Lecture 11

Wednesday, February 16, 2022 8:59 AM

Admin

① OH Today 3:30 - 4:30pm.

② HW set #1 past due.

Kalman Filtering - Part I

Objectives

① Kalman Filter overview

② State estimation using observers

③ Formulation of Kalman Filters.

What is the Kalman Filter?

- The Kalman Filter is an algorithm that takes a set of noisy, inaccurate measurements (observations) and creates estimates of underlying system's unknown variables.
  - → These estimates accounting for several measurements tend to be more accurate than those based on single, instantaneous ones.
  - → These estimates are based on a joint probability distribution over the system variables, which are often called "states"

    future behaviors depend on past.

  - Where is a Kalman Filter useful? → In estimating the state of noise-corrupted systems.
    - → often for the purpose of control.
      - Ex. 1 Robotic vehicles (trajectory planning)
      - Ex. 2 Aircraft control (navigation)
      - Ex. 3 Bioinstrumentation (EMG control)
      - Ex. 4 Control of immune cells

      Require "sensor fusion" due to low data quality.

  - How does it work?
    - → Kalman Filters are recursive "prediction-error-correction" algorithms much like RLS, and use a weighted average of prior system data with greater weight placed on measurements in which we have higher certainty.
      - Relies on good "observability" (sensing)
      - Usually is a func. of sensor quality, number/type and computing power

— Relies on good observability (sensing)

— usually is a func. of sensor quality, number/type, and computing power.

— what exactly is a system "state"?

— A system's state is a set of variables (usually a minimal set) required to accurately describe its dynamic behavior.

Example: — The behavior of a rocket falling back to Earth.

$$x_t = \begin{bmatrix} p \\ v \end{bmatrix} \begin{array}{l} \text{— position} \\ \text{— velocity.} \end{array}$$

— The immune phenotype of a cell $x_t = \begin{bmatrix} iNOS \\ Arg1 \end{bmatrix}$

State Estimation using observers

- Consider a linear, time-varying, deterministic, dynamic system with a discrete time form:

$$x_{t+1} = \underline{\underline{A_t}} x_t + B_t u_t$$

indices. time step. (~1 msec)

$$p_{t+1} = p_t + v_t \Delta t$$

$$\begin{bmatrix} p_{t+1} \\ v_{t+1} \end{bmatrix} = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} p_t \\ v_t \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} u_t$$

where,

x_t: n-dimensional state vector $\mathbb{R}^{n \times 1}$

u_t: r-dim input vector $\mathbb{R}^{r \times 1}$

A_t: state transition matrix $(n \times n)$

B_t: input matrix $(n \times r)$

The outputs of the system are funcs. of the state vector:

$$y_t = H_t x_t$$

$$y_t = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} p_t \\ v_t \end{bmatrix}$$

where,

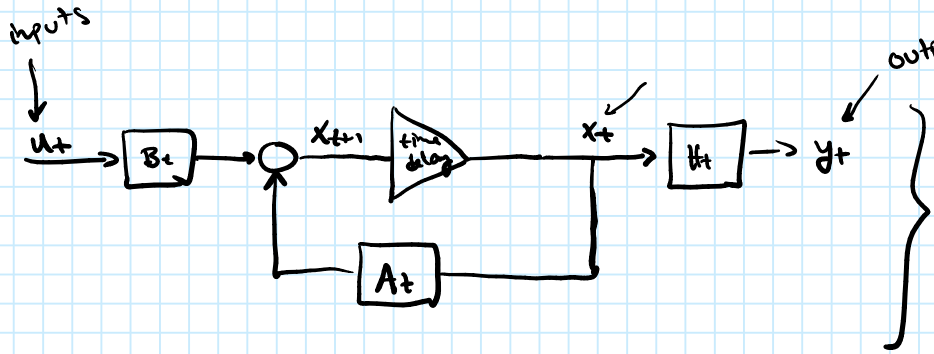
y_t: l-dimensional output vector $\mathbb{R}^{l \times 1}$

A_t: observation matrix in $\mathbb{R}^{l \times n}$

Given matrices, A_t, B_t, and H_t, along w/ initial conditions, x_0, we could simulate the system and predict its next state

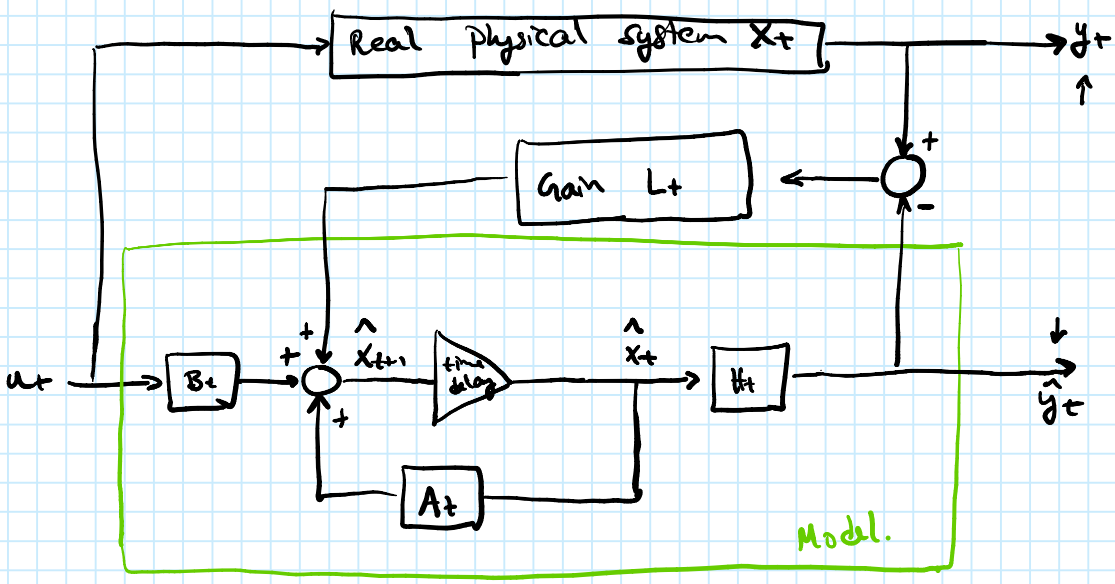
Given matrices, A_t, B_t, and H_t, along w/ initial conditions, x_0, we could simulate the system and predict its next state and set of outputs.

→ In practice, however, model parameters are not exactly known and observed outputs will differ from simulated predictions



Dynamic simulator of a deterministic system.

- For systems with inexact parameter estimates, we can use a real-time simulator called a "dynamic state observer".



(Luenberger's state observer for linear, deterministic systems)

→ Here, discrepancies between the actual & predicted outputs of the system are fed back to the model using gain matrix L_t.

→ The eqns. representing the state observer (estimator) can be written as:

written as:

$$\hat{x}_{t+1} = A_t \hat{x}_t + B_t u_t + L_t (y_t - \hat{y}_t)$$

$$\hat{y}_t = H_t \hat{x}_t$$

$L_t \in \mathbb{R}^{n \times \ell}$ (n: # states, ℓ: # outputs.)

* state estimates are written as $\hat{x}_t$ to distinguish from actual states, x_t.

- If the system is observable (can re-construct state from sensor data.) and a proper gain is chosen, the state estimate will eventually converge to the actual state, regardless of the initial state estimate x_0.

  → The observer "forgets" the initial conditions.

** **Note:** A special case of state observer is the estimation of model parameters θ as in RLS:

→ Setting inputs u_t to zero and replacing the stated transition matrix with an $n \times n$ identity matrix, we get:

$$\hat{x}_{t+1} = \hat{x}_t + L_t (\hat{y}_t - \hat{y}_t), \text{ or}$$

$$\hat{\theta}_{t+1} = \hat{\theta}_t + K_t (y_t - \hat{y}_t)$$

→ state & parameter estimation methods are analogous, both relying on Prediction-Error-Correction methods.