w}_k \boldsymbol{w}_k^T] = \mathrm{diag}([1.0 \;\; 0.01]^{\mathrm{T}})$.

1 Problem: Linear Dynamics Uncertainty Propagation

Scenario #1: The laser and communication system suddenly fails and R2D2 is programmed in the event of this failure to immediately cease applying control and coast to a stop to wait for help. You are a member of the search party tasked with determining where R2D2 ended up. Suppose that the last known estimate of his position (at the time of failure) was 500 m with a variance of 100 m^2 and the last known estimate of his speed was 5 m/s with a variance of 0.1 $(\text{m/s})^2$.

- Write a program that estimates the robot's state and covariance k time-steps after the failure event.
- Use your program to compute and plot (on a single set of axes) the probability density functions representing the robot's position at: $t = [0, 10, 20, 30, 60, 120]$ seconds after the instant of communication loss. Each p.d.f. should cover the range of exactly four standard deviations around the mean (similar to Homework 5 Problem 2).

Hints: Your plot should have a collection of six p.d.f. curves with mean centered on the robot's position at each time instant (e.g., the first peak should be at 500 meters and subsequent peaks to the right of that as the robot comes to rest.) Make sure each p.d.f. is plotted over a different range of x values corresponding to $[\mu - 4\sigma, \mu + 4\sigma]$. Collectively, the solution displays all p.d.f.s in a window that spans roughly about 400m to 1000m in the x-axis and 0 to about 0.04 in the (probability) y axis.

2 Problem: Discrete-Time Kalman Filter Implementation

Scenario #2. In this scenario the laser measurement system was restarted and now works fine. A 1 minute log of the laser measurements (`y`) and control inputs (`u`) is provided in the MATLAB file `h7_p2.mat` (along with the true robot state history `x_true`).

```
data =
        y: [1×120 double]
     time: [0 ... ]
        u: [20 20 20 ... ]
   x_true: [2×120 double]
```

1. Process the measurements in the .mat file using your discrete-time Kalman filter function developed in the last homework. Assume the following initial estimate:

$$\hat{\boldsymbol{x}}_{0|0} = [500, 5]^{\mathrm{T}}$$
$$\boldsymbol{P}_{0|0} = \mathbf{1}_{2\times 2} 1E10;$$

   The covariance is very large but this is the default value the laser-based tracker uses after restarting. *Hint: wrap a for loop around the code that performs one iteration of the Kalman filter while appropriately changing the inputs in each loop.*

2. Produce a plot similar to the one on the following page that show that compares the true, estimated, and sensor-based positions, and true and estimated speeds. Include uncertainty bounds indicating $\pm 3\sigma$ uncertainty around the mean of each estimate (or error). A toy example of how to plot transparent uncertainty bounds is provided in the `plotBounds.m` file. Make sure to include a legend and label your axes. Adjust the bounds of each plot to the values provided on the example output using the `ylim` command.

Example output: