EECE 212 Final Project Report

Marcos Rodriguez, Nigel Webb

I. Dynamic systems with disturbances

gen_state.m takes inputs:

- t: which is the time spacing vector from 0 to 60 seconds.
- pv0: a column vector containing the initial x and y positions and velocities.
- sigma_x: x multiplier for standard deviation.
- sigma_y: y multiplier for standard deviation

gen_state outputs:

- x: which is a time-varying state vector.

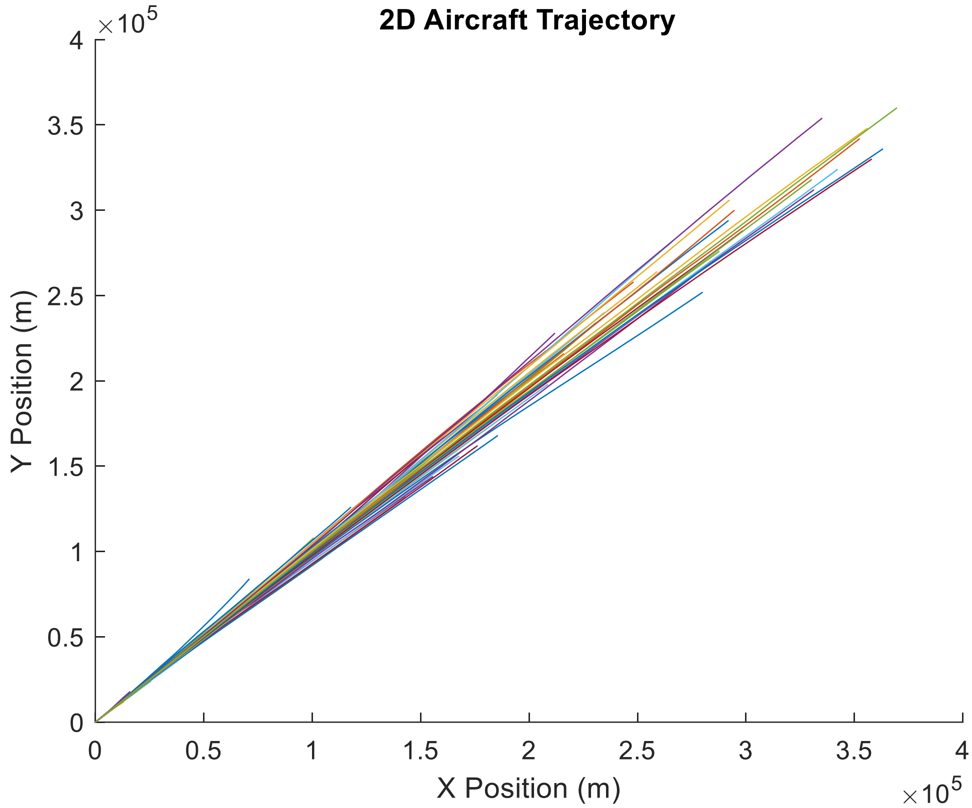
Within gen_state, we defined:

- The state transition matrix A represents a 2D constant-velocity model.
- The input matrix B applies sigma_x and sigma_y multipliers to the x and y velocity states, respectively.
- time_span and time_len are defined to generate a row vector from 0 to 60 and give its length.
- randInp generates two random input row vectors with length time_len = 61.
- x is initialized as the 4 by 61 time varying state vector depending on time_len.
- The first column of x is then filled with the values of pv0 and the rest of the columns are filled by iterating through A*x(:,i-1) + B*randInp(:,i) from 2 to time_len = 61

project_1.m is the plot script for gen_state:

- pv0 is defined as a column vector [0; 100; 0; 100] in the order of [x position, x velocity, y position, y velocity].
- Within project_1, sigma_x and sigma_y are defined as 1 to give variation from a true constant-velocity path.
    - As sigma values are increased, there is more deviance from true constant-velocity path.
- The 2D trajectory is then plotted over t = 0 to 60 by iterating through the function gen_state over those t values.
- Labels and a legend were then added to the plot of the 2D trajectory.

project_1 outputs the figure:



with sigma_x and sigma_y values set equal to 1, initial positions set to zero, and initial velocities set to 100.

II. Using Sensors to Measure States

measure_state.m takes inputs:

- C: which is the general sensor matrix defined in the plot script project_2.m
- noise_lvls: which is a vector containing the noise levels that are to be used as the diagonals of output matrix D.
- state_vectors: holds the sequence of state vectors that are to be measured

measure_state outputs:

- D: which is the time-varying measurement vector

Within measure_state, we defined:

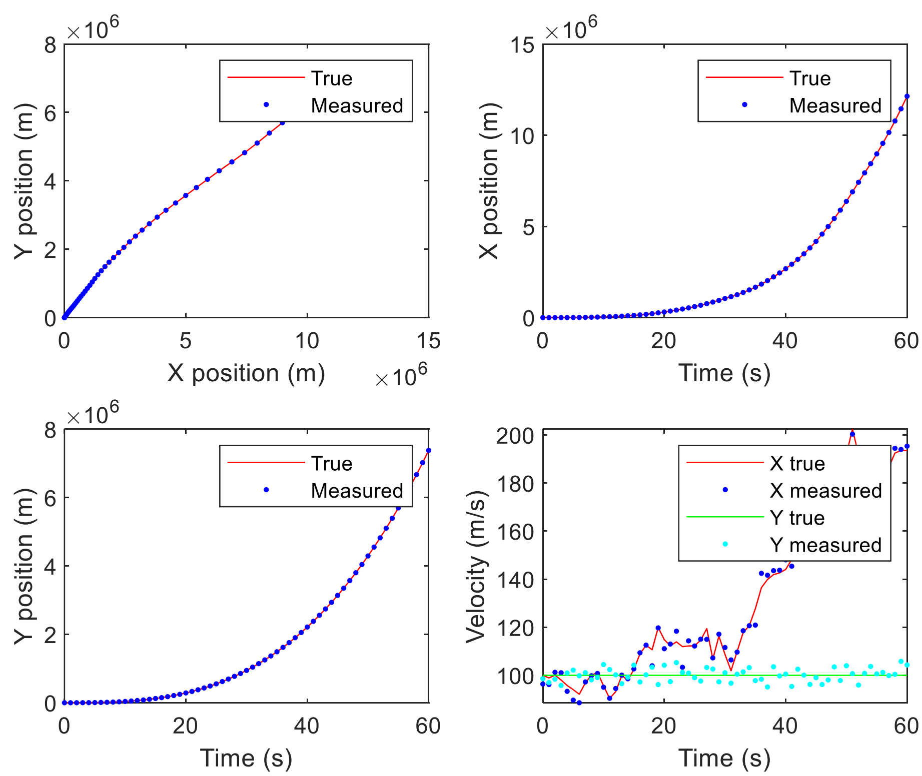
- D was initialized as a matrix with the values of noise_lvls on the diagonals
- The matrix D was then filled with the calculated measured state vectors using the formula Cq[n] + Dw[n] according to the project instructions.
  - Within measure_state, the MATLAB function mvnrnd was used to return the measured state vectors with inputs:

  - Mean vector mu: a zero vector of size length of C by 1
  - Sigma: matrix D taken as input
  - n: size of state_vectors input by 2 number of random vectors
- If noise levels are 0 or less, measure_state returns an error.

project_2.m is the plot script for measure_state:

- The parameters for gen_state and measure_state were defined with:
  - sigma_x = 1, sigma_y = 1, pv0 = [0; 100; 0; 100], t = 0 to 60
- noise_lvls was defined as [25; 10; 25; 10] with order [x position, x velocity, y position, y velocity].
- C was defined as the system matrix for measure_state, which is an identiry matrix of size 4 by 4.
- The output for gen_state is calculated similar to project_1.
- The measured states with noise are calculated by iterating through measure_state with inputs C, noise_lvls, and the output of gen_state for t = 0 to 60.
- Four graphs were then generated, showing:
  - Y-position vs X-position
  - X-position vs time
  - Y-position vs time
  - Velocity v time

project_2 outputs the figure:



With the initial positions set to zero, initial velocities set to 100, sigma_x and sigma_y set to 1, and noise_lvls set to 25 for positions and 10 for velocities.

III. Exploiting the Dynamic Model to Improve Sensor Measurements

RLS_estimation.m takes inputs:

- A: which is the state model matrix.
- R: which is the forgetting factor matrix.
- y: which is the matrix of output vectors with each column being a measurement vector.

RLS_estimation outputs:

- Q_hat: is the matrix of estimated state vectors using RLS. Each column is an estimated measurement vector.

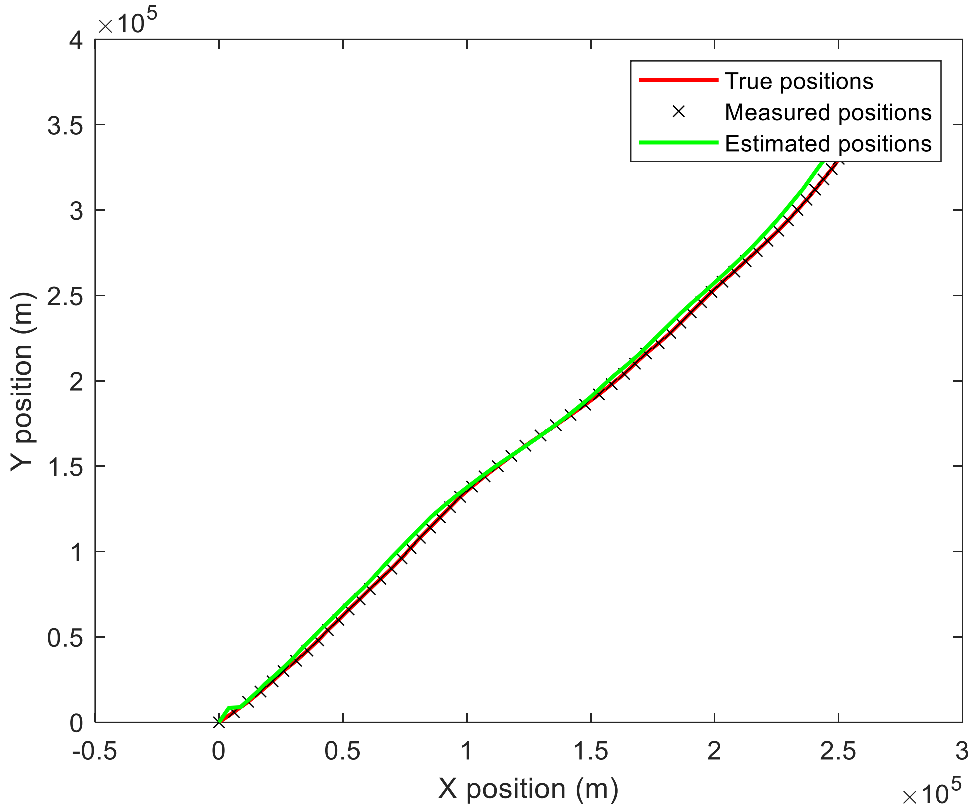
Within RLS_estimation, we defined:

- k as the number of measurements taken from the size of matrix y with dimension two.
- n as the number of states taken from the size of the state model matrix A with dimension one.
- P is defined as the first covariance matrix with identity matrix of size n.
- The initial estimate of Q_hat is initialized as a zero matrix of size n by 1.
- The rest of the function calculates the Kalman gain, covariance matrix, and estimated state vector as according to project instructions over the interval of 1 to k.
- After calculating the above from 1 to k, the intital estimate is removed from the output as it is no longer necessary to compute RLS estimation.

project_3.m is the plot script for RLS_estimation:

- The parameters for gen_state, measure_state, and RLS_estimation are defined as:
    - sigma_x = 5, sigma_y = 5, pv0 = [0; 100; 0; 100], and noise_lvls = [25; 10; 25; 10] with order [x position, x velocity, y position, y velocity].
- The true states were generated with gen_state from time 0 to 60.
- The measured states were generated with measure_state with inputs of a identity matrix of size 4 by 4, the noise_lvls column vector, and the values of the true states from gen_state.
- The RLS filter estimated states were calculated by defining matrix A as an identity matrix of size 4 by 4 and a forgetting factor R of 0.999. These were inputs to RLS_estimation along with the measured states to give the estimated states.
- The true states, measured states, and estimated states were plotted as the true, measured, and estimated positions of 2D trajectory.

project_3 outputs the figure:



With the initial positions set to zero, initial velocities set to 100, sigma_x and sigma_y set to 5, noise_lvls set to 25 for positions and 10 for velocities, and the forgetting factor R set to 0.999.