

# Assignment A5: Kalman Filter (Dynamic Bayes) State Estimation

*CS 4300*  
*Fall 2015*

**Assigned:** 20 October 2015

**Due:** 12 November 2015

For this problem, handin a lab report pdf (include name, date, assignment and class number in pdf) which (a) models a linear trajectory and tracks it with a Kalman Filter, and (b) does the same for a projectile trajectory. The state vector has 4 elements:  $x$ ,  $y$ ,  $\dot{x}$ , and  $\dot{y}$ . The state transition model is:

$$x^{t+1} = x^t + \dot{x}^t \Delta t$$

$$y^{t+1} = y^t + \dot{y}^t \Delta t$$

$$\dot{x}^{t+1} = \dot{x}^t$$

$$\dot{y}^{t+1} = \dot{y}^t - \alpha g$$

where  $\alpha = 0$  in the linear case, and 1 in the projectile case.

For the linear case, run for time  $t = 0$  to  $t_{max} = 10$ , and  $\Delta t$  is 0.1 seconds; set  $\theta$  to  $\frac{\pi}{4}$ .

For the projectile case, run for time  $t = 0$  to  $t_{max} = 35$ , and  $\Delta t$  is 0.1 seconds.

To demonstrate (a), develop a Matlab function with specification:

```
function [x_trace,a_trace,z_trace,Sigma2_trace] =  
CS4300_A5_driver_lin(...  
    x0,y0,vx0,vy0,max_time,del_t,theta)  
% CS4300_A5_driver_lin - driver function for linear Kalman Filter  
% On input:
```

```

%     x0 (float): initial x location
%     y0 (float): initial y location
%     vx0 (float): initial x velocity
%     vy0 (float): initial y velocity
%     max_time (float): max time for tracking
%     del_t (float): time step size
%     theta (float): angle of line (in radians)
% On output:
%     x_trace (nx2 array): each row has estimated pose (x,y vals)
%     a_trace (nx2 array): actual location at each time step
%     z_trace (nx2 array): sensed location at each time step
%     Sigma2_trace (struct array): covariance of estimated location
%         (i).Sigma2 (2x2 array): covariance matrix for i_th step
% Call:
%     [xt,at,zt,St] = CS4300_A5_driver_lin(0,0,1,1,1,0.1,pi/4);
% Author:
%     Your Name
%     UU
%     Fall 2015
%

```

The Kalman Filter function specification is:

```

function [mu_t,Sigma_t] =
CS4300_KF(mu_tm1,Sigma_tm1,u_t,z_t,A_t,R_t,...
    B_t,C_t,Q_t)
% CS4300_KF - one step in Kalman Filter
% On input:
%     mu_tm1 (nx1 vector): previous state estimate vector
%     Sigma_tm1 (nxn matrix): state covariance matrix
%     u_t (nx1 vector): control vector
%     z_t (nx1 vector): measurement vector
%     A_t (nxn matrix): state transition matrix
%     R_t (nxn matrix): state transition covariance matrix
%     C_t (nxn matrix): linear transform for measurement equation
%     Q_t (nxn matrix): noise covariance matrix
% On output:
%     mu_t (nx1 vector): next state estimate
%     Sigma_t (nxn matrix): state covariance matrix
% Call:

```

```

% [x, Sigma2] = CS4300_KF(x, Sigma2, u, z, A, B, R, C, Q);
% Author:
% Your Name
% UU
% Fall 2015
%

```

For (a), develop  $A, B, C, R, Q$  matrixes that yield a result similar to Figure 1.

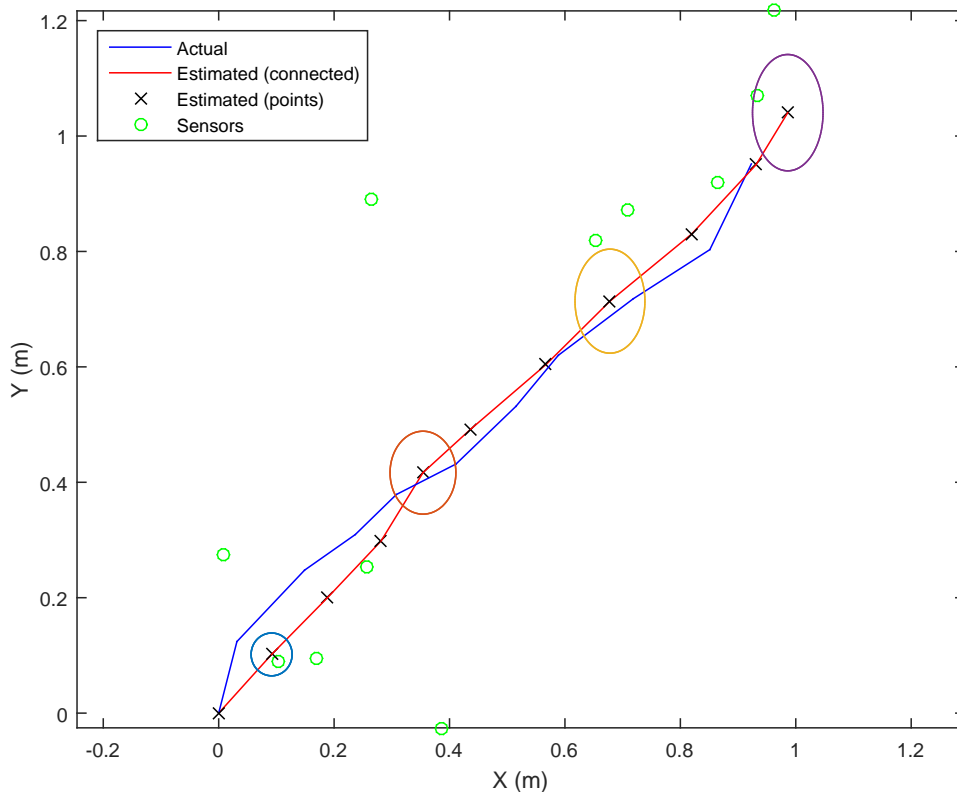


Figure 1: Results of Linear Trajectory KF.

The projectile tracker specification is:

```

function [xt, at, zt, St] = CS4300_driver_proj(dt, t_max, obs_freq, g, r, q)
% CS4300_driver_proj - driver for Kalman Filter projectile tracking
% This script demonstrates the use of the kalman filter in projectile
% tracking.

```

```

%
% A simple model for a 2D projectile is obtained and used for the
% kalman
% filter transition model. The actual trajectory of the projectile is
% simulated using a more complex and accurate model. Observations of
% the
% actual projectile trajectory are obtained. The kalman filter is used
% to
% generate beliefs on the actual projectile state. All three
% trajectories
% (the simple model, the complex model, and the kalman belief) are
% plotted,
% along with the observations made.
% On input:
%   dt (float): simulation time step in seconds (e.g., 0.1)
%   t_max (float): max simulation time in seconds (e.g., 35)
%   obs_freq (float): frequency with which time steps are observed,
%     range [0,1] (e.g., .35)
%   g (float): gravitational acceleration in m/s^2 (9.8)
%   r (float): transition noise; (e.g., 2)
%     increase for faster response to observations
%   q (float): observation noise; increase for greater
%     observation spread (e.g., 10)
% On output:
%   xt (nx4 array): trace of estimated pose
%   at (nx4 array): trace of actual pose
%   zt (mx2 array): trace of sensor data
%   St (struct array): covariance matrix trace
%     (i).Sigma2 (4x4 array): estimated vector covariance
% Call:
%   [xt,at,zt,St] = CS4300_driver_proj(0.1,35,0.35,9.8,2,10);
% Author:
%   Your name
%   UU
%   Fal 2015
%

```

For (b), develop  $A, B, C, R, Q$  matrixes that yield a result similar to Figure 2.

Describe the algorithms in the method section, and verify their correctness.

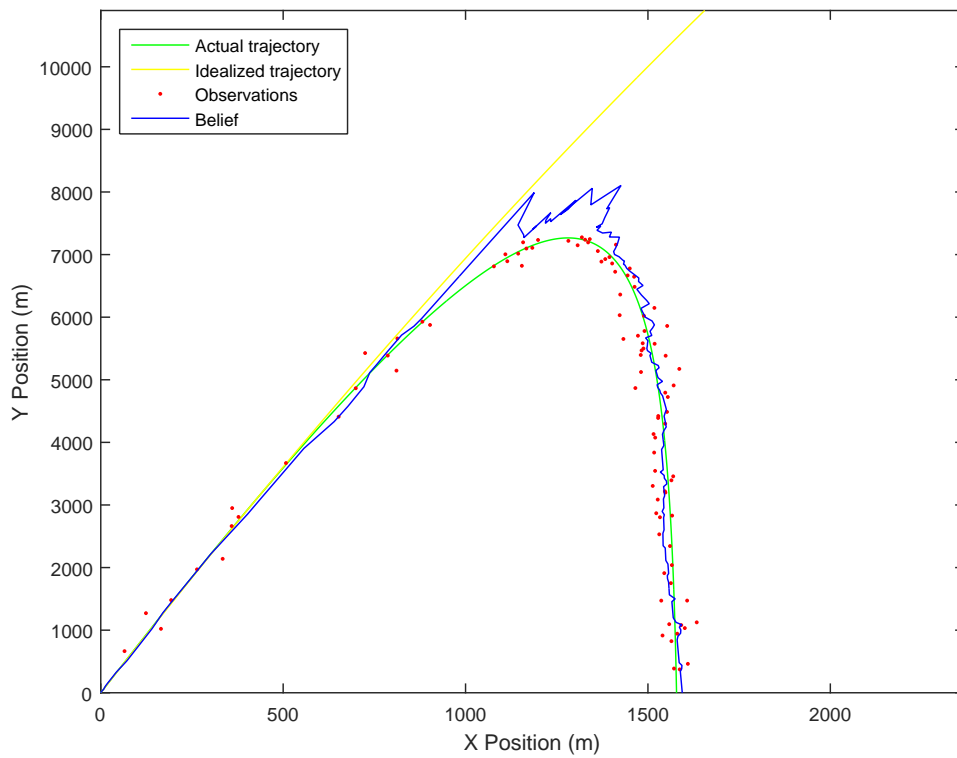


Figure 2: Results of Projectile Trajectory KF.

You should handin the report pdf as well as the source code used in the study. The code should conform to the style requested in the class materials (no matter what the language).

In addition, please turn in a hardcopy of the report in class before the start of class on November 12, 2015.

Write a lab report in the format (please do not deviate from this format!) described in the course materials.

Discuss the statistical framework to establish a confidence interval on the means, and the hypothesis test.