System

insert picture

Physical State

Translation (2D/3D)

-Vector in reference frame (minimal)

Rotation ( 1 DOF, 3 DOF)

-axis w/ angle about that axis

-4 numbers (1 angle, 3 for vector along axis in 3D)

-Quaternian uses 4 numbers

-We can have minimal or redundant representation

-DCM (direction cosine matrix)

-n2 numbers to characterize (using vectors on each point for 3 points on object)

-DCM is minimized with constraints to an orthonormal matrix

The minimal representation of state can be achieved with euler angles

-euler angles in rotation for a 3 DOF state are the time history of inputs

-.

-This is difficult because there is an unknown number of inputs

insert picture showing state updates with inputs

State Estimation

given{u,z}, compute the best guess for x

-we want to describe in terms of probability distribution of x

insert picture of probability labeled with mean, median, and mode

minimum variance state estimation

x: true state

x^ = Xmv = argxmin E[(x-x^)(x-x^)] (minimize the square of the error)

"error" e=x-x^

x^ = argxmin(||e||2), where ||e||2 is a quadratic function

Linear Quadratic Estimator (LQE)

equal to a Kalman filter

insert first picture of Kalman filter

insert second picture of Kalman filter

residual: the difference between expected and actual.

The model can be updated by multiplying by a constant because the model is linear

x^[t+1] = Ax^[t] + Bu[t] + d[t]

d[t] is the process noise. This noise is assumed to be Gaussian normal between 0 and Q.

z^[t] = Cx^[t] + n[t]

n[t] is the measurement noise and is assumed to Gaussian normal between 0 and R.

z~[t] = z[t] - z^[t] = z[t] - Cx^[t]

x~[t] = Kz~[t] (innovation)

x^[t] <-- x^[t] + x~[t]

This model updates twice for each times the system updates once. The first model update is a dynamic update, based on the estimate of the system, and the second model update is the error correction.

A prior:

x^-(t+1) = Ax^+(t) + Bu(t) This is found with the new input.

x^+(t+1) = Ax^-(t) + kz~(t) This is found with the adjustment for the new output.

K is the Kalman filter gain. Its output dimension is x, its input is z.

K = argkmin E[eeT] This is to minimize k, so we take the derivative, set it equal to 0, and then solve for k.

e = x(t) - x^+(t)

Error covariance:

P(t) = E[e(t)e(t)T]

Kalman Filter:

1) Time update

x^-(t+1) = Ax^+(t) + Bu(t)

P-(t+1) = AP+(t)AT+Q

2) Measurement Update

k(t+1) = P-CT(CP-CT+R)-1

x^+=x^- + K(z-cx^-)

The "K(z-cx^-)" term is the Kalman filter gain, times the residual

P+=(I-KC)P-

In order for Kalman filter to be optimal:

1) Everything must be linear

2) All noise must be Gaussian

3) We want to minimize the quadratic error

Excluded Kalman filter (used when the plant is not linear)

1) Linearize the plant if it is not linear

2) Use like a normal Kalman filter