Module 07
Dynamic State Estimation for Dynamical Systems

Ahmad F. Taha

EE 5243: Introduction to Cyber-Physical Systems

Email: ahmad.taha@utsa.edu

Webpage: http://engineering.utsa.edu/~taha/index.html

October 19, 2015
Module 07 Outline

In this module, we will:

1. Introduce dynamic state estimation (DSE)
2. Discuss classes of observers/estimators + Applications
4. Deterministic observers
5. Unknown input observers for linear & nonlinear systems
6. Examples
What is *dynamic state estimation* (DSE)?

- Accurately depicting what’s happening inside a system

Precisely: estimating internal system states

- In circuits: *voltages and currents*
- Water networks: *amount of water flowing*
- Chemical plants: *concentrations*
- Robots and UAVs: *location & speed*
- Humans: *temperature, blood pressure, glucose level*

*So how does having estimates help me?*

- Well, if you have estimates, you can do control
- And if you do good control, you become better off!

In power systems: DSE can tell me what’s happening to generators & lines

⇒ Preventing/Predicting Blackouts!
Observers vs. State Estimators — What’s the Difference?

- Dynamic observer: dynamical system that *observes* the internal system state, given a set of input & output measurements
- State estimator: estimates the system’s states under different assumptions
- Estimators: utilized for state estimation and parametric identification
- Observers: used for deterministic systems, Estimators: for stochastic dynamical systems
- If statistical information on process and measurement is available, stochastic estimators can be utilized
- This assumption is strict for many dynamical systems
- Quantifying distributions of measurement and process noise is very challenging
Current DSE Methods — Stochastic Estimators

- Stochastic estimators:
  - Extended Kalman Filter (EKF)
  - Unscented Kalman filter (UKF)
  - Square-root Unscented Kalman filter (SRUKF)
  - Cubature Kalman Filter (CKF)

  * Stochastic estimators used if distributions of measurement & process noise are available

- System dynamics:

  \[
  x_k = f(x_{k-1}, u_{k-1}) + w_{k-1}
  \]

  \[
  y_k = h(x_k, u_k) + v_k
  \]

  - \( w_{k-1} \sim N(0, Q_{k-1}) \) and \( v_k \sim N(0, R_k) \): process & measurement noise
  - \( Q_{k-1} \) and \( R_k \): covariance of \( q_{k-1} \) & \( r_k \)
Most stochastic estimators have two main steps: predictions & updates

EKF (CLKF+Nonlinearities) algorithm:

(1) Prediction:

State estimate prediction: \( \hat{x}_{k|k-1} = f(\hat{x}_{k-1|k-1}, u_{k-1}) \)

Predicted covariance estimate: \( P_{k|k-1} = F_{k-1} P_{k-1|k-1} F_{k-1}^\top + Q_{k-1} \)

(2) Update:

Innovation or measurement residual: \( \tilde{y}_k = z_k - h(\hat{x}_{k|k-1}) \)

Innovation (or residual) covariance: \( S_k = H_k P_{k|k-1} H_k^\top + R_k \)

Near-optimal Kalman gain: \( K_k = P_{k|k-1} H_k^\top S_k^{-1} \)

Updated covariance estimate: \( P_{k|k} = (I - K_k H_k) P_{k|k-1} \)

Updated state estimate: \( \hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k \tilde{y}_k \)

\( F_{k-1} = \frac{\partial f}{\partial x} \bigg|_{\hat{x}_{k-1|k-1}, u_{k-1}} \), \( H_k = \frac{\partial h}{\partial x} \bigg|_{\hat{x}_{k|k-1}} \)
Current DSE Methods — Deterministic Estimators (Observers)

- Deterministic observers for:
  - LTI systems
  - LTI systems + Unknown Inputs
  - LTI systems + Unknown Inputs + Measurement Noise / Attack Vectors
  - Nonlinear systems (bounded nonlinearity)
  - Nonlinear systems + Unknown Inputs
  - Nonlinear systems + Unknown Inputs + Measurement Noise / Attack Vectors
  - LTI delayed systems
  - LTI delayed systems + Unknown Inputs
  - Hybrid systems
  - ... and many more

* Deterministic estimators used if measurement and process noise distributions are not available
What are Dynamical State Observers?

- Controllers often need values for the full state-vector of the plant
- This is nearly impossible in most complex systems
- *Why?* You simply can’t put sensors everywhere, and some states are unaccessible
- Observer: a dynamical system that **estimates** the states of the system based on the plant’s *inputs* and *outputs* \(^1\)

---

\(^1\)Figure from the 2013 ACC Workshop on: *Robust State and Unknown Input Estimation: A Practical Guide to Design and Applications*, by Stefen Hui and Stanislaw H. Żak.
Luenberger Observer and Plant Dynamics

- **Plant Dynamics:**
  \[
  \dot{x} = Ax + Bu \\
y = Cx, \ x(0) \text{ not given}
  \]

- **Observers Dynamics:**
  \[
  \dot{\hat{x}} = A\hat{x} + Bu + L(y - \hat{y}) \leftarrow \text{Innovation} \\
  \dot{x} = A\hat{x} + Bu + LC(x - \hat{x})
  \]

- **Error dynamics \(^2\):**
  \[
  \dot{e} = \dot{x} - \hat{x} = (A - LC)(x - \hat{x}) \to 0, \text{ as } t \to \infty, \iff \lambda_i(A - LC) < 0
  \]

\(^2\) Figure from the 2013 ACC Workshop on: *Robust State and Unknown Input Estimation: A Practical Guide to Design and Applications*, by Stefen Hui and Stanislaw H. Żak.

© Ahmad F. Taha
Observer-Based Control (OBC)

- After designing an observer for an LTI system, obtain state estimates \( \hat{x}(t) \)
- What to do with \( \hat{x}(t) \)? Well, use it for control \( \Rightarrow \) Observer-Based Control!
- OBC dynamics:

\[
\begin{align*}
\dot{\hat{x}} &= A\hat{x} + \text{Innovation}(y, u) \\
u &= \text{ControlLaw}(v), \quad v = [\hat{x} \quad y \quad r]
\end{align*}
\]
Observer-Based Control — The Equations

- Closed-loop dynamics:

\[
\begin{align*}
\dot{x} &= Ax - BK \hat{x} \\
\dot{\hat{x}} &= A\hat{x} + L(y - \hat{y}) - BK \hat{x}
\end{align*}
\]

- Or

\[
\begin{bmatrix}
\dot{x} \\
\dot{\hat{x}}
\end{bmatrix} =
\begin{bmatrix}
A & -BK \\
LC & A - LC - BK
\end{bmatrix}
\begin{bmatrix}
x \\
\hat{x}
\end{bmatrix}
\]

- Transformation:

\[
\begin{bmatrix}
x \\
e
\end{bmatrix} =
\begin{bmatrix}
x \\
x - \hat{x}
\end{bmatrix} =
\begin{bmatrix}
I & 0 \\
I & -I
\end{bmatrix}
\begin{bmatrix}
x \\
\hat{x}
\end{bmatrix}
\]

- Hence, we can write:

\[
\begin{bmatrix}
\dot{x} \\
\dot{e}
\end{bmatrix} =
\underbrace{egin{bmatrix}
A - BK & BK \\
0 & A - LC
\end{bmatrix}}_{A_{cl}}
\begin{bmatrix}
x \\
e
\end{bmatrix}
\]

- If the system is controllable & observable \(\Rightarrow\) eig\((A_{cl})\) can be arbitrarily assigned by proper \(K\) and \(L\)

- What if the system is stabilizable and detectable?
Unknown Input Observers (UIO) — Why?

- Deterministic observers work well without uncertainties
- Fail to accurately estimate the plant state under uncertainties
- **Solution?** Design of Unknown Input Observers (UIO)
- Unknown input $u_2$ models uncertainties, disturbances or nonlinearities
- **Main idea:** come up with a clever innovation term that nullifies that effect of unknown $u_2$

\[
\begin{align*}
\dot{x}_p &= A_p \hat{x}_p + \text{Innovation}(y, u_1) \\
u_1 &= \text{ControlLaw}(v), \quad v = \begin{bmatrix} \hat{x}_p & y & r \end{bmatrix}
\end{align*}
\]

- $r \rightarrow v \rightarrow u_1 = \text{ControlLaw}(v)$
- $\hat{x}_p, y$
- $\hat{x}_p = A_p \hat{x}_p + \text{Innovation}(y, u_1)$
- $u_1 = \text{ControlLaw}(v), \quad v = \begin{bmatrix} \hat{x}_p & y & r \end{bmatrix}$
- $\dot{x}_p = A_p x_p + B_p^{(1)} u_1 + B_p^{(2)} u_2$
- $u_2$
- $y$
Most Well-Known UIOs

- Different UIOs have been developed:
  - UIOs for LTI systems [Bhattacharyya, 1978]
  - Hui and Žak [Hui & Žak, 2005]
  - Sliding-mode differentiator UIO [Floquet et al., 2006]
  - Hou and Müller observer [Zhang et al., 2012]
  - Observers for Lipschitz nonlinear systems [Chen & Saif, 2006]
  - Walcott-Žak sliding mode observer [Walcott & Žak, 1987]
  - Utkin’s sliding mode observer [Utkin, 1992]

- Some observers have performance guarantees

- Most UIOs have assumptions related to the LTI SS matrices

- We will discuss some UIOs
System and UIO Dynamics — One UIO Architecture

- **Plant Dynamics:**
  \[
  \dot{x}_p = A_p x_p + B^{(1)}_p u_1 + B^{(2)}_p u_2 \\
  y = C_p x_p, \quad x_p(0) \text{ not given}
  \]

- **n states, } m_1 \text{ known inputs, } m_2 \text{ unknown inputs, } p \text{ measurable outputs}

- **UIO Dynamics:**
  \[
  \dot{x}_c = A_c x_c + B^{(1)}_c y + B^{(2)}_c u_1, \\
  \hat{x}_p = x_c + M y,
  \]

- **Error dynamics:**
  \[
  \dot{e} = \dot{x} - \dot{\hat{x}} = (I - MC_p)(A - LC_p)e
  \]

- **Objective:** Design } M, } L, } A_c, } B^{(1)}_c \text{ and } } B^{(2)}_c \text{ such that } e(t) \to 0 \text{ as } t \to \infty

- **Assumptions:**
  1. Pair \((A_p, C_p)\) is detectable
  2. \(\text{rank}(C_p B^{(2)}_p) = \text{rank}(B^{(2)}_p)\) — rank matching condition implies that there must be at least as many independent outputs as unknown inputs
  3. \(x_c(0) = (I - MC_p)v, \ v \text{ is arbitrary vector}\)
UIO Design

- We want to estimate \( x_p \)
- The presented observer assumes unknown initial plant conditions
- UIO is motivated by writing \( x_p \) as:

\[
x_p = (I - MC_p)x_p + MC_p x_p = (I - MC_p)x_p + My
\]

- **Objective**: analyze the unknown portion of \( x_p \), that is \( x_c = (I - MC_p)x_p \)
- We then have: \( \dot{x}_c = (I - MC_p)\dot{x}_p + \text{AddedConvergenceTerm} \)
- Then, obtain \( \hat{x}_p = x_c + My \)
- Design matrix parameters such that unknown input \( u_2 \) is nullified [Hui & Žak, 2005]
UIO Design — 2

- **UIO Dynamics** [Hui & Žak, 2005] (recall that \( x_p = x_c + My \)):

\[
\dot{x}_c = (I - MC_p)\dot{x}_p + \text{AddedConvergenceTerm} \\
= (I - MC_p)\left( A_p x_p + B_p^{(1)} u_1 + B_p^{(2)} u_2 \right) + \text{AddedConvergenceTerm} \\
= (I - MC_p)\left( A_p x_c + A_p M y + B_p^{(1)} u_1 + \underbrace{L(y - C_p x_c - C_p M y)}_{\text{AddedConvergenceTerm}} \right) \\
\dot{x}_c = A_c x_c + B_c^{(1)} y + B_c^{(2)} u_1, \\
\hat{x}_p = x_c + M y,
\]

where:

* \((I - MC_p)B_p^{(2)} = 0\)
* \(A_c = (I - MC_p)(A_p - LC_p), B_c^{(2)} = (I - MC_p)B_p^{(1)}\)
* \(B_c^{(1)} = (I - MC_p)(A_p M + L - LC_p M)\)
UIO Design Parameters

- Given $A_p, B_p^{(1)}, B_p^{(2)}, C_p$, find $M, L$ such that $e(t) \rightarrow 0$ as $t \rightarrow \infty$

- Precisely, $M \in \mathbb{R}^{n \times p}$ is chosen such that

$$\left( I - MC_p \right) B_p^{(2)} = 0$$

- Solution:

$$M = B_p^{(2)} \left( (C_p B_p^{(2)})^\dagger + H_0 \left( I_p - (C_p B_p^{(2)}) (C_p B_p^{(2)})^\dagger \right) \right)$$

- $H_0$ is a design matrix

- $L$ is an added gain to improve the convergence of the estimated state $(\hat{x}_p)$

- **Note:** the above solution encapsulates the projection nature of $MC_p$: $(MC_p)^2 = MC_p$ and hence $I - MC_p$ is also a projection

- Basically, nullifying the unknown input by $(I - MC_p)$

- **Note:** This UIO design can be easily extended to reduced-order designs; read [Hui & Žak, 2005] for more
Numerical Results for the UIO

- Given a stable LTI MIMO system with 2 known, 2 unknown inputs, 4 outputs.
- Unknown inputs are all $u_2(t) = 0.5 \sin(t)$, SS matrices:

$$A_p = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ -1 & -5 & -10 & -10 & -5 \end{bmatrix}, \quad B_p^{(1)} = B_p^{(2)} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & -1 \\ 1 & 0 \\ 0 & 0 \end{bmatrix}, \quad C_p = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

- UIO state estimates converge to the actual states.
Sliding Mode Observers — Introduction

- Sliding model control: nonlinear control method whose structure depends on the current state of the system
- Sliding mode observers (SMO): nonlinear observers driving state trajectories of estimation error to zero or to a bounded neighborhood
- SMOs have high resilience to measurement noise
- See [Utkin, 1992] for more on SMOs
System and SMO Dynamics — Second UIO Architecture

- **Plant Dynamics:**
  \[
  \dot{x}_p = A_p x_p + B_p^{(1)} u_1 + B_p^{(2)} u_2 \\
y = C_p x_p
  \]

  * **Assumption:** unknown input \( u_2 \) is bounded, i.e., \( \|u_2\| \leq \rho \)

- **SMO Dynamics** [[Hui & Žak, 2005]]:
  \[
  \dot{x}_p = A_p \hat{x}_p + B_p^{(1)} u_1 + L(y - \hat{y}) - B_p^{(2)} E(\hat{y}, y, \eta) \\
  \hat{y} = C_p \hat{x}_p,
  \]

- \( u_1 \) and \( y \): readily available signals for the SMO

- \( E(\cdot) \) is defined as (\( \eta \) is SMO gain):
  \[
  E(\hat{y}, y, \eta) = \begin{cases} 
  \eta \frac{F(\hat{y} - y)}{\|F(\hat{y} - y)\|_2}, & \text{if } F(\hat{y} - y) \neq 0 \\
  0, & \text{if } F(\hat{y} - y) = 0.
  \end{cases}
  \]

- **SMO design objective:** find matrices \( F \) and \( L \)
SMO Design

- $F \in \mathbb{R}^{m_2 \times p}$ satisfies: $FC_p = (B_p^{(2)})^T P$

- $L$ is chosen to guarantee the asymptotic stability of $A_p - LC_p$

- Thus, for $Q = Q^T > 0$, there is a unique $P = P^T > 0$ such that $P$ satisfies:

  $$(A_p - LC_p)^T P + P(A_p - LC_p) = -Q, \quad P = P^T > 0$$

- $E(\cdot)$ guarantees that $e(t)$ is insensitive to the unknown input $u_2(t)$ and the estimation error converges asymptotically to zero

  * If for the chosen $Q$, no matrix $F$ satisfies the above equality, another matrix $Q$ can be chosen

  * A design algorithm (to find matrices $F, L, P$) is presented in [Hui & Žak, 2005]
The SMO design problem boils down to solving matrix equalities

Can we solve the matrix design problem using LMIs? Yes!

We have two (nonlinear) matrix equations in terms of $P, F, L$:

$$(A_p - LC_p)^\top P + P(A_p - LC_p) = -Q$$

$$P = P^\top$$

$$FC_p = (B_p^{(2)})^\top P$$

LMI trick: set $Y = PL$, rewrite above system of linear matrix equations as:

$$A_p^\top P + PA_p - C_p^\top Y^\top - YC_p = -Q$$

$$P = P^\top$$

$$FC_p = (B_p^{(2)})^\top P$$
SMO Design Using CVX

Sample CVX code:

```cvx
cvx_clear

cvx_begin sdp quiet

variable P(n, n) symmetric
variable Y(n, p)
variable F(m2, p)

minimize(0)

subject to

Ap'*P + P*Ap - Y*Cp - Cp'*Y' <= 0
F*Cp-Bp2'*P==0;
P >= 0

cvx_end

L = P\Y;
```
Numerical Example

- Linearized dynamics of a power system:

\[
A = \begin{bmatrix}
-41 & 0 & 0 & 0 \\
27.67 & -16.67 & -55.33 & 0 \\
0 & 0.01 & -0.01 & 0 \\
0 & 0 & 1 & 0 \\
\end{bmatrix}, \quad B_1 = \begin{bmatrix}
2 \\
0 \\
0 \\
0 \\
\end{bmatrix}, \quad B_2 = \begin{bmatrix}
2 \\
0 \\
0 \\
0 \\
\end{bmatrix}, \quad C^\top = \begin{bmatrix}
1 & 0 \\
0 & 0 \\
0 & 0 \\
0 & 1 \\
\end{bmatrix}
\]

- Solving for \( P, L, F \) using CVX, we obtain:

\[
L = \begin{bmatrix}
-28.22 & -0.12 \\
12.23 & -39.15 \\
-0 & 5.92 \\
0.05 & 3.76 \\
\end{bmatrix}, \quad F = \begin{bmatrix}
4.89 & 0.42 \\
\end{bmatrix}, \quad P = \begin{bmatrix}
2.45 & 0 & 0 & 0.21 \\
0 & 0.19 & 0.36 & 0.36 \\
0 & 0.36 & 11.43 & -15.01 \\
0.21 & 0.36 & -15.01 & 43.62 \\
\end{bmatrix}
\]

- After simulating the observer, we obtain:

©Ahmad F. Taha
Dynamic Observers for NL Systems — Architecture # 1

- **Question**: What if system dynamics are nonlinear?
- **Answer**: Use deterministic estimators for nonlinear systems

System dynamics:

\[
\dot{x} = Ax + B_1u_1 + \phi(x, u) + B_2u_2
\]

- **Nonlinear term in the dynamics** \(\phi(x, u)\) is:
  - Globally Lipschitz (*Lipschitz Continuous*):
    \[
    \|\phi(x, u) - \phi(z, u)\| \leq L\|x - z\|, \quad L \geq 0
    \]
  - One-sided Lipschitz:
    \[
    \langle \phi(x, u) - \phi(z, u), x - z \rangle \leq k_1\|x - z\|^2
    \]
  - Quadratically inner-bounded:
    \[
    (\phi(x, u) - \phi(z, u))^\top (\phi(x, u) - \phi(z, u)) \leq k_2\|x - z\|^2 + k_3 \langle \phi(x, u) - \phi(z, u), x - z \rangle
    \]
    * Lipschitz continuity ⇒ quadratic inner-boundedness
    * Example: if \(\phi(x) = \sin(x)\), then \(L = 1\)
Finding Lipschitz Constants — Examples

- **Example 1:** if \( \phi(x) = x^2 \), what is the Lipschitz constant \( L \) if \( x \) is defined on the interval \([-2, 2]\)?

  - **Solution:** applying the definition, we have:

  \[
  \|\phi(x_2) - \phi(x_1)\| = |x_2^2 - x_1^2| = |x_2 - x_1||x_2 + x_1| \leq 4|x_2 - x_1| \Rightarrow L = 4
  \]

- **Example 2:** find \( L \) if \( \phi(x) = \begin{bmatrix} ax_1 + bx_2 \\ 1 - \cos(cx_1) \end{bmatrix} \), \( x \in \mathbb{R}^2_+ \) and \( a, b, c \in \mathbb{R}_+ \)

  - **Solution:**

  \[
  \|\phi(x) - \phi(z)\| = \left\| \begin{bmatrix} ax_1 + bx_2 \\ 1 - \cos(cx_1) \end{bmatrix} - \begin{bmatrix} az_1 + bz_2 \\ 1 - \cos(cz_1) \end{bmatrix} \right\| \\
  = \left\| \begin{bmatrix} a(x_1 - z_1) + b(x_2 - z_2) \\ \cos(cz_1) - \cos(cx_1) \end{bmatrix} \right\| = \left\| \begin{bmatrix} a(x_1 - z_1) + b(x_2 - z_2) \\ -2 \sin(0.5c(z_1 + x_1)) \sin(0.5c(z_1 - x_1)) \end{bmatrix} \right\| \\
  \leq \left\| \begin{bmatrix} a(x_1 - z_1) + b(x_2 - z_2) \\ 2 \sin(0.5c(x_1 - z_1)) \end{bmatrix} \right\| \leq \left\| \begin{bmatrix} a(x_1 - z_1) + b(x_2 - z_2) \\ c(x_1 - z_1) \end{bmatrix} \right\| \\
  = \left\| \begin{bmatrix} a & b \\ c & 0 \end{bmatrix} \begin{bmatrix} x_1 - z_1 \\ x_2 - z_2 \end{bmatrix} \right\| = \left\| \begin{bmatrix} a & b \\ c & 0 \end{bmatrix} \right\| \|x - z\| \leq \left\| \begin{bmatrix} a & b \\ c & 0 \end{bmatrix} \right\| \|x - z\| \\
  \leq \sqrt{2} \left\| \begin{bmatrix} a & b \\ c & 0 \end{bmatrix} \right\|_\infty \|x - z\| \Rightarrow L = \sqrt{2} \max(a + b, c)
  \]
Observer Design

- **Plant dynamics under unknown inputs:**
  \[
  \dot{x} = Ax + B_1 u_1 + \phi(x, u) + B_2 u_2 \\
  y = Cx
  \]

- **Observer dynamics [Zhang et al., 2012]:**
  \[
  \dot{\hat{x}} = A\hat{x} + B_1 u_1 + \phi(\hat{x}, u) + L(y - C\hat{x})
  \]

- **Matrix-gain \(L\) determined as follows:**
  1. Given \(k_1, k_2, k_3\), solve this LMI for \(\epsilon_1, \epsilon_2, \sigma\) and \(P = P^\top \succeq 0\):
     \[
     \begin{bmatrix}
     A^\top P + PA + (\epsilon_1 k_1 + \epsilon_2 k_2)I_n - \sigma C^\top C & P + \frac{k_3 \epsilon_2 - \epsilon_1}{2} I_n \\
     \left(P + \frac{k_3 \epsilon_2 - \epsilon_1}{2} I_n\right)^\top & -\epsilon_2 I_n
     \end{bmatrix} < 0
     \]
  2. Compute observer gain \(L\):
     \[
     L = \frac{\sigma}{2} P^{-1} C^\top
     \]

- **Extension:** reduced-order DSE

- Read [Zhang et al., 2012] to understand the derivation of the above LMI
Simulation Example

- Nonlinear power system, consider Lipschitz parameters: \( \rho = \varphi = \mu = 1 \)
- Using CVX, we solve for \( P, \epsilon_1, \epsilon_2 \) and \( \sigma \):
  \[ \epsilon_1 = 0.0122, \epsilon_2 = 0.0144, \sigma = 6.424, \]
- Then, the observer gain-matrix \( L \) is computed:

\[
P = \begin{bmatrix}
0.4894 & -0.017 & 0.062 & -0.46 \\
-0.01 & 0.005 & 0 & 0.006 \\
0.062 & 0 & 0.77 & 0.02 \\
-0.46 & 0.006 & 0.02 & 0.49
\end{bmatrix},
\]

\[
L = \frac{\sigma}{2} P^{-1} C^\top = \begin{bmatrix}
-6.02 & 15.93 & 31.86 & 12.04 \\
-15.74 & 42.50 & 85.02 & 31.503 \\
4.20 & 0.06 & 0.12 & -8.46 \\
-3.11 & 8.69 & 17.39 & 6.23
\end{bmatrix}
\]

- Given \( L \), plot the observer response given random estimator initial conditions:
Here, we introduce an observer design for a specific class of nonlinear systems with unknown inputs.

Observer design based on the methods presented in [Chen & Saif, 2006].

Observer design assumes:
1. $B_2$ is full-column rank
2. Nonlinear function is Lipschitz

The design problem is formulated as an SDP.
Observer Design for NL Systems

- System dynamics:
  \[
  \dot{x} = Ax + B_1u_1 + \phi(x) + B_2u_2 \\
y = Cx
  \]

- Proposed observer dynamics:
  \[
  \dot{z} = Nz + Gu + Ly + M\phi(\hat{x}) \\
  \hat{x} = z - Ey
  \]

* Matrices $E, K, N, G, L$ and $M$ are obtained from the matrix equalities that ensure the asymptotic stability of estimation error

* Lipschitz constant $\gamma$: $\|\phi(x_1) - \phi(x_2)\| \leq \gamma\|x_1 - x_2\|$

- Authors in [Chen & Saif, 2006] develop matrix equations that guarantee $e = x - \hat{x}$ converges to zero

- Can you re-derive the equations? Design matrix parameters s.t. $e \to 0$

- Read [Chen & Saif, 2006] to understand the design algorithm
Observer Design Algorithm for NL Systems

**Algorithm 1** Observer with Unknown Input Design Algorithm

1: **given** parameters: $A, B_1, B_2, C$ and $\gamma$ (the Lipschitz constant)
2: **compute** matrices $U, V, \bar{A}$ and $\bar{B}_1$:

\[
U = -B_2(CB_2)^\dagger \\
V = I - (CB_2)(CB_2)^\dagger \\
\bar{A} = (I + UC)A \\
\bar{B}_1 = VCA
\]

3: **find** matrices $\bar{Y}, \bar{K}$ and a symmetric positive definite matrix $P$ that are a solution for this LMI:

\[
\begin{bmatrix}
\Psi_{11} & \Psi_{12} \\
\Psi_{12}^\top & I_{2n}
\end{bmatrix} < 0
\]

where

\[
\Psi_{11} = \bar{A}^\top P + P\bar{A} + \bar{B}_1^\top \bar{Y}^\top \bar{Y}\bar{B}_1 - C^\top \bar{K}^\top - \bar{K}C + \gamma I,
\]

\[
\Psi_{12} = \sqrt{\gamma} \left( P(I + UC) + \bar{Y}(VC) \right)
\]

4: **obtain** matrices $Y$ and $K$ and the observer parameters $N, G, L$ and $M$:

\[
Y = P^{-1}\bar{Y}, \quad K = P^{-1}\bar{K} \\
E = U + YV, \quad M = I + EC \\
N = MA - KC, \quad G = MB_1 \\
L = K(I + CE) - MAE
\]

5: **simulate** the UIO given the computed matrices
SMO Design Using CVX

\[
[p \ n] = \text{size}(C);\ [n \ m1] = \text{size}(B1);\ [n \ m2] = \text{size}(B2);
U = -B2*\text{pinv}(C*B2);\ V = \text{eye}(\text{length}(C*B2))-(C*B2)*\text{pinv}(C*B2);
\]

\[
\text{cvx_begin sdp quiet}
\text{variable}\ P(n,n)\ \text{symmetric}
\text{variable}\ Ybar(n,p)
\text{variable}\ Kbar(n,p)
\minimize(1)
\text{subject to}
P \geq 0;
-
[((\text{eye}(n)+U*C)*A)'*P + P*((\text{eye}(n)+U*C)*A) + \ldots
(V*C*A)’*Ybar’ + Ybar*(V*C*A) - C’*Kbar’ - Kbar*C + \ldots
\gamma*\text{eye}(\text{length}(Kbar*C)) , \sqrt{\gamma}*(P*(\text{eye}(n)+U*C)+Ybar*(V*C));
(\sqrt{\gamma}*(P*(\text{eye}(n)+U*C)+Ybar*(V*C)))’ , -\text{eye}(n)] \geq 0;
\text{cvx_end}
\]

\[
Y = \text{inv}(P)*Ybar;\ K = \text{inv}(P)*Kbar;
E = U+Y*V;\ M = \text{eye}(n)+E*C;
N = M*A-K*C;\ G = M*B1;
L = K*(\text{eye}(p)+C*E)-M*A*E;
\]
Numerical Example [Chen & Saif, 2006]

- Consider this dynamical system:

\[
A = \begin{bmatrix}
-1 & -1 & 0 \\
-1 & 0 & 0 \\
0 & -1 & -1
\end{bmatrix},
B_1 = 0, B_2 = \begin{bmatrix}
-1 \\
0 \\
0
\end{bmatrix},
C = \begin{bmatrix}
1 & 0 & 0 \\
0 & 0 & 1
\end{bmatrix}^T,
\phi = \begin{bmatrix}
0.5 \sin(x_2) \\
0.6 \cos(x_3) \\
0
\end{bmatrix},
u_2 = 2 \sin(5t)
\]

- Applying the algorithm, we obtain:

\[
U = \begin{bmatrix}
-1 \\
0 \\
0
\end{bmatrix},
V = \begin{bmatrix}
0 & 0 \\
0 & 1
\end{bmatrix},
P = \begin{bmatrix}
50.25 & 0 & 0 \\
0 & 0.89 & 0 \\
0 & 0 & 50.25
\end{bmatrix},
Y = \begin{bmatrix}
0 & 0 \\
0 & 1.3874 \\
0 & -50.25
\end{bmatrix}
\]

- Compute matrices \( K, E, M, N, G, L \) and simulate the observer.

- Converging estimates:
Comparison between DSE Techniques

<table>
<thead>
<tr>
<th>Functionality/Characteristic</th>
<th>Kalman Filter Derivatives</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>EKF</td>
</tr>
<tr>
<td><strong>System’s Nonlinearities</strong></td>
<td>×</td>
</tr>
<tr>
<td><strong>Feasibility</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Tolerance to Different Initial Conditions</strong></td>
<td>×</td>
</tr>
<tr>
<td><strong>Tolerance to Unknown Inputs</strong></td>
<td>×</td>
</tr>
<tr>
<td><strong>Tolerance to Cyber-Attacks</strong></td>
<td>×</td>
</tr>
<tr>
<td><strong>Tolerance to Process &amp; Measurement Noise</strong></td>
<td>✓</td>
</tr>
<tr>
<td><strong>Guaranteed Convergence</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Numerical Stability</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Computational Complexity</strong></td>
<td>$\mathcal{O}(n^3)$</td>
</tr>
</tbody>
</table>
Questions And Suggestions?

Any questions?

Thank You!

Please visit
engineering.utsa.edu/~taha

IFF you want to know more 😊


