Adaptive Control: Introduction, Overview, and Applications

Eugene Lavretsky, Ph.D.

E-mail: eugene.lavretsky@boeing.com
Phone: 714-235-7736

Robust & Adaptive Control Theory Workshop
AIAA GN&C, Aug 14 – 15, 2004

Course Overview

- **Motivating Example**
- **Review of Lyapunov Stability Theory**
  - Nonlinear systems and equilibrium points
  - Linearization
  - Lyapunov's direct method
  - Barbalat's Lemma and Lyapunov-like Lemma
- **Model Reference Adaptive Control**
  - Basic concepts
  - 1^{st} order systems
  - n^{th} order systems
  - Robustness to Parametric / Non-Parametric Uncertainties
- **Neural Networks, (NN)**
  - Architectures
  - Using sigmoids
  - Using Radial Basis Functions, (RBF)
- **Adaptive NeuroControl**
- **Design Example: Adaptive Reconfigurable Flight Control using RBF NN-s**

References

- J-J. E. Slotine and W. Li, *Applied Nonlinear Control*, Prentice-Hall, New Jersey, 1991
- S. Haykin, *Neural Networks: A Comprehensive Foundation*, 2nd edition, Prentice-Hall, New Jersey, 1999
- H. K., Khalil, *Nonlinear Systems*, 2nd edition, Prentice-Hall, New Jersey, 2002

Motivating Example

(Model Reference Adaptive Control of an Unknown Mass)

- Plant dynamics: $m\,\ddot{x} = u$
  - u – motor force, (control input)
  - x – position
  - m – unknown mass
- Reference Model: $\ddot{x}_m + \lambda_1\,\dot{x}_m + \lambda_2\,x_m = r(t)$
  - $r(t)$ – desired positioning command
  - tracking error: $e_x(t) = x(t) - x_m(t)$
- Control Input: $u = \hat{m}\left(\ddot{x}_m - 2\alpha\,\dot{e}_x - \alpha^2\,e_x\right), \quad \alpha > 0$
- Error Dynamics, *if mass was known*, $(\hat{m} = m)$

  $$\ddot{e}_x + 2\alpha\,\dot{e}_x + \alpha^2\,e_x = 0$$

  - since $\alpha > 0$ we get: $\lim\limits_{t\to\infty} e_x(t) = 0$

Motivating Example
(continued)

- Unknown mass: $\hat{m} \neq m$
  - need an on-line mass estimator
  - mass estimation error: $e_m(t) = \hat{m}(t) - m$
  - control input: $u = \hat{m}\underbrace{\left(\ddot{x}_m - \alpha_1 \dot{e}_x - \alpha_2 e_x\right)}_{v(t)} = m\,v(t) + e_m\,v(t)$
  - error dynamics:

$$m\left(\ddot{e}_x + 2\alpha\,\dot{e}_x + \alpha^2 e_x\right) = e_m\,v$$

$$m\underbrace{\left(\underbrace{\frac{d}{dt}\left(\dot{e}_x + \alpha\,e_x\right)}_{\dot{s}} + \alpha\underbrace{\left(\dot{e}_x + \alpha\,e_x\right)}_{s}\right)}_{\dot{s}+\alpha s} = e_m\,v \longrightarrow m\left(\dot{s} + \alpha\,s\right) = e_m\,v$$

$$V\left(s, e_m\right) = m\,s^2 + \gamma^{-1} e_m^2$$

  - Lyapunov's direct method yields adaptive law:

$$\dot{\hat{m}} = -\gamma\,v\,s, \quad \gamma > 0 \longrightarrow \lim_{t\to\infty} s(t) = 0 \longrightarrow \lim_{t\to\infty} e_x(t) = 0$$

Lyapunov Stability Theory

Alexander Michailovich Lyapunov 1857-1918

- Russian mathematician and engineer who laid out the foundation of the Stability Theory
- Results published in 1892, Russia
- Translated into French, 1907
- Reprinted by Princeton University, 1947
- American Control Engineering Community Interest, 1960’s

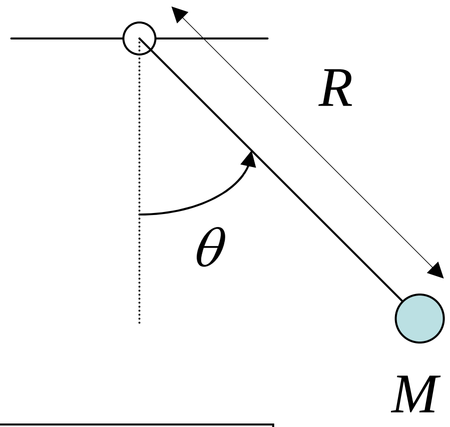
Nonlinear Dynamic Systems and Equilibrium Points

- A nonlinear dynamic system can usually be represented by a set of *n* differential equations in the form: $\dot{x} = f(x,t), \quad \text{with } x \in R^n, t \in R$
  - x is the state of the system
  - t is time
- If f does not depend *explicitly* on time then the system is said to be *<u>autonomous</u>*: $\dot{x} = f(x)$
- A state x_e is an equilibrium if once $x(t) = x_e$, it remains equal to x_e for all future times: $0 = f(x)$

Example: Equilibrium Points of a Pendulum

- System dynamics: $M R^2 \ddot{\theta} + b\dot{\theta} + M g R \sin(\theta) = 0$
- State space representation, $\left(x_1 = \theta, \quad x_2 = \dot{\theta}\right)$

$$\dot{x}_1 = x_2$$
$$\dot{x}_2 = -\frac{b}{M R^2} x_2 - \frac{g}{R}\sin(x_1)$$



- Equilibrium points:

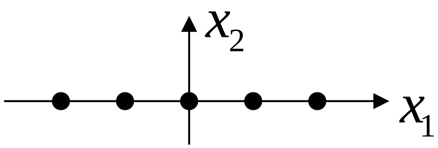
$$0 = x_2$$
$$0 = -\frac{b}{M R^2} x_2 - \frac{g}{R}\sin(x_1)$$

$\Rightarrow$ $x_2 = 0, \quad \sin(x_1) = 0$

$\Downarrow$

$$x_e = \begin{pmatrix} \pi k \\ 0 \end{pmatrix}, \quad \left(k = 0, \pm 1, \pm 2, \ldots\right)$$

$\Leftarrow$



Example: Linear Time-Invariant (LTI) Systems

- LTI system dynamics: $\dot{x} = A x$
  - has a single equilibrium point (the origin 0) if *A* is nonsingular
  - has an infinity of equilibrium points in the null-space of *A*: $A x_e = 0$
- LTI system trajectories: $x(t) = \exp\left(A(t - t_0)\right) x(t_0)$
- If *A* has all its eigenvalues in the left half plane then the system trajectories converge to the origin exponentially fast

State Transformation

- Suppose that x_e is an equilibrium point
- Introduce a new variable: $y = x - x_e$
- Substituting for $x = y + x_e$ into $\boxed{\dot{x} = f(x)}$
- New system dynamics: $\boxed{\dot{y} = f(y + x_e)}$
- New equilibrium: $y = 0$, (since $f(x_e) = 0$)
- *Conclusion*: study the behavior of the new system in the neighborhood of the origin

Nominal Motion

- Let $x^*(t)$ be the solution of $\dot{x} = f(x)$
  - the nominal motion trajectory corresponding to initial conditions $x^*(0) = x_0$
- Perturb the initial condition $x(0) = x_0 + \delta x_0$
- Study the stability of the motion error: $e(t) = x(t) - x^*(t)$
- The error dynamics:
  - non-autonomous!

$$\dot{e} = f\left(x^*(t) + e(t)\right) - f\left(x^*(t)\right) = g(e,t)$$
$$e(0) = \delta x_0$$

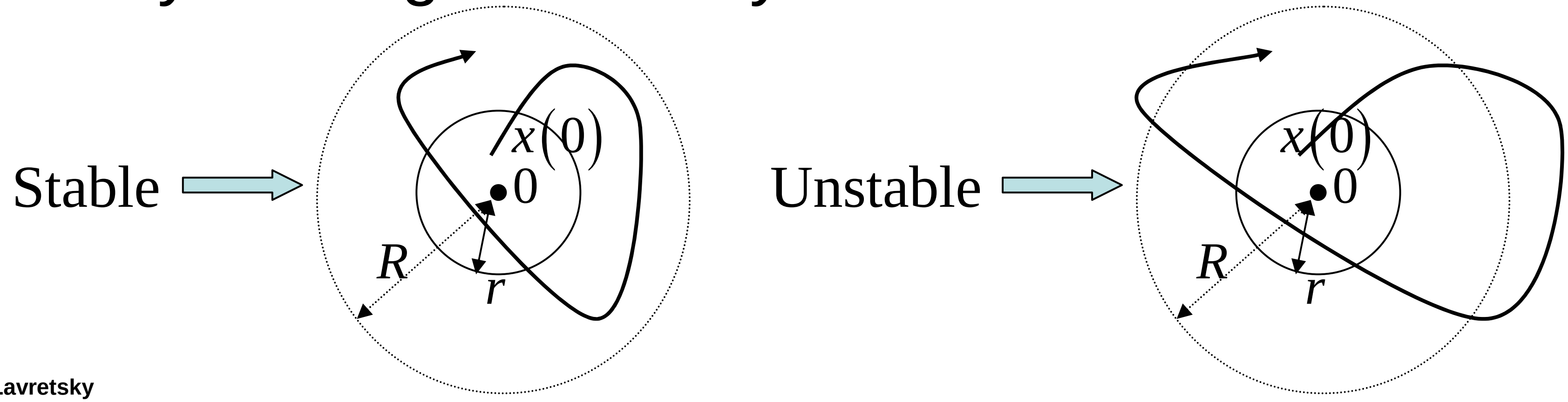
- *Conclusion*: Instead of studying stability of the nominal motion, study stability of the error dynamics w.r.t. the origin

Lyapunov Stability

- **Definition**: The equilibrium state *x* = 0 of autonomous nonlinear dynamic system is said to be *stable* if:

$$\forall R > 0, \quad \exists r > 0, \quad \left\{ \|x(0)\| < r \right\} \Rightarrow \left\{ \forall t \geq 0, \|x(t)\| < R \right\}$$

- Lyapunov Stability means that the system trajectory can be kept arbitrary close to the origin by starting sufficiently close to it



Asymptotic Stability

- **Definition**: An equilibrium point 0 is *asymptotically stable* if it is stable and if in addition: $\exists r > 0, \quad \left\{ \|x(0)\| < r \right\} \Rightarrow \left\{ \lim_{t\to\infty} \|x(t)\| = 0 \right\}$
- Asymptotic stability means that the equilibrium is stable, and that in addition, states started close to 0 actually converge to 0 as time *t* goes to infinity
- Equilibrium point that is stable but not asymptotically stable is called *marginally stable*

Exponential Stability

- **Definition**: An equilibrium point 0 is <u>*exponentially stable*</u> if:

$$\exists\ r, \alpha, \lambda > 0, \quad \forall \left\{ \left\| x(0) \right\| < r \ \wedge\ t > 0 \right\} : \quad \left\| x(t) \right\| \le \alpha \left\| x(0) \right\| e^{-\lambda t},$$

- The state vector of an exponentially stable system converges to the origin faster than an exponential function
- Exponential stability implies asymptotic stability

Local and Global Stability

- **Definition**: If asymptotic (exponential) stability holds for any initial states, the equilibrium point is called globally asymptotically (exponentially) stable.
- Linear time-invariant (LTI) systems are either exponentially stable, marginally stable, or unstable. Stability is always global.
- Local stability notion is needed only for nonlinear systems.
- **Warning**: State convergence does not imply stability!

Lyapunov's 1st Method

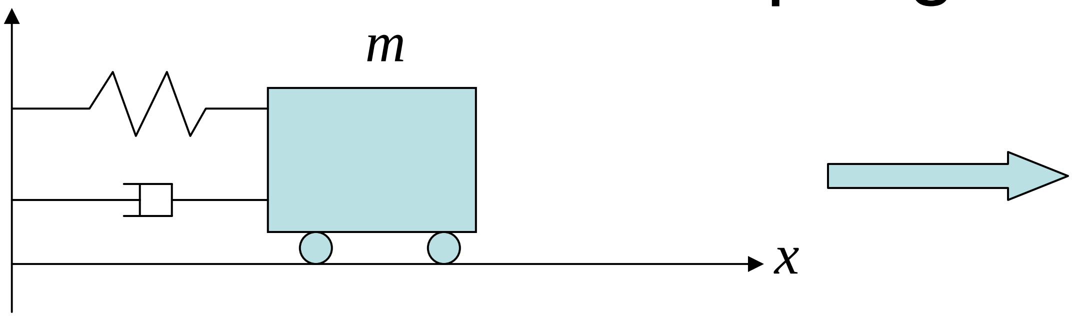
- Consider autonomous nonlinear dynamic system: $\dot{x} = f(x)$
- Assume that *f*(*x*) is continuously differentiable
- Perform linearization: $$\dot{x} = \underbrace{\left(\frac{\partial f(x)}{\partial x}\right)_{x=0}}_{A} x + \underbrace{f_{h.o.t.}(x)}_{\text{higher-order terms}} \cong A x$$
- **Theorem**
  - If A is Hurwitz then the equilibrium is asymptotically stable, (locally!)
  - If A has at least one eigenvalue in right-half complex plane then the equilibrium is unstable
  - If A has at least one eigenvalue on the imaginary axis then one cannot conclude anything from the linear approximation

Lyapunov's Direct (2nd) Method

- **Fundamental Physical Observation**
  - If the total *energy* of a mechanical (or electrical) system is continuously dissipated, then the system, *whether linear or nonlinear*, must eventually settle down to an equilibrium point.
- **Main Idea**
  - Analyze stability of an *n*-dimensional dynamic system by examining the variation of a single *scalar* function, (system energy).

Lyapunov's Direct Method (Motivating Example)

- Nonlinear mass-spring-damper system



$$\underbrace{m\ddot{x} + b\,\dot{x}\left|\dot{x}\right|}_{\text{damping}} + \underbrace{k_0\,x + k_1\,x^3}_{\text{spring term}} = 0$$

- **Question**: If the mass is pulled away and then released, will the resulting motion be stable?
  – Stability definitions are hard to verify
  – Linearization method fails, (linear system is only marginally stable

Lyapunov's Direct Method (Motivating Example, continued)

- Total mechanical energy

$$V(x) = \underbrace{\frac{1}{2} m \dot{x}^2}_{\text{kinetic}} + \underbrace{\int_0^x \left(k_0 x + k_1 x^3\right) dx}_{\text{potential}} = \frac{1}{2} m \dot{x}^2 + \frac{1}{2} k_0 x^2 + \frac{1}{4} k_1 x^4$$

- Total energy rate of change along the system's motion:

$$\dot{V}(x) = m \dot{x} \ddot{x} + \left(k_0 x + k_1 x^3\right) \dot{x} = \dot{x}\left(-b \dot{x} |\dot{x}|\right) = -b |\dot{x}|^3 \le 0$$

- ***Conclusion***: Energy of the system is dissipated until the mass settles down: $\dot{x} = 0$

Lyapunov's Direct Method (Overview)

- Method
  - based on generalization of energy concepts
- Procedure
  - generate a scalar "energy-like function (*Lyapunov function*) for the dynamic system, and examine its variation in time, (derivative along the system trajectories)
  - if energy is dissipated (derivative of the Lyapunov function is non-positive) then conclusions about system stability may be drawn

Positive Definite Functions

- **Definition**: A scalar continuous function $V(x)$ is called *locally positive definite* if

$$V(0) = 0 \wedge \{\forall x \neq 0 \wedge \|x\| < R\} \Rightarrow V(x) > 0$$

- If $V(0) = 0 \wedge \{\forall x \neq 0\} \Rightarrow V(x) > 0$ then $V(x)$ is *globally positive definite*
- Remarks
  – a positive definite function must have a unique minimum $\min_{x \in B_R} V(x) = V(x_{\min}) = V_{\min}$
  – if $V_{\min}$ =/= 0 or $x_{\min}$ =/= 0 then use

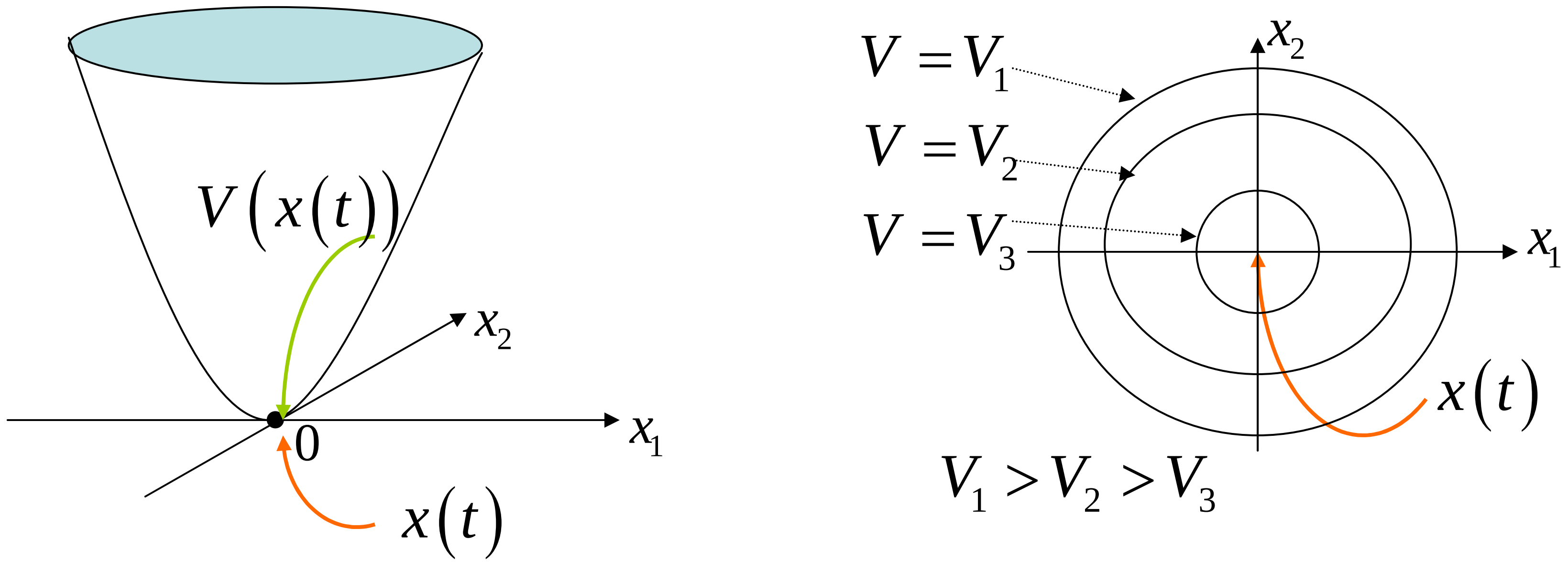
$$W(x) = V(x - x_{\min}) - V_{\min}$$

Lyapunov Functions

- **Definition**: If in a ball B_R the function $V(x)$ is positive definite, has continuous partial derivatives, and if its time derivative along any state trajectory of the system $\dot{x} = f(x)$ is negative semi-definite, i.e., $\dot{V}(x) \le 0$ then $V(x)$ is said to be a *Lyapunov function* for the system.
- *Time derivative* of the Lyapunov function

$$\dot{V}(x) = \nabla V(x)\, f(x) \le 0, \quad \nabla V(x) = \begin{pmatrix} \dfrac{\partial V(x)}{\partial x_1} & \dots & \dfrac{\partial V(x)}{\partial x_n} \end{pmatrix} \in R^n$$

Lyapunov Function (Geometric Interpretation)



- Lyapunov function is a bowl, (locally)
- $V(x(t))$ always moves down the bowl
- System state moves across contour curves of the bowl towards the origin

Lyapunov Stability Theorem

- If in a ball B_R there exists a scalar function $V(x)$ with continuous partial derivatives such that $\boxed{\forall x \in B_R : \quad V(x) > 0 \wedge \dot{V}(x) \le 0}$ then the equilibrium point 0 is ***stable***
  - If the time derivative is locally negative definite $\boxed{\dot{V}(x) < 0}$ then the stability is ***asymptotic***
    - If $V(x)$ is radially unbounded, i.e., $\lim\limits_{\|x\| \to \infty} V(x) = \infty$, then the origin is ***globally asymptotically stable***
- $V(x)$ is called the Lyapunov function of the system

Example: Local Stability

- Pendulum with viscous damping: $\ddot{\theta} + \dot{\theta} + \sin\theta = 0$
- State vector: $x = \begin{pmatrix} \theta & \dot{\theta} \end{pmatrix}^T$
- Lyapunov function candidate: $V(x) = (1 - \cos\theta) + \dfrac{\dot{\theta}^2}{2}$
  - represents the total energy of the pendulum
  - locally positive definite
  - time-derivative is *negative semi-definite*

$$\dot{V}(x) = \frac{\partial V(x)}{\partial \theta}\dot{\theta} + \frac{\partial V(x)}{\partial \dot{\theta}}\ddot{\theta} = \dot{\theta}\sin\theta + \dot{\theta}\underbrace{\ddot{\theta}}_{-\dot{\theta}-\sin\theta} = -\dot{\theta}^2 \le 0$$

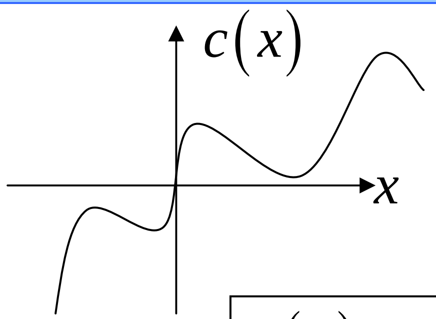
- ***Conclusion***: System is locally stable

Example: Asymptotic Stability

- System Dynamics: $$\begin{aligned}\dot{x}_1 &= x_1\left(x_1^2 + x_2^2 - 2\right) - 4\, x_1\, x_2^2 \\ \dot{x}_2 &= x_2\left(x_1^2 + x_2^2 - 2\right) + 4\, x_1^2\, x_2\end{aligned}$$
- Lyapunov function candidate: $V\left(x_1, x_2\right) = x_1^2 + x_2^2$
  - positive definite
  - time-derivative is *negative definite* in the 2-dimensional ball defined by $x_1^2 + x_2^2 < 2$

  $$\dot{V}\left(x_1, x_2\right) = 2\left(x_1^2 + x_2^2\right)\left(x_1^2 + x_2^2 - 2\right) < 0$$
- <u>*Conclusion*</u>: System is *locally* asymptotically stable

Example: Global Asymptotic Stability



- Nonlinear 1st order system

$$\dot{x} = -c(x), \quad \text{where: } x\,c(x) > 0$$

- Lyapunov function candidate:

$$V(x) = x^2$$

  - globally positive definite
  - time-derivative is negative definite

$$\dot{V}(x) = 2\,x\,\dot{x} = -2\,x\,c(x) < 0$$

- ***Conclusion***: System is globally asymptotically stable
- **Remark**: Trajectories of a 1st order system are monotonic functions of time, (why?)

La Salle's Invariant Set Theorems

- It often happens that the time-derivative of the Lyapunov function is only negative *semi*-definite
- It is still possible to draw conclusions on the *asymptotic* stability
- Invariant Set Theorems (attributed to La Salle) extend the concept of Lyapunov function

Example: 2nd Order Nonlinear System

- System dynamics: $\ddot{x} + b(\dot{x}) + c(x) = 0$
  - where *b*(*x*) and *c*(*x*) are continuous functions verifying the sign conditions: $\dot{x}\,b(\dot{x}) > 0, \text{ for } \dot{x} \neq 0$ and $x\,c(x) > 0, \text{ for } x \neq 0$
- Lyapunov function candidate: $V(x, \dot{x}) = \frac{1}{2}\dot{x}^2 + \int_0^x c(y)\,dy$
  - positive definite
  - time-derivative is negative *semi*-definite

$$\dot{V} = \dot{x}\,\ddot{x} + c(x)\,\dot{x} = -\dot{x}\,b(\dot{x}) \le 0$$

  - system energy is dissipated $\dot{x}\,b(\dot{x}) = 0 \Leftrightarrow \dot{x} = 0 \Rightarrow \ddot{x} = -c(x)$
  - system cannot get "stuck" at a non-zero equilibrium
- <u>*Conclusion*</u>: Origin is globally asymptotically stable

Lyapunov Functions for LTI Systems

- LTI system dynamics: $\dot{x} = A\,x$
- Lyapunov function candidate: $V(x) = x^T P\,x$
  - where *P* is symmetric positive definite matrix
  - function V(x) is positive definite
- Time-derivative of *V*(*x*(*t*)) along the system trajectories: $\dot{V}(x) = \dot{x}^T P\,x + x^T P\,\dot{x} = x^T \underbrace{\left(A^T P + P\,A\right)}_{-Q} x = -x^T Q\,x < 0$
  - where Q is symmetric positive definite matrix
  - Lyapunov equation: $A^T P + P\,A = -Q$
- Stability analysis procedure:
  - choose a symmetric positive definite Q
  - solve the Lyapunov equation for P
  - check whether P is positive definite

Stability of LTI Systems

- **Theorem**
  - An LTI system is stable (globally exponentially) if and only if for any symmetric positive definite matrix Q, the unique matrix solution P of the Lyapunov equation is symmetric and positive definite
- **Remark**: In most practical cases Q is chosen to be a diagonal matrix with *positive* diagonal elements

Barbalat’s Lemma: Preliminaries

- Invariant set theorems of La Salle provide asymptotic stability analysis tools for *autonomous* systems with a negative *semi*-definite time-derivative of a Lyapunov function
- Barbalat’s Lemma extends Lyapunov stability analysis to *non-autonomous* systems, (such as adaptive model reference control)

Barbalat's Lemma

- **Lemma**
  - If a differentiable function $f(t)$ has a finite limit as $t \to \infty$ and if $\dot{f}(t)$ is ***uniformly continuous***, then $\lim_{t\to\infty} \dot{f}(t) = 0$
- **Remarks**
  - *uniform continuity* of a function is difficult to verify directly
  - simple *sufficient condition*:
    - if derivative is bounded then function is uniformly continuous
  - The fact that derivative goes to zero does not imply that the function has a limit, as t tends to infinity. The converse is also not true, (in general)
  - Uniform continuity condition is very important!

Example: LTI System

- **Statement**: Output of a stable LTI system is uniformly continuous in time
  - System dynamics: $\dot{x} = A\,x + B\,u$
  - Control input *u* is bounded
  - System output: $y = C\,x$
- **Proof**: Since *u* is bounded and the system is stable then *x* is bounded. Consequently, the output time-derivative $\dot{y} = C\,\dot{x} = C\left(A\,x + B\,u\right)$ is bounded. Thus, (using Barbalat's Lemma), we conclude that the output *y* is <u>*uniformly continuous*</u> in time.

Lyapunov-Like Lemma

- If a scalar function $V(x,t)$ satisfies the following conditions
  - function is lower bounded
  - its time-derivative along the system trajectories is negative semi-definite and uniformly continuous in time
- Then: $\lim_{t \to \infty} \dot{V}(x,t) = 0$
- **Question**: Why is this fact so important?
- **Answer**: It provides *theoretical* foundations for *stable* adaptive control design

Example: Stable Adaptation

- Closed-loop error dynamics of an adaptive system $\dot{e} = -e + \theta\, w(t),\ \dot{\theta} = -e\, w(t)$
  - where *e* is the tracking error, θ is the parameter error, and *w*(*t*) is a bounded continuous function
- Stability Analysis
  - Consider Lyapunov function candidate: $V(e,\theta) = e^2 + \theta^2$
    - it is positive definite
    - its time-derivative is negative semi-definite $\dot{V}(e,\theta) = 2e(-e + \theta w) + 2\theta(-e w) = -2e^2 \le 0$
    - consequently, *e* and θ are bounded
    - since $\ddot{V}(e,\theta) = -4e(-e + \theta w)$ is bounded, $\dot{V}(e,\theta)$ is uniformly continuous
    - hence: $\lim_{t\to\infty}(-2e^2) = \lim_{t\to\infty}\dot{V}(e,\theta) = 0 \Rightarrow \lim_{t\to\infty} e(t)$

Adaptive Control

Introduction

- Basic Ideas in Adaptive Control
  - estimate uncertain plant / controller parameters on-line, while using measured system signals
  - use estimated parameters in control input computation
- Adaptive controller is a dynamic system with on-line parameter estimation
  - inherently nonlinear
  - analysis and design rely on the Lyapunov Stability Theory

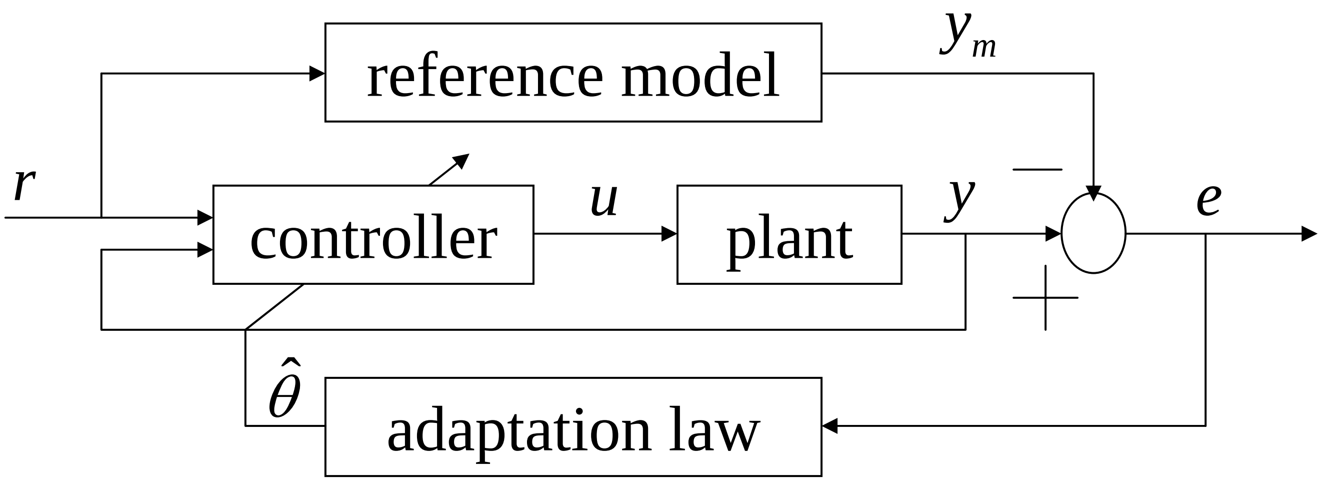
Historical Perspective

- Research in adaptive control started in the early 1950's
  – autopilot design for high-performance aircraft
- Interest diminished due to the crash of a test flight
  – Question: X-?? aircraft tested
- Last decade witnessed the development of a coherent theory and many practical applications

Concepts

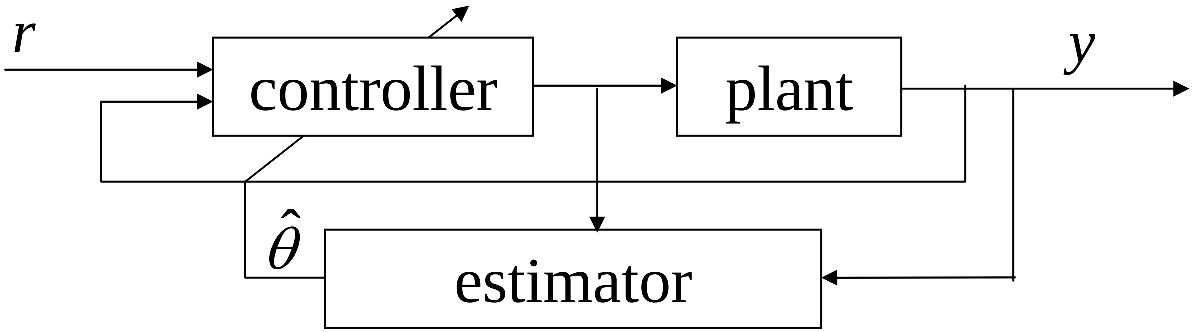
- **Why Adaptive Control?**
  - dealing with complex systems that have unpredictable parameter deviations and uncertainties
- **Basic Objective**
  - maintain consistent performance of a system in the presence of uncertainty and variations in plant parameters
- Adaptive control is superior to robust control in dealing with uncertainties in constant or slow-varying parameters
- Robust control has advantages in dealing with disturbances, quickly varying parameters, and unmodeled dynamics
- ***Solution***: Adaptive augmentation of a Robust Baseline controller

Model-Reference Adaptive Control (MRAC)



- *Plant* has a known structure but the parameters are unknown
- *Reference model* specifies the ideal (desired) response y_m to the external command r
- *Controller* is parameterized and provides tracking
- *Adaptation* is used to adjust parameters in the control law

Self-Tuning Controllers (STC)



- Combines a controller with an on-line (recursive) plant parameter estimator
- Reference model can be added
- Performs simultaneous parameter identification and control
- Uses ***Certainty Equivalence Principle***
  - controller parameters are computed from the estimates of the plant parameters as if they were the true ones

Direct vs. Indirect Adaptive Control

- Indirect
  - estimate plant parameters
  - compute controller parameters
  - relies on convergence of the estimated parameters to their true unknown values
- Direct
  - no plant parameter estimation
  - estimate controller parameters (gains) only
- MRAC and STC can be designed using both Direct and Indirect approaches
- *We consider Direct MRAC design*

MRAC Design of 1st Order Systems

- System Dynamics: $\dot{x} = a\,x + b\left(u - f(x)\right)$
  - a, b are constant *unknown* parameters
  - *uncertain* nonlinear function: $f(x) = \sum_{i=1}^{N} \theta_i\,\varphi_i(x) = \theta^T\,\Phi(x)$
    - vector of constant *unknown* parameters: $\theta = \begin{pmatrix}\theta_1 & \dots & \theta_N\end{pmatrix}^T$
    - vector of known basis functions: $\Phi(x) = \begin{pmatrix}\varphi_1(x) & \dots & \varphi_N(x)\end{pmatrix}^T$
- Stable Reference Model: $\dot{x}_m = a_m\,x_m + b_m\,r, \quad (a_m < 0)$
- **Control Goal**
  - find u such that: $\lim_{t\to\infty}\left(x(t) - x_m(t)\right) = 0$

MRAC Design of 1st Order Systems (continued)

- Control Feedback: $u = \hat{k}_x\, x + \hat{k}_r\, r + \hat{\theta}^T \Phi(x)$
  - (N + 2) parameters to estimate on-line: $\hat{k}_x, \hat{k}_r, \hat{\theta}$
- Closed-Loop System: $\dot{x} = \left(a + b\,\hat{k}_x\right) x + b\left(\hat{k}_r\, r + \left(\hat{\theta} - \theta\right)^T \Phi(x)\right)$
- Desired Dynamics: $\dot{x}_m = a_m\, x_m + b_m\, r$
- Matching Conditions Assumption
  - there exist *ideal* gains (k_x, k_r) such that: $a + b\,k_x = a_m$, $b\,k_r = b_m$
  - *Note*: knowledge of the ideal gains is not required, only their existence is needed
  - consequently:

$$a + b\,\hat{k}_x - a_m = a + b\,\hat{k}_x - a - b\,k_x = b\left(\hat{k}_x - k_x\right) = b\,\Delta k_x$$

$$b\,\hat{k}_r - b_m = b\,\hat{k}_r - b\,k_r = b\left(\hat{k}_r - k_r\right) = b\,\Delta k_r$$

MRAC Design of 1st Order Systems (continued)

- Tracking Error: $e(t) = x(t) - x_m(t)$
- Error Dynamics:

$$\dot{e}(t) = \dot{x}(t) - \dot{x}_m(t) = \left(a + b\,\hat{k}_x\right)x + b\left(\hat{k}_r\,r + \underbrace{\left(\hat{\theta} - \theta\right)^T}_{\Delta\theta}\Phi(x)\right) - a_m\,x_m - b_m\,r \pm a_m\,x$$

$$= a_m\left(x - x_m\right) + \left(a + b\,\hat{k}_x - a_m\right)x + b\left(\hat{k}_r - k_r\right)r + b\,\Delta\theta^T\,\Phi(x)$$

$$= a_m\,e + b\left(\Delta k_x\,x + \Delta k_r\,r + \Delta\theta^T\,\Phi(x)\right)$$

- Lyapunov Function Candidate:

$$V\left(e(t), \Delta k_x(t), \Delta k_r(t), \Delta\theta(t)\right) = e^2 + |b|\left(\gamma_x^{-1}\,\Delta k_x^2 + \gamma_r^{-1}\,\Delta k_r^2 + \Delta\theta^T\,\Gamma_\theta^{-1}\,\Delta\theta\right)$$

  – where: $\gamma_x > 0, \gamma_r > 0$, and $\Gamma = \Gamma^T > 0$ is symmetric positive definite matrix

MRAC Design of 1st Order Systems (continued)

- Time-derivative of the Lyapunov function

$$
\begin{aligned}
&\dot{V}\left(e,\Delta k_x,\Delta k_r,\Delta\theta\right)=2\,e\,\dot{e}+2\left|b\right|\left(\gamma_x^{-1}\,\Delta k_x\,\dot{\hat{k}}_x+\gamma_r^{-1}\,\Delta k_r\,\dot{\hat{k}}_r+\Delta\theta^T\,\Gamma_\theta^{-1}\,\dot{\hat{\theta}}\right)\\
&=2\,e\left(a_m\,e+b\left(\Delta k_x\,x+\Delta k_r\,r\right)+\Delta\theta^T\,\Phi\left(x\right)\right)\\
&+2\left|b\right|\left(\gamma_x^{-1}\,\Delta k_x\,\dot{\hat{k}}_x+\gamma_r^{-1}\,\Delta k_r\,\dot{\hat{k}}_r+\Delta\theta^T\,\Gamma_\theta^{-1}\,\dot{\hat{\theta}}\right)\\
&=2\,a_m\,e^2+2\left|b\right|\left(\Delta k_x\left(x\,e\,\mathrm{sgn}\left(b\right)+\gamma_x^{-1}\,\dot{\hat{k}}_x\right)\right)\\
&+2\left|b\right|\left(\Delta k_r\left(r\,e\,\mathrm{sgn}\left(b\right)+\gamma_r^{-1}\,\dot{\hat{k}}_r\right)\right)+2\,\left|b\right|\Delta\theta^T\left(\Phi\left(x\right)e\,\mathrm{sgn}\left(b\right)+\Gamma_\theta^{-1}\,\dot{\hat{\theta}}\right)
\end{aligned}
$$

MRAC Design of 1st Order Systems (continued)

- Adaptive Control Design Idea
  - Choose adaptive laws, (on-line parameter updates) such that the time-derivative of the Lyapunov function decreases along the error dynamics trajectories

$$\dot{\hat{k}}_x = -\gamma_x \, x \, e \operatorname{sgn}(b)$$
$$\dot{\hat{k}}_r = -\gamma_r \, r \, e \operatorname{sgn}(b)$$
$$\dot{\hat{\theta}} = -\Gamma_\theta \, \Phi(x) \, e \operatorname{sgn}(b)$$

- Time-derivative of the Lyapunov function becomes semi-negative definite!

$$\dot{V}\left(e(t), \Delta k_x(t), \Delta k_r(t), \Delta\theta(t)\right) = 2 \underbrace{a_m}_{<0} e(t)^2 \le 0$$

MRAC Design of 1st Order Systems (continued)

- Closed-Loop System Stability Analysis
  - Since $V \geq 0 \text{ and } \dot{V} \leq 0$ then all the parameter estimation errors are bounded
  - Since the true (unknown) parameters are constant then all the estimated parameters are bounded
- Assumption
  - reference input r(t) is bounded
- Consequently, x_m and $\dot{x}_m$ are bounded

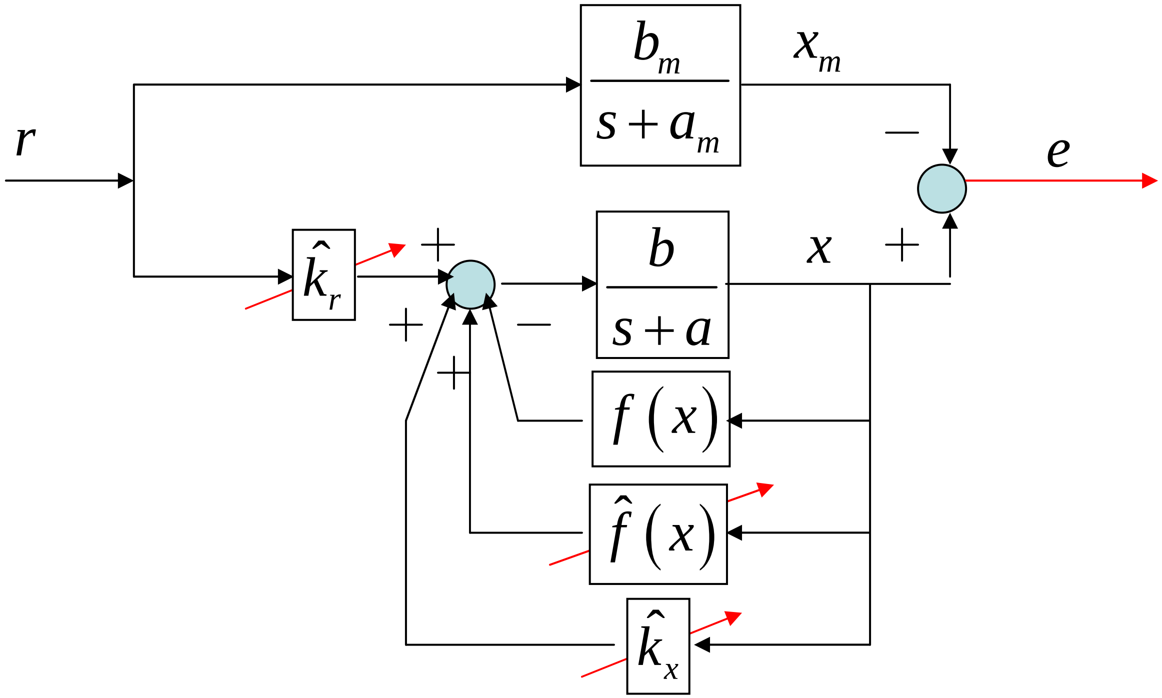
MRAC Design of 1st Order Systems (continued)

- Since $x = e + x_m$ then x is bounded
- Consequently, the adaptive control feedback *u* is bounded
- Thus, $\dot{x}$ is bounded, and $\dot{e} = \dot{x} - \dot{x}_m$ is bounded, as well
- It immediately follows that $\boxed{\ddot{V} = 4\, a_m\, e(t)\, \dot{e}(t)}$ is bounded
- Using Barbalat's Lemma we conclude that $\dot{V}(t)$ is uniformly continuous function of time

MRAC Design of 1st Order Systems (completed)

- Using Lyapunov-like Lemma: $\lim_{t\to\infty} \dot{V}(x,t) = 0$
- Since $\dot{V} = 2\,a_m\,e(t)^2$ it follows that: $\lim_{t\to\infty} e(t) = 0$
- **Conclusions**
  - achieved asymptotic tracking: $x(t) \to x_m(t)$, as $t \to \infty$
  - all signals in the closed-loop system are bounded

MRAC Design of 1st Order Systems (Block-Diagram)



- Adaptive gains: $\hat{k}_x(t), \hat{k}_r(t)$
- On-line function estimation: $\hat{f}(x) = \hat{\theta}^T(t)\Phi(x) = \sum_{i=1}^{N} \hat{\theta}_i(t)\varphi_i(x)$

Adaptive Dynamic Inversion (ADI) Control

ADI Design of 1st Order Systems

- System Dynamics: $\dot{x} = a\,x + b\,u + f(x)$
  - a, b are constant ***unknown*** parameters
  - ***uncertain*** nonlinear function: $f(x) = \sum_{i=1}^{N} \theta_i\, \varphi_i(x) = \theta^T\, \Phi(x)$
    - vector of constant ***unknown*** parameters: $\theta = \begin{pmatrix}\theta_1 & \ldots & \theta_N\end{pmatrix}^T$
    - vector of known basis functions: $\Phi(x) = \begin{pmatrix}\varphi_1(x) & \ldots & \varphi_N(x)\end{pmatrix}^T$
- Stable Reference Model: $\dot{x}_m = a_m\, x_m + b_m\, r, \quad (a_m < 0)$
- **Control Goal**
  - find u such that: $\lim_{t\to\infty}\left(x(t) - x_m(t)\right) = 0$

ADI Design of 1st Order Systems (continued)

- Rewrite system dynamics:

$$\dot{x} = \hat{a}\,x + \hat{b}\,u + \hat{f}(x) - \underbrace{(\hat{a} - a)}_{\Delta a}x - \underbrace{(\hat{b} - b)}_{\Delta b}u - \underbrace{\left(\hat{f}(x) - f(x)\right)}_{\Delta f(x)}$$

- Function estimation error:

$$\Delta f(x) \triangleq \hat{f}(x) - f(x) = \underbrace{(\hat{\theta} - \theta)}_{\Delta\theta}^{T}\,\Phi(x)$$

- On-line estimated parameters: $\hat{a},\quad \hat{b},\quad \hat{\theta}$
- Parameter estimation errors

$$\Delta a \triangleq \hat{a} - a,\quad \Delta b \triangleq \hat{b} - b,\quad \Delta\theta \triangleq \hat{\theta} - \theta$$

ADI Design of 1st Order Systems (continued)

- ADI Control Feedback: $u = \frac{1}{\hat{b}}\left(\left(a_m - \hat{a}\right)x + b_m\, r\right) - \hat{\theta}^T\, \Phi\left(x\right)$
  - $(N + 2)$ parameters to estimate on-line: $\hat{a}, \hat{b}, \hat{\theta}$
  - Need to protect $\hat{b}$ from crossing zero
- Closed-Loop System: $\dot{x} = a_m\, x + b_m\, r - \Delta a\, x - \Delta b\, u - \Delta\theta\, \Phi\left(x\right)$
- Desired Dynamics: $\dot{x}_m = a_m\, x_m + b_m\, r$
- Tracking error: $e \triangleq x - x_m$
- Tracking error dynamics: $\dot{e} = a_m\, e - \Delta a\, x - \Delta b\, u - \Delta\theta\, \Phi\left(x\right)$
- Lyapunov function candidate

$$V\left(e(t), \Delta a(t), \Delta b(t), \Delta\theta(t)\right) = e^2 + \gamma_a^{-1}\, \Delta a^2 + \gamma_b^{-1}\, \Delta b^2 + \Delta\theta^T\, \Gamma_\theta^{-1}\, \Delta\theta$$

ADI Design of 1st Order Systems (continued)

- Time-derivative of the Lyapunov function

$$\dot{V}(e,\Delta a,\Delta b,\Delta\theta) = 2\,e\,\dot{e} + 2\left(\gamma_a^{-1}\,\Delta a\,\dot{\hat{a}} + \gamma_b^{-1}\,\Delta b\,\dot{\hat{b}} + \Delta\theta^T\,\Gamma_\theta^{-1}\,\dot{\hat{\theta}}\right)$$
$$= 2e\left(a_m\,e - \Delta a\,x - \Delta b\,u - \Delta\theta\,\Phi(x)\right)$$
$$+2\left(\gamma_a^{-1}\,\Delta a\,\dot{\hat{a}} + \gamma_b^{-1}\,\Delta b\,\dot{\hat{b}} + \Delta\theta^T\,\Gamma_\theta^{-1}\,\dot{\hat{\theta}}\right)$$
$$= 2\,a_m\,e^2 + \Delta a\left(\gamma_a^{-1}\,\dot{\hat{a}} - x\,e\right) + \Delta b\left(\gamma_b^{-1}\,\dot{\hat{b}} - u\,e\right) + \Delta\theta^T\left(\Gamma_\theta^{-1}\,\dot{\hat{\theta}} - \Phi(x)\,e\right)$$

- Adaptive laws

$$\dot{\hat{a}} = \gamma_a\,x\,e$$
$$\dot{\hat{b}} = \gamma_b\,u\,e$$
$$\dot{\hat{\theta}} = \Gamma_\theta\,\Phi(x)\,e$$

$$\Rightarrow \quad \dot{V}(e,\Delta a,\Delta b,\Delta\theta) = 2\,a_m\,e^2 \le 0$$

System energy decreases

ADI Design of 1st Order Systems
(stability analysis)

- Similar to MRAC
- Using Barbalat's Lemma and Lyapunov-like Lemma: $\lim_{t\to\infty}\dot{V}(x,t)=\lim_{t\to\infty}\left[2\,a_m\,e(t)^2\right]=0$
- Consequently: $\lim_{t\to\infty}e(t)=0$ ⟹ $x(t)\to x_m(t)$, as $t\to\infty$
- **Conclusions**
  - asymptotic tracking
  - all signals in the closed-loop system are bounded

Parameter Convergence ?

- Convergence of adaptive (on-line estimated) parameters to their true unknown values depends on the reference signal $r(t)$
- If $r(t)$ is very simply, (zero or constant), it is possible to have non-ideal controller parameters that would drive the tracking error to zero
- Need conditions for parameter convergence

Persistency of Excitation (PE)

- Tracking error dynamics is a stable filter

$$\dot{e}(t) = a_m\, e + b\underbrace{\left(\Delta k_x\, x + \Delta k_r\, r + \Delta\theta^T\, \Phi(x)\right)}_{\text{Input}}$$

- Since the filter input signal is uniformly continuous and the tracking error asymptotically converges to zero, then when time *t* is large: $\Delta k_x\, x + \Delta k_r\, r + \Delta\theta^T\, \Phi(x) \cong 0$
- Using vector form:

$$\begin{pmatrix} x & r & \Phi^T(x) \end{pmatrix} \begin{pmatrix} \Delta k_x \\ \Delta k_r \\ \Delta\theta \end{pmatrix} \cong 0$$

Persistency of Excitation (PE) (completed)

- If *r*(*t*) is such that $v = \begin{pmatrix} x & r & \Phi^T(x) \end{pmatrix}^T$ satisfies the so-called "*persistent excitation*" conditions, then the adaptive parameter convergence will take place
  - PE Condition: $\exists \alpha > 0 \quad \forall t \quad \exists T > 0 \quad \int_t^{t+T} v(\tau) v^T(\tau) d\tau > \alpha I_{N+2}$
- PE Condition implies that parameter errors converge to zero
  - for linear systems: *m* - sinusoids ensure convergence of (2 *m*) - parameters
  - not known for nonlinear systems

ADI vs. MRAC

- No knowledge about $\operatorname{sgn} b$
- Adaptive laws are similar
- Both methods yield asymptotic tracking that does not rely on Persistency of Excitation (PE) conditions
- ADI needs protection against $\hat{b}$ crossing zero
  - If PE takes place and initial parameter $\hat{b}(0)$ has wrong sign then a control singularity may occur
- Regressor vector $\Phi(x)$ must have ***bounded*** components, (needed for stability proof)

Example: MRAC of a 1st-Order *Linear* System

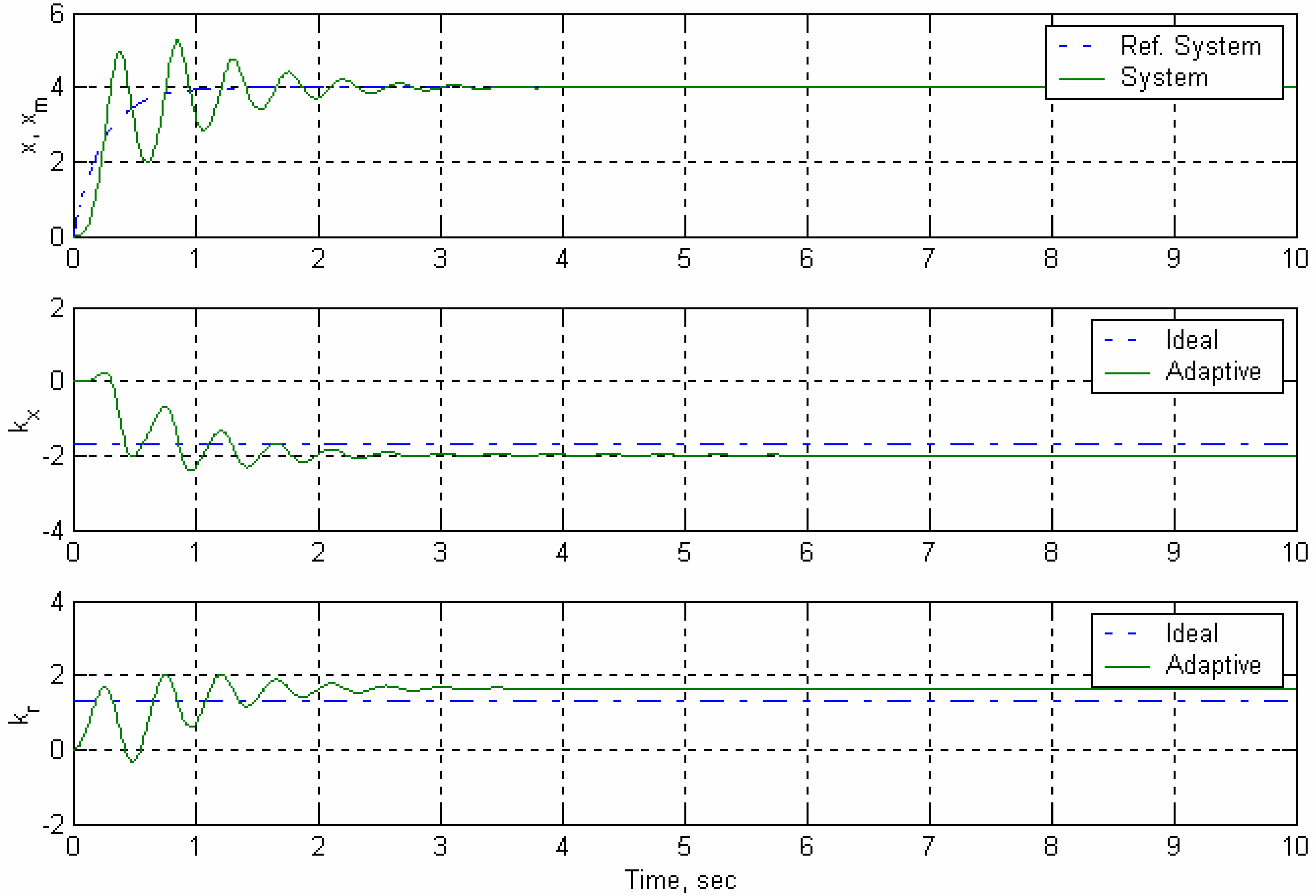
- Unstable Dynamics: $\dot{x} = x + 3u, \quad x(0) = 0$
  - plant parameters $a = 1, \quad b = 3$ are ***unknown*** to the adaptive controller
- Reference Model: $\dot{x}_m = -4\,x_m + 4\,r(t), \quad x_m(0) = 0$
- Adaptive Control: $u = \hat{k}_x\,x + \hat{k}_r\,r$
- Parameter Adaptation:

$$\dot{\hat{k}}_x = -2\,x\,e, \quad \hat{k}_x(0) = 0$$
$$\dot{\hat{k}}_r = -2\,r\,e, \quad \hat{k}_r(0) = 0$$

- Two Reference Inputs:

$$r(t) = 4$$
$$r(t) = 4\sin(3t)$$

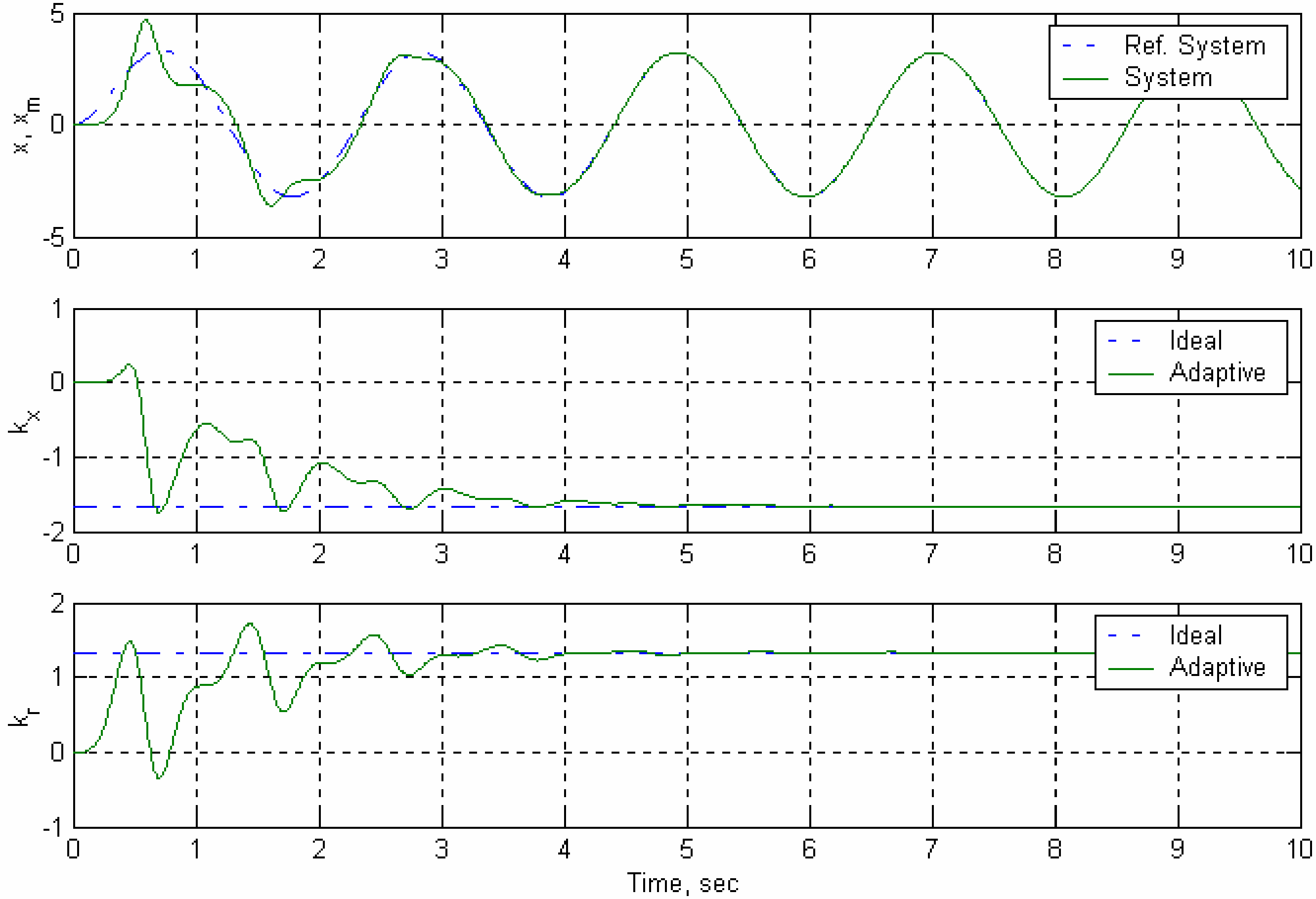
1st-Order *Linear* System MRAC Simulation w/o PE: $r(t) = 4$



Tracking Error Converges to Zero

Parameter Errors don't Converge to Zero

1^{st}-Order *Linear* System
MRAC Simulation with PE: $r(t) = 4\sin(3t)$



Tracking and Parameter Errors Converge to Zero

Example: MRAC of a 1st-Order *Nonlinear* System

- Unstable Dynamics: $\dot{x} = x + 3\left(u - f(x)\right), \quad x(0) = 0$
  - plant parameters $a = 1, \quad b = 3$ are *unknown*
  - nonlinearity: $f(x) = \theta^T \Phi(x)$
    - *known* basis functions: $\Phi(x) = \begin{pmatrix} x^3 & e^{-(x+0.5)^2 10} & e^{-(x-0.5)^2 10} & \sin(2x) \end{pmatrix}^T$
    - *unknown* parameters: $\theta = \begin{pmatrix} 0.01 & -1 & 1 & 0.5 \end{pmatrix}^T$
- Reference Model: $\dot{x}_m = -4\,x_m + 4\,r(t), \quad x_m(0) = 0$
- Adaptive Control: $u = \hat{k}_x\, x + \hat{k}_r\, r + \hat{\theta}^T \Phi(x)$
- Parameter Adaptation:

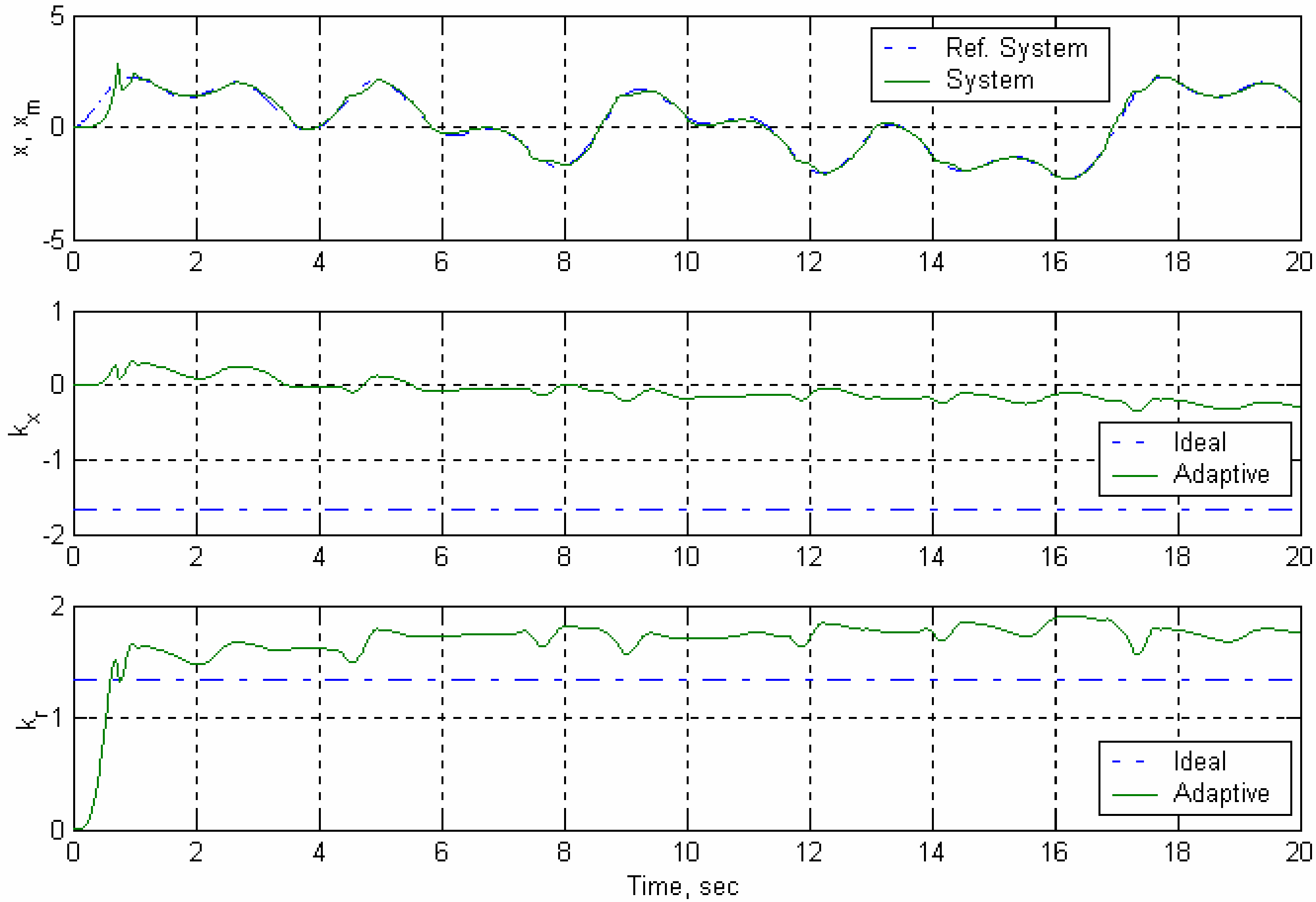
$$\dot{\hat{k}}_x = -2\,x\,e, \quad \hat{k}_x(0) = 0$$

$$\dot{\hat{k}}_r = -2\,r\,e, \quad \hat{k}_r(0) = 0$$

$$\dot{\hat{\theta}} = -2\,\Phi(x)\,e, \quad \hat{\theta}(0) = 0_4$$

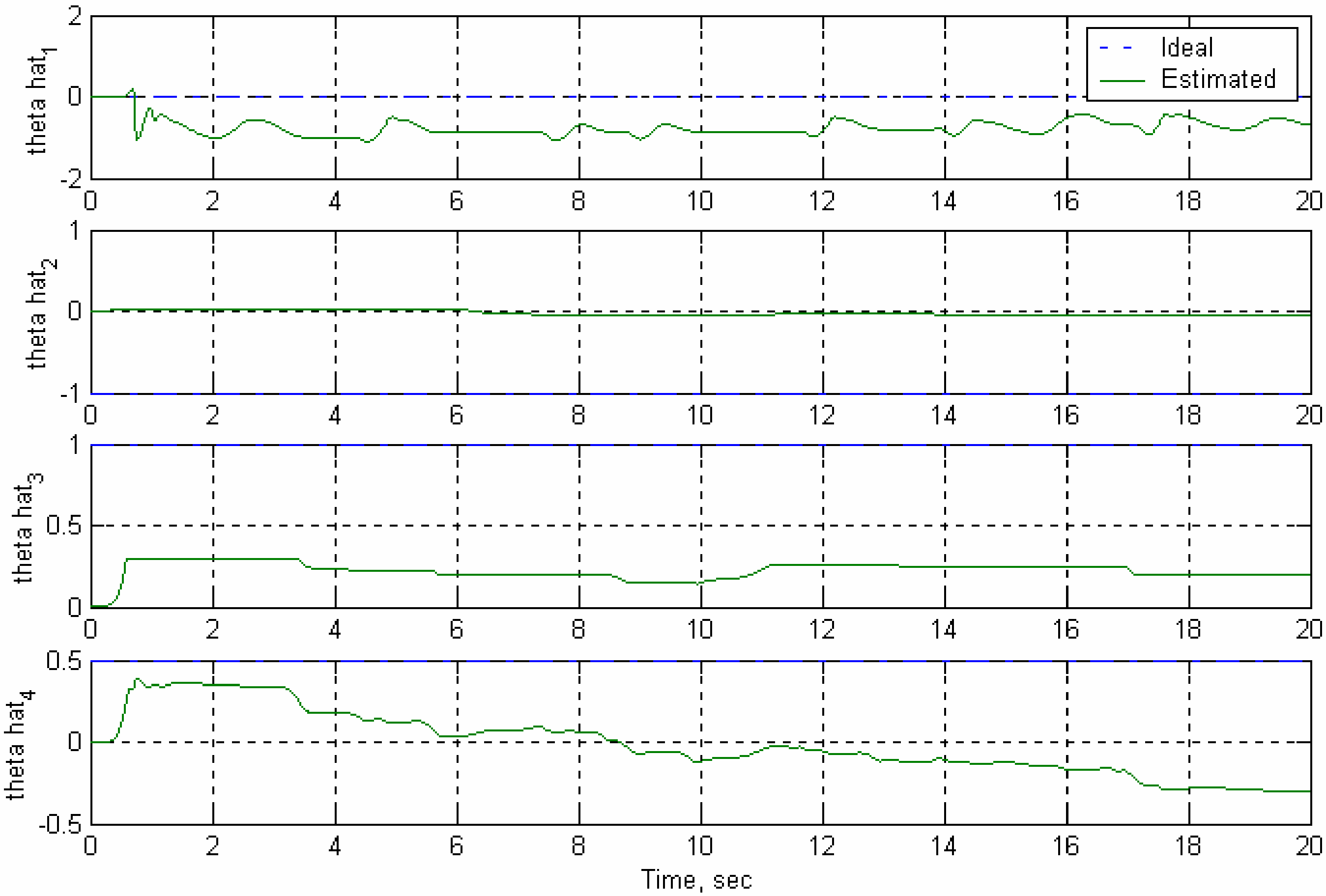
- Reference Input: $r(t) = \sin(3t) + \sin\left(\frac{3t}{2}\right) + \sin\left(\frac{3t}{4}\right) + \sin\left(\frac{3t}{8}\right)$

1st-Order *Nonlinear* System
MRAC Simulation



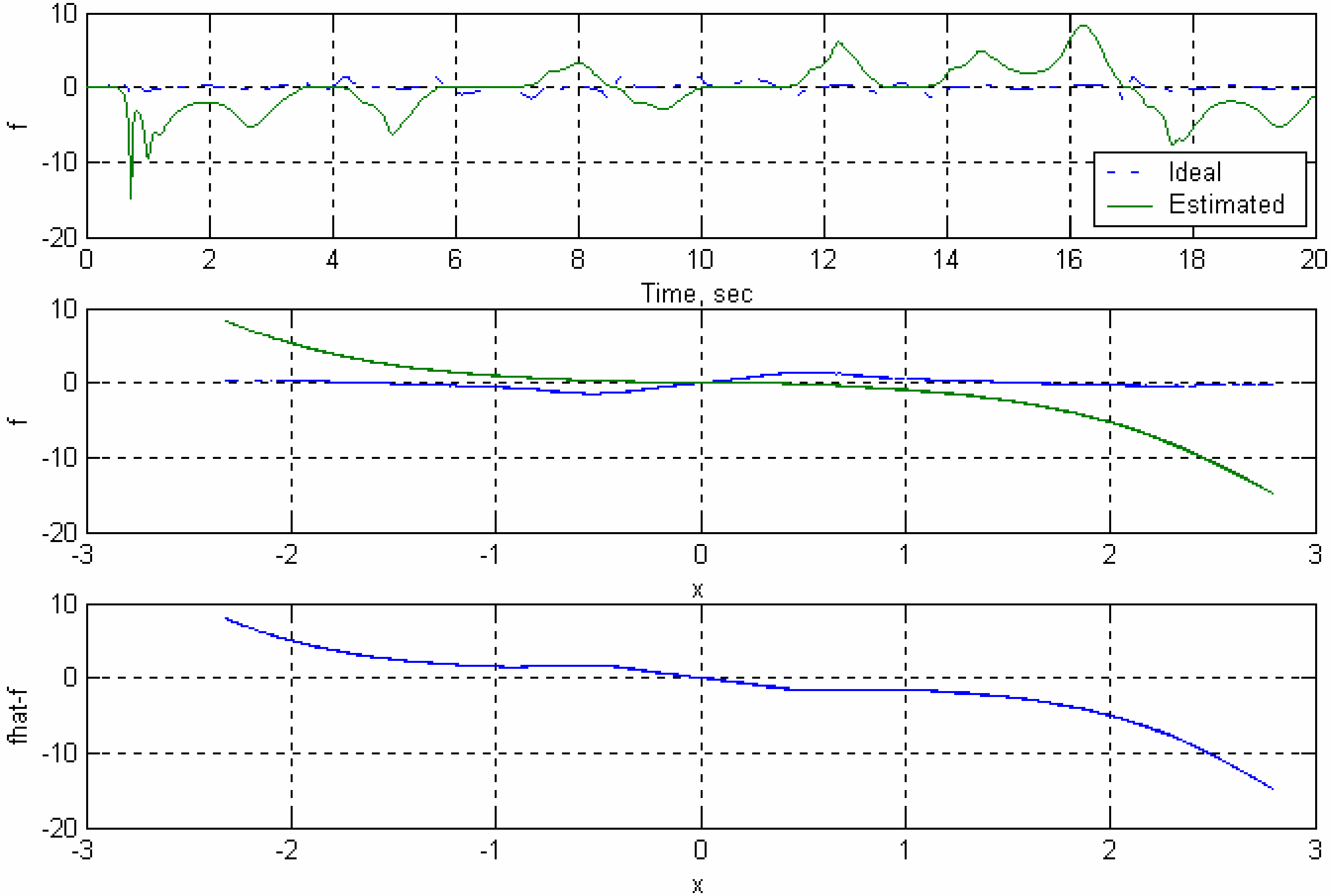
Good Tracking & Poor Parameter Estimation

1st-Order *Nonlinear* System
MRAC Simulation, (continued)



Nonlinearity: Poor Parameter Estimation

1st-Order *Nonlinear* System MRAC Simulation, (completed)



Nonlinearity: Poor Estimation

Example: MRAC of a 1st-Order *Nonlinear* System with *Local* Nonlinearity

- Unstable Dynamics: $\dot{x} = x + 3\left(u - f(x)\right), \quad x(0) = 0$
  - plant parameters $a = 1, \quad b = 3$ are *unknown*
  - nonlinearity: $f(x) = \theta^T \Phi(x)$
    - *known* basis functions: $\Phi(x) = \left(x^3 \quad e^{-(x+0.5)^2 10} \quad e^{-(x-0.5)^2 10} \quad \sin(2x)\right)^T$
    - *unknown* parameters: $\theta = \left(0 \quad -1 \quad 1 \quad 0\right)^T$
- Reference Model: $\dot{x}_m = -4x_m + 4r(t), \quad x_m(0) = 0$
- Adaptive Control: $u = \hat{k}_x x + \hat{k}_r r + \hat{\theta}^T \Phi(x)$
- Parameter Adaptation:

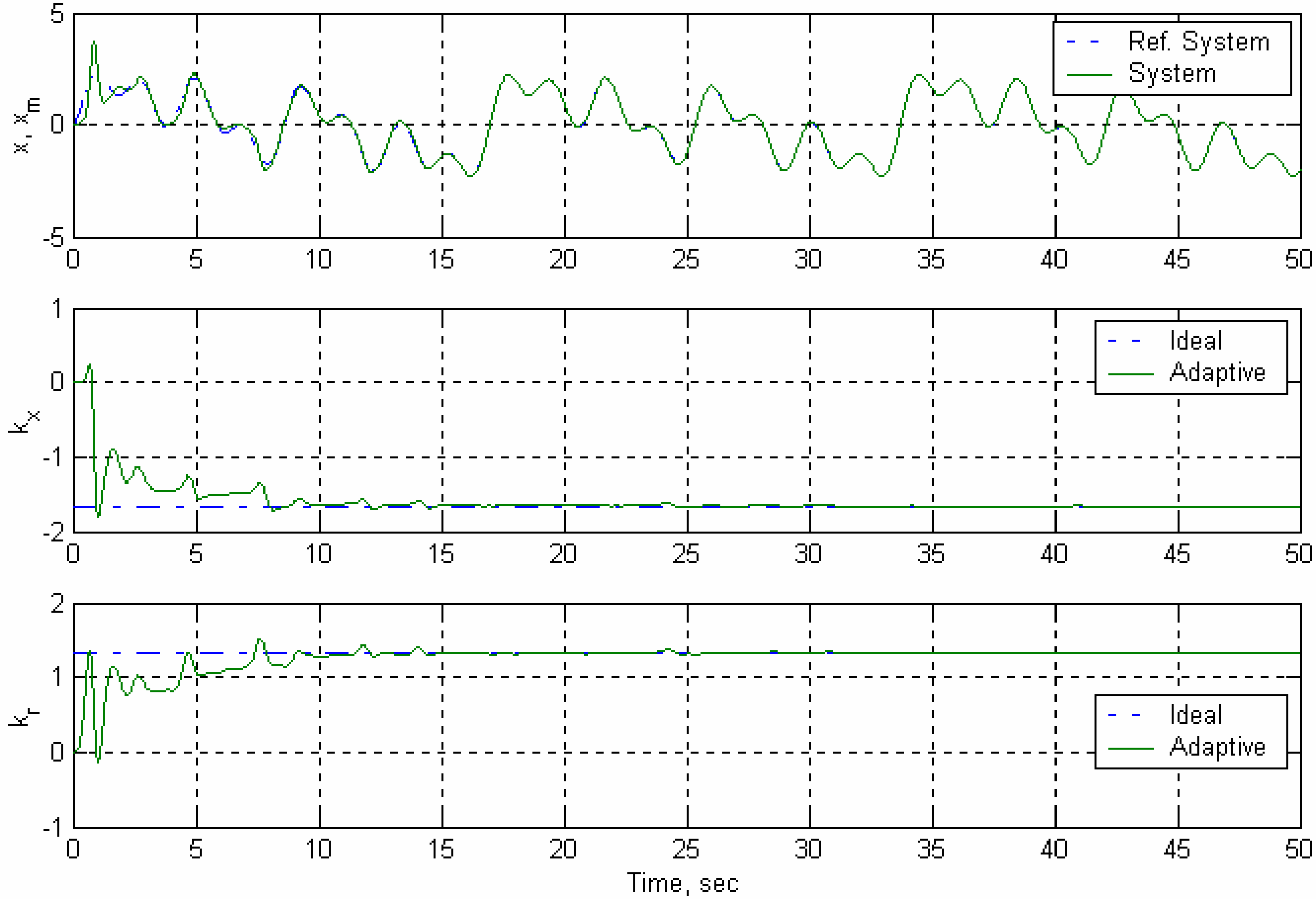
$$\dot{\hat{k}}_x = -2xe, \quad \hat{k}_x(0) = 0$$

$$\dot{\hat{k}}_r = -2re, \quad \hat{k}_r(0) = 0$$

$$\dot{\hat{\theta}} = -2\Phi(x)e, \quad \hat{\theta}(0) = 0_4$$

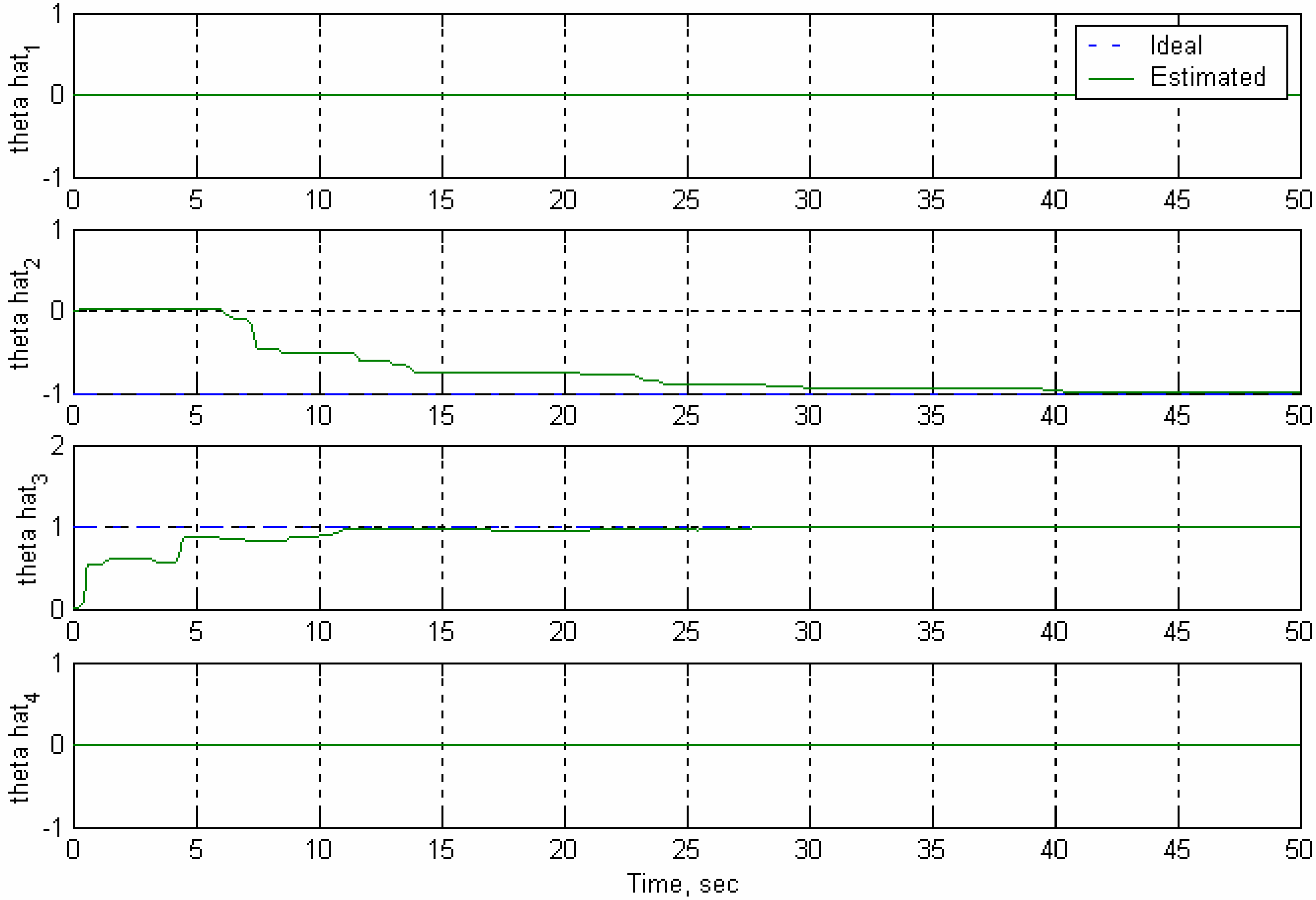
- Reference Input: $r(t) = \sin(3t) + \sin\left(\frac{3t}{2}\right) + \sin\left(\frac{3t}{4}\right) + \sin\left(\frac{3t}{8}\right)$

1st-Order *Nonlinear* System with *Local* Nonlinearity: MRAC Simulation



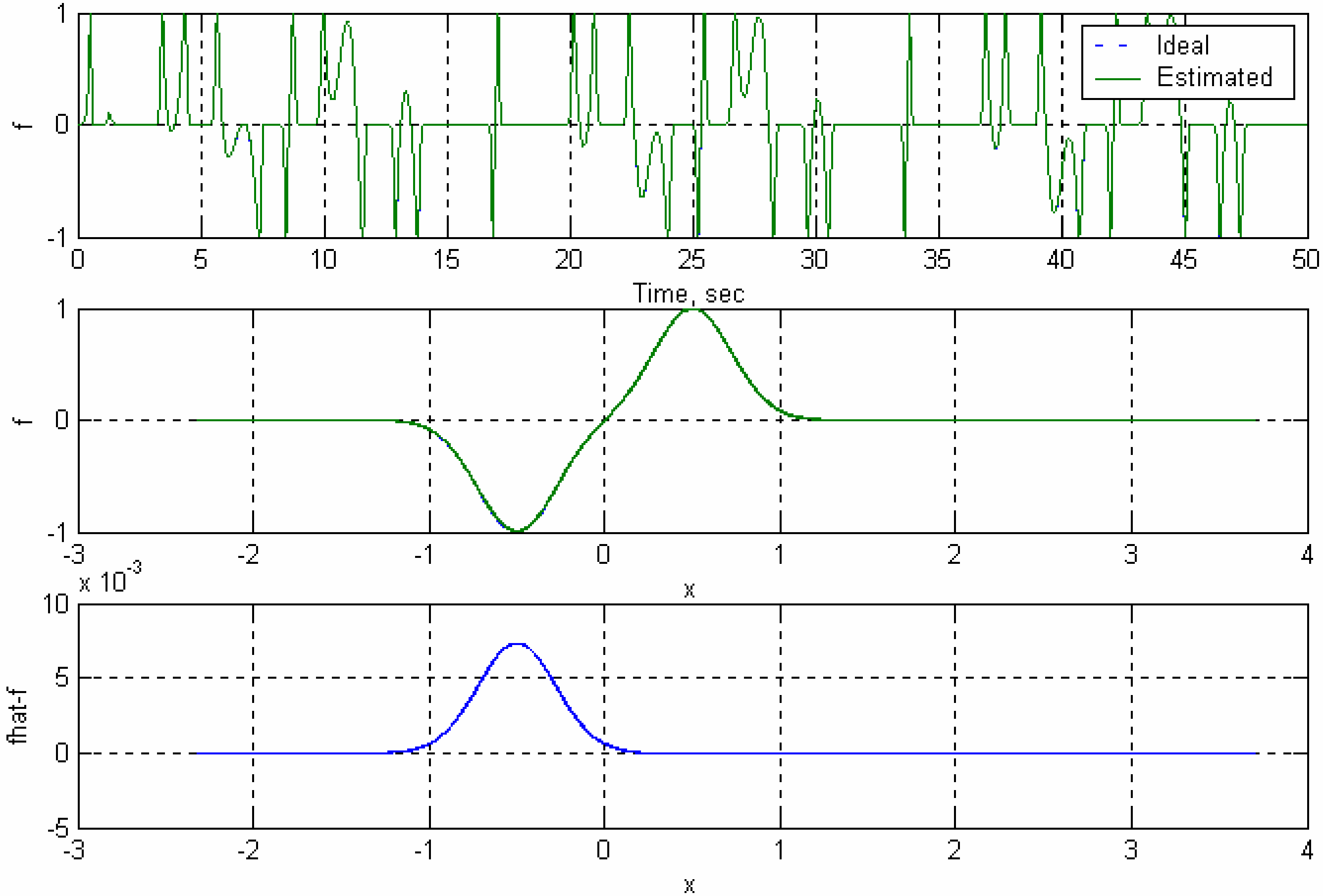
Good Tracking & Parameter Estimation

1^{st}-Order *Nonlinear* System with *Local* Nonlinearity: MRAC Simulation, (continued)



Nonlinearity: Good Parameter Estimation

1^{st}-Order *Nonlinear* System with *Local* Nonlinearity: MRAC Simulation, (completed)



Nonlinearity: Good Function Approximation

MRAC of a 1st-Order *Nonlinear* System
Conclusions & Observations

- *Direct* MRAC provides good tracking in spite of unknown parameters and nonlinear uncertainties in the system dynamics
- Parameter convergence IS NOT guaranteed
- Sufficient Condition for Parameter Convergence
  - Reference input $r(t)$ satisfies Persistency of Excitation
    - PE is hard to verify / compute
  - Enforced for linear systems with *local* nonlinearities
- A control strategy that depends on parameter convergence, (such as *indirect* MRAC), is unreliable, unless PE condition takes place

MRAC Design of n^{th} Order Systems

- System Dynamics: $\dot{x} = A x + B \Lambda \left(u - f(x) \right), \quad x \in R^n, \quad u \in R^m$
  - $A \in R^{n \times n}, \Lambda = \operatorname{diag}\left(\lambda_1 \quad \ldots \quad \lambda_m \right) \in R^{m \times m}$ are constant ***unknown*** matrices
  - $B \in R^{n \times m}$ is ***known*** constant matrix
  - $\forall\, i = 1, \ldots, m \quad \operatorname{sgn}\left(\lambda_i \right)$ is ***known***
  - ***uncertain matched*** nonlinear function: $f(x) = \Theta^T \Phi(x) \in R^m$
    - ***matrix*** of constant ***unknown*** parameters: $\Theta \in R^{m \times N}$
    - vector of *N* ***known*** basis functions: $\Phi(x) = \left(\varphi_1(x) \quad \ldots \quad \varphi_N(x) \right)^T$
- Stable Reference Model: $\dot{x}_m = A_m x_m + B_m r, \quad \left(A_m \text{ is Hurwitz} \right)$

  $r \in R^m, \quad A_m \in R^{n \times n}, \quad B_m \in R^{n \times m}$
- **Control Goal**
  - find *u* such that: $\lim\limits_{t \to \infty} \left\| x(t) - x_m(t) \right\| = 0$

MRAC Design of n^{th} Order Systems (continued)

- Control Feedback: $u = \hat{K}_x^T x + \hat{K}_r^T r + \hat{\Theta}^T \Phi(x)$
  - ($m\, n + m^2 + m\, N$) - parameters to estimate: $\hat{K}_x, \quad \hat{K}_r, \quad \hat{\Theta}$
- Closed-Loop System: $\dot{x} = \left(A + B\Lambda\hat{K}_x^T\right)x + B\Lambda\left(\hat{K}_r^T r + \left(\hat{\Theta} - \Theta\right)^T \Phi(x)\right)$
- Desired Dynamics: $\dot{x}_m = A_m\, x_m + B_m\, r$
- Model Matching Conditions
  - there exist *ideal* gains (K_x, K_r) such that: $A + B\Lambda K_x^T = A_m$, $B\Lambda K_r^T = B_m$
  - *Note*: knowledge of the ideal gains is not required

$$A + B\Lambda\hat{K}_x^T - A_m = A + B\Lambda\hat{K}_x^T - A - B\Lambda K_x^T = B\Lambda\left(\hat{K}_x - K_x\right)^T = B\Lambda\Delta K_x^T$$

$$B\Lambda\hat{K}_r^T - B_m = B\Lambda\hat{K}_r^T - B\Lambda K_r^T = B\Lambda\left(\hat{K}_r - K_r\right)^T = B\Lambda\Delta\hat{K}_r^T$$

MRAC Design of n^{th} Order Systems (continued)

- Tracking Error: $e(t) = x(t) - x_m(t)$
- Error Dynamics:

$$\dot{e}(t) = \dot{x}(t) - \dot{x}_m(t) =$$

$$\left(A + B\Lambda\hat{K}_x^T\right)x + B\Lambda\left(\hat{K}_r^T r + \underbrace{\left(\hat{\Theta} - \Theta\right)}_{\Delta\Theta}{}^T \Phi(x)\right) - A_m x_m - B_m r \pm A_m x$$

$$= A_m\left(x - x_m\right) + \left(A + B\Lambda\hat{K}_x^T - A_m\right)x + B\Lambda\left(\hat{K}_r - K_r\right)^T r + B\Lambda\Delta\Theta^T\Phi(x)$$

$$= A_m e + B\Lambda\left(\Delta K_x^T x + \Delta K_r^T r + \Delta\Theta^T\Phi(x)\right)$$

MRAC Design of n^{th} Order Systems (continued)

- Lyapunov Function Candidate

$$V(e, \Delta K_x, \Delta K_r, \Delta\Theta) = e^T P e + \operatorname{trace}\left(\Delta K_x^T \Gamma_x^{-1} \Delta K_x |\Lambda|\right) + \operatorname{trace}\left(\Delta K_r^T \Gamma_r^{-1} \Delta K_r |\Lambda|\right) + \operatorname{trace}\left(\Delta\Theta^T \Gamma_\Theta^{-1} \Delta\Theta |\Lambda|\right)$$

  - where: $\operatorname{trace}(S) \triangleq \sum s_{ii}$
  - $|\Lambda| \triangleq \operatorname{diag}\left(|\lambda_1| \quad \ldots \quad |\lambda_m^i|\right)$ is diagonal matrix with positive elements
  - $\Gamma_x = \Gamma_x^T > 0, \quad \Gamma_r = \Gamma_r^T > 0, \quad \Gamma_\Theta = \Gamma_\Theta^T > 0$ are symmetric positive definite matrices
  - $P = P^T > 0$ is unique symmetric positive definite solution of the algebraic Lyapunov equation $P A_m + A_m^T P = -Q$
    - $Q = Q^T > 0$ is any symmetric positive definite matrix

MRAC Design of n^{th} Order Systems (continued)

- Adaptive Control Design
  - Choose adaptive laws, (on-line parameter updates) such that the time-derivative of the Lyapunov function decreases along the error dynamics trajectories

$$\dot{\hat{K}}_x = -\Gamma_x \, x \, e^T P B \operatorname{sgn}(\Lambda)$$
$$\dot{\hat{K}}_r = -\Gamma_r \, r \, e^T P B \operatorname{sgn}(\Lambda)$$
$$\dot{\hat{\Theta}} = -\Gamma_\Theta \, \Phi(x) \, e^T P B \operatorname{sgn}(\Lambda)$$

- Time-derivative of the Lyapunov function becomes semi-negative definite!

$$\dot{V}\left(e(t), \Delta K_x(t), \Delta K_r(t), \Delta\Theta(t)\right) = -e^T(t) \, Q \, e(t) \le 0$$

MRAC Design of n^{th} Order Systems (completed)

- Using Barbalat's and Lyapunov-like Lemmas: $\lim_{t\to\infty} \dot{V}(x,t) = 0$
- Since $\dot{V} = -e^T(t)\,Q\,e^T(t)$ it follows that: $\lim_{t\to\infty} \|e(t)\| = 0$
- **Conclusions**
  - achieved asymptotic tracking: $x(t) \to x_m(t), \text{ as } t \to \infty$
  - all signals in the closed-loop system are bounded
- <u>*Remark*</u>
  - Parameter convergence IS NOT guaranteed

Robustness of Adaptive Control

- Adaptive controllers are designed to control real physical systems
  - non-parametric uncertainties may lead to performance degradation and / or instability
    - low-frequency unmodeled dynamics, (structural vibrations)
    - low-frequency unmodeled dynamics, (Coulomb friction)
    - measurement noise
    - computation round-off errors and sampling delays
- Need to enforce robustness of MRAC

Parameter Drift in MRAC

- When $r(t)$ is ***persistently exciting*** the system, both simulation and analysis indicate that MRAC systems are robust w.r.t non-parametric uncertainties
- When $r(t)$ IS NOT ***persistently exciting*** even small uncertainties may lead to severe problems
  - estimated parameters drift slowly as time goes on, and suddenly diverge sharply
  - reference input contains insufficient parameter information
  - adaptation has difficulty distinguishing parameter information from noise

Parameter Drift in MRAC: Summary

- Occurs when signals are not persistently exciting
- Mainly caused by measurement noise and disturbances
- Does not effect tracking accuracy until the instability occurs
- Leads to *sudden* failure

Dead-Zone Modification

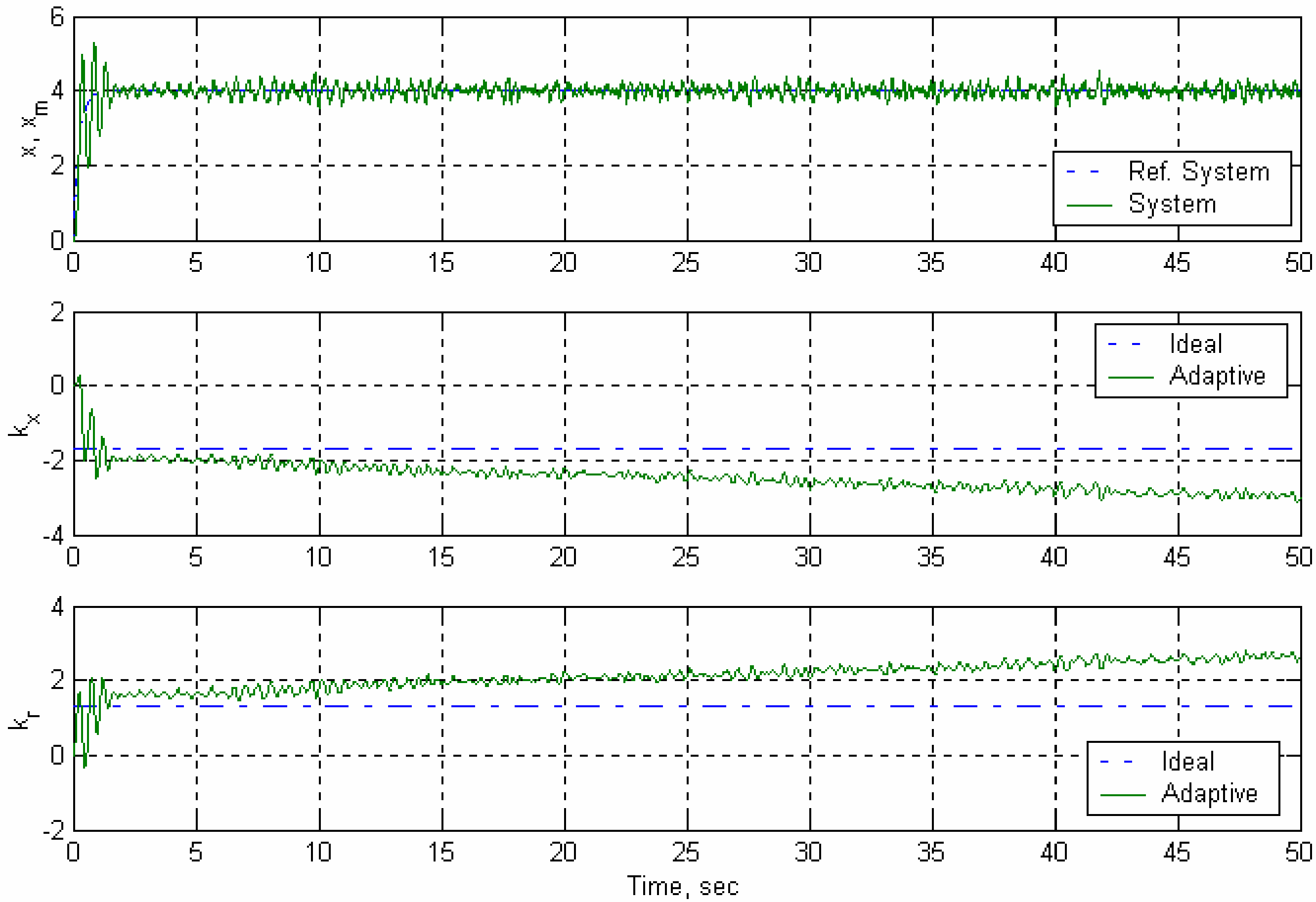
- Method is based on the observation that small tracking errors contain mostly noise and disturbance
- Solution
  - Turn off the adaptation process for "small" tracking errors
  - MRAC using Dead-Zone ⇨
  - ε is the size of the dead-zone
- Outcome
  - *Bounded* Tracking

$$\dot{\hat{K}}_x = \begin{cases} -\Gamma_x \, x \, e^T P B \operatorname{sgn}(\Lambda), & \|e\| > \varepsilon \\ 0, & \|e\| \le \varepsilon \end{cases}$$

$$\dot{\hat{K}}_r = \begin{cases} -\Gamma_r \, r \, e^T P B \operatorname{sgn}(\Lambda), & \|e\| > \varepsilon \\ 0, & \|e\| \le \varepsilon \end{cases}$$

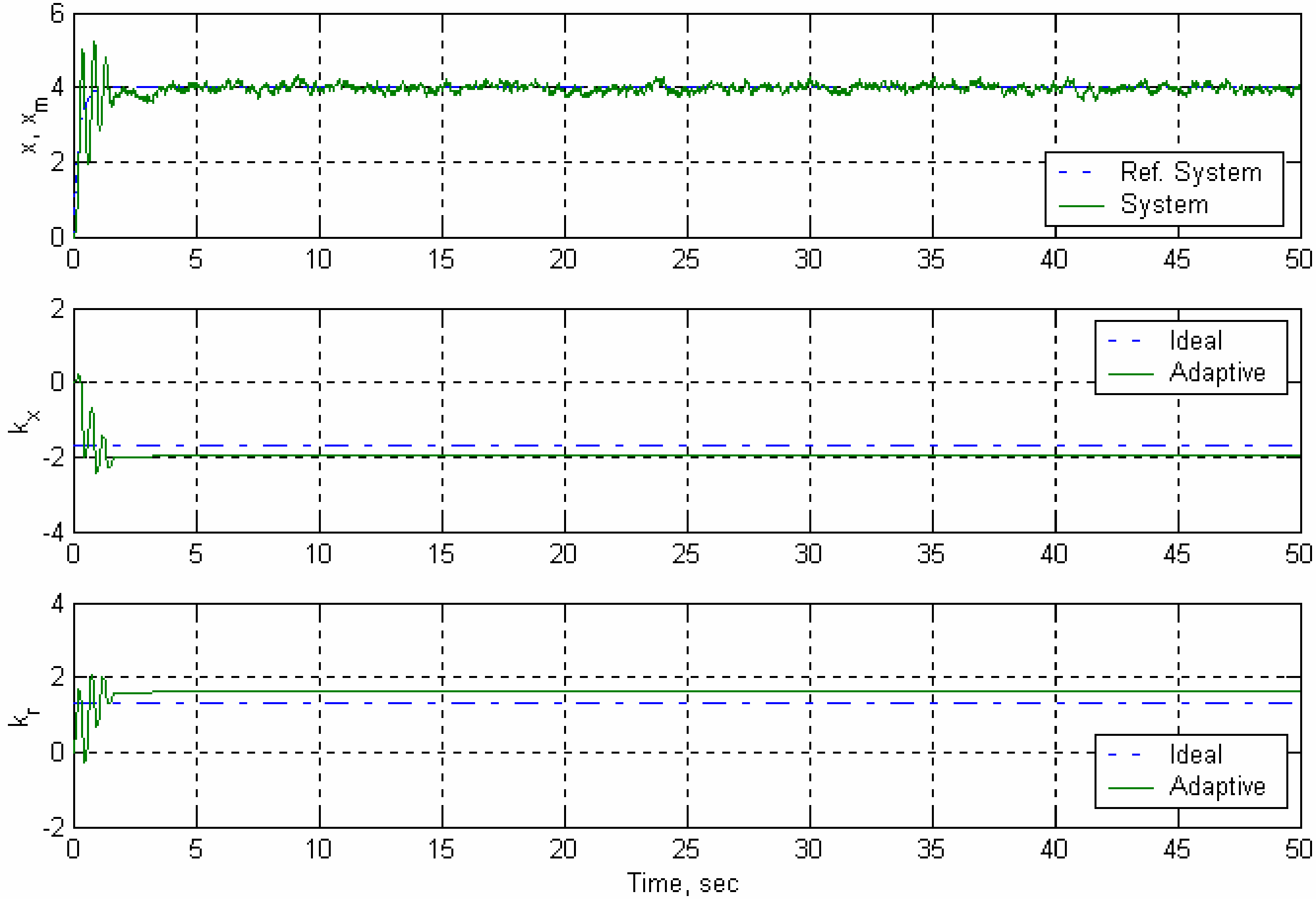
$$\dot{\hat{\Theta}} = \begin{cases} -\Gamma_\Theta \, \Phi(x) \, e^T P B \operatorname{sgn}(\Lambda), & \|e\| > \varepsilon \\ 0, & \|e\| \le \varepsilon \end{cases}$$

1st-Order *Linear* System with Noise MRAC w/o Dead-Zone: $r(t) = 4$



- Satisfactory Tracking
- **Parameter Drift** due to measurement noise

1st-Order *Linear* System with Noise MRAC *with* Dead-Zone: $r(t) = 4$



- Satisfactory Tracking
- **No Parameter Drift**

Parametric and Non-Parametric Uncertainties

- Parametric Uncertainties are often easy to characterize
  - Example: $m\ddot{x} = u$
    - uncertainty in mass *m* is parametric
    - neglected motor dynamics, measurement noise, sensor dynamics are non-parametric uncertainties
- Both Parametric and Non-Parametric Uncertainties occur during Function Approximation

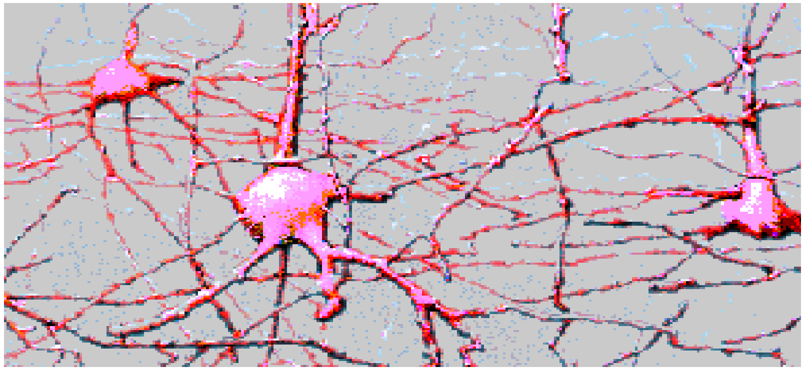
$$\hat{f}(x) = \sum_{i=1}^{N} \theta_i \varphi_i(x) + \varepsilon(x)$$

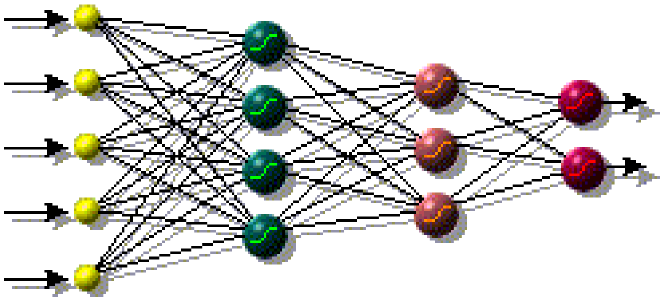
parametric

non-parametric

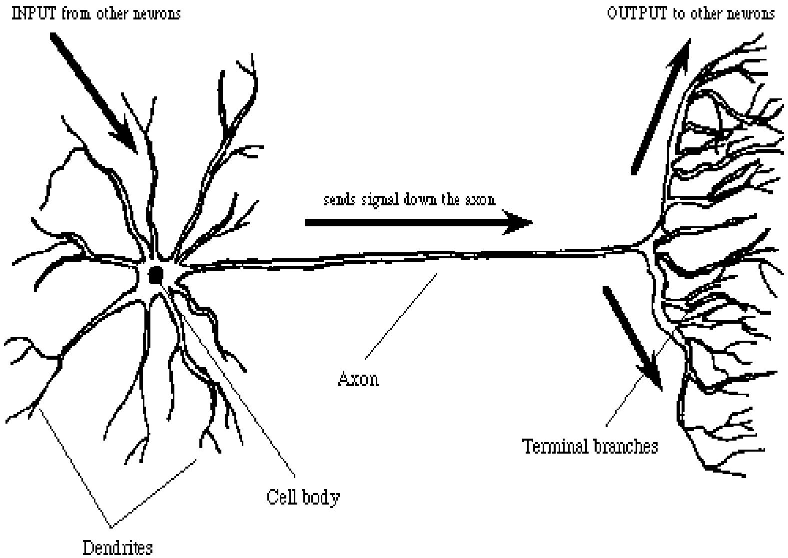
Enforcing Robustness in MRAC Systems

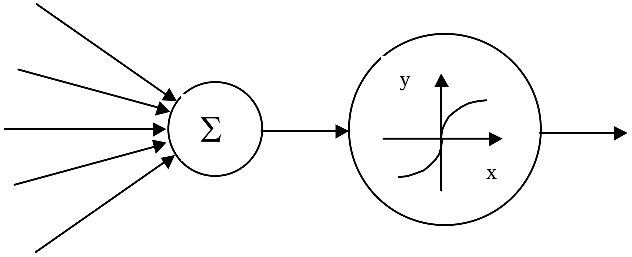
- Non-Parametric Uncertainty
  - Dead-Zone modification
  - Others ?
- Parametric Uncertainty
  - Need a set of basis functions that can approximate a large class of functions within a given tolerance
    - Fourier series
    - Splines
    - Polynomials
    - Artificial Neural Networks
      - sigmoidal
      - RBF





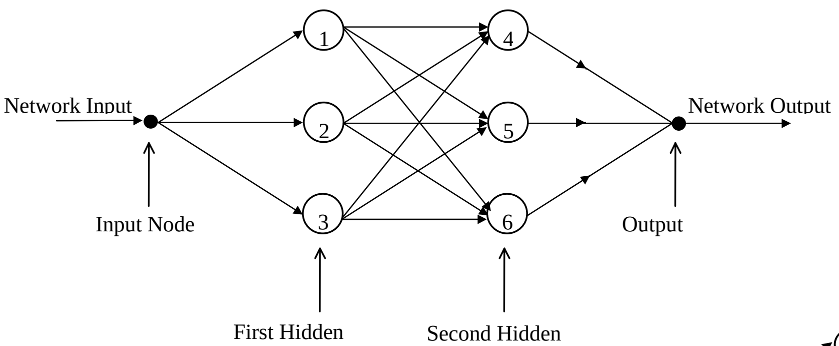
Artificial Neural Networks



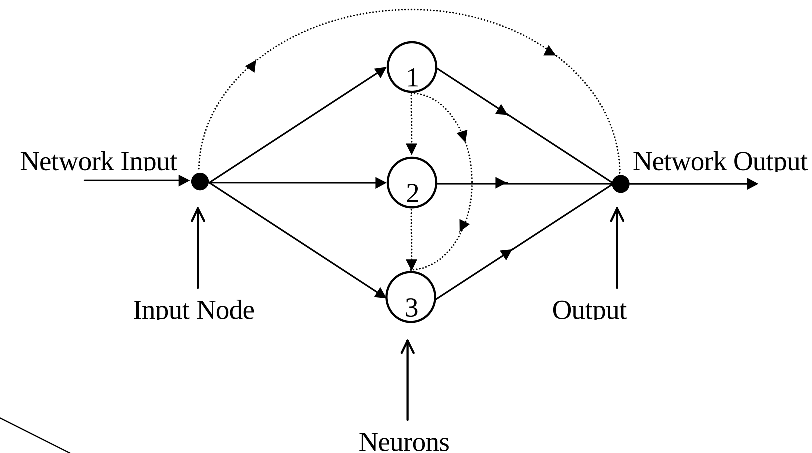


NN Architectures

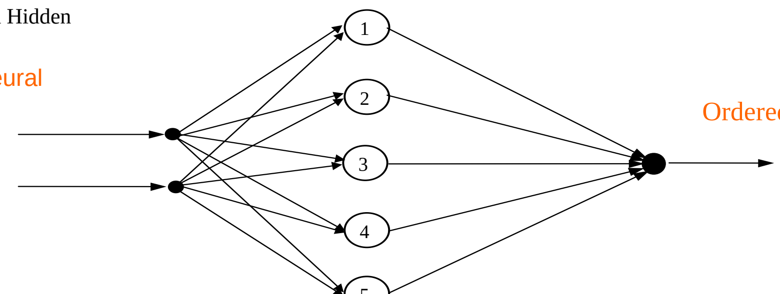
- ***Artificial Neural Networks*** are multi-input-multi-output systems composed of many inter-connected nonlinear processing elements (neurons) operating in parallel



Two-Hidden-Layers Neural Network



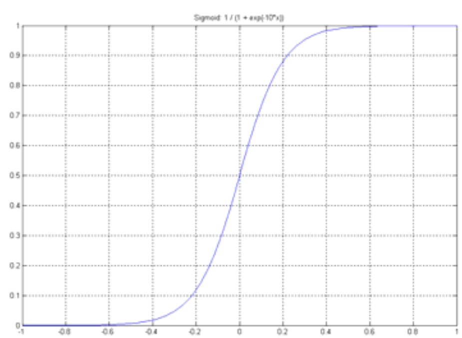
Ordered Neural Network

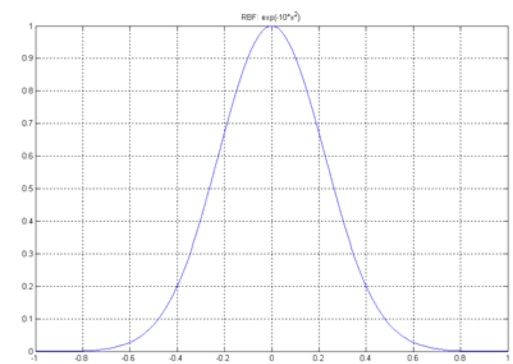


Single-Hidden-Layer Neural Network

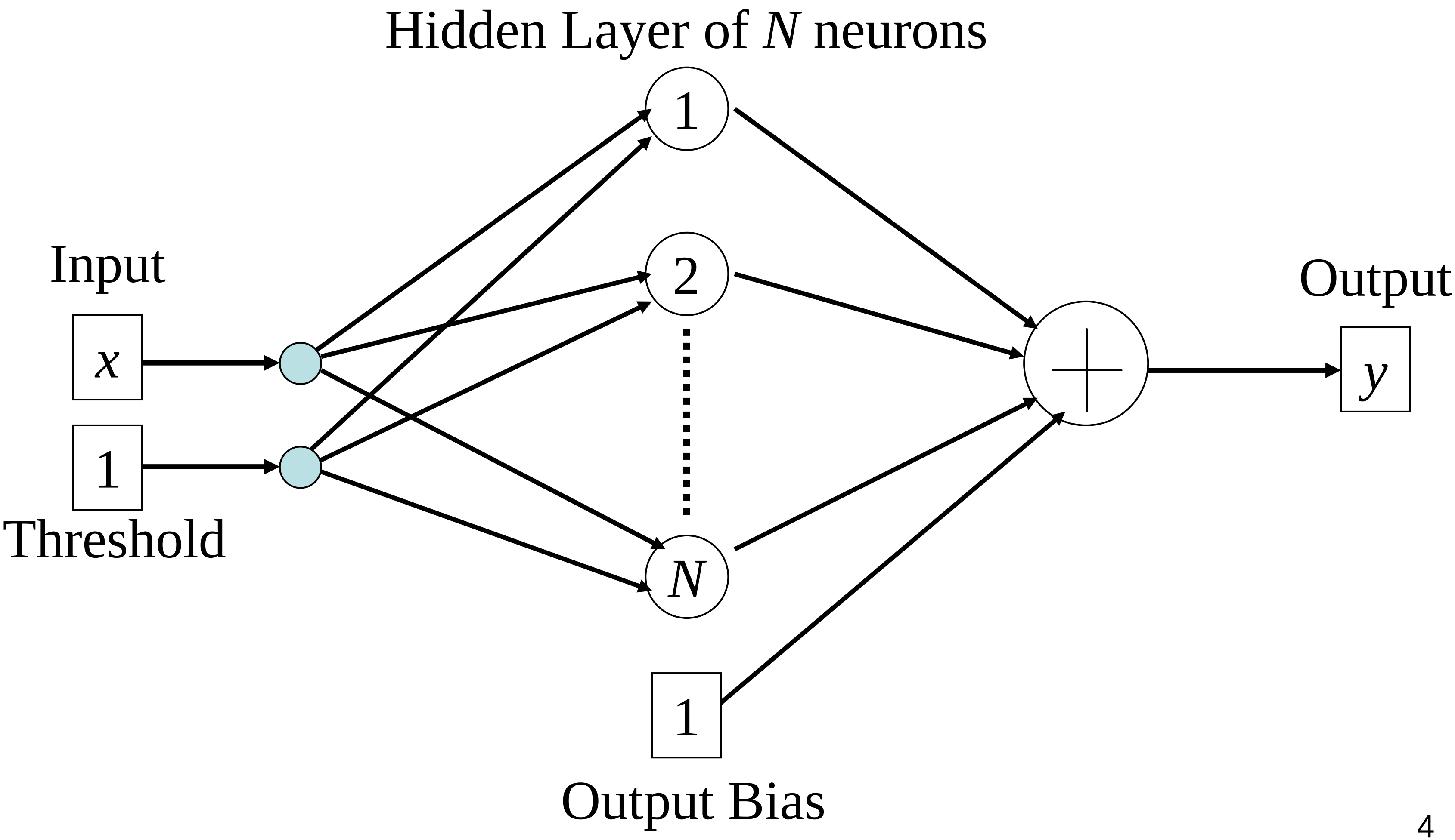
Single Hidden Layer (SHL) Feedforward Neural Networks (FNN)

- Three distinct characteristics
  - model of each neuron includes a ***nonlinear*** activation function
    - sigmoid $\sigma(s) = \dfrac{1}{1+e^{-s}}$
    - radial basis function $\varphi(x) = e^{-\frac{\|x-r\|^2}{2\sigma^2}}$
  - a single layer of N ***hidden*** neurons
  - ***feedforward*** connectivity





SHL FNN Architecture



SHL FNN Function

- Maps *n* - dimensional input into *m* - dimensional output: $x \to NN(x), \quad x \in R^n, \quad NN(x) \in R^m$
- Functional Dependence
  - sigmoidal: $NN(x) = W^T \vec{\sigma}\left(V^T x + \theta\right) + b$
  - RBF:

$$NN(x) = W^T \underbrace{\begin{pmatrix} \varphi\left(\left\|x - C_1\right\|\right) \\ \vdots \\ \varphi\left(\left\|x - C_N\right\|\right) \end{pmatrix}}_{\Phi(x)} + b = W^T \Phi(x) + b$$

Sigmoidal NN

- Matrix form: $NN(x) = W^T \vec{\sigma}\left(V^T \begin{pmatrix} x \\ 1 \end{pmatrix}\right) + c$

- Vector of hidden layer sigmoids:

$$\vec{\sigma}\left(V^T x + \theta\right) = \left(\sigma\left(v_1^T x + \theta_1\right) \quad \ldots \quad \sigma\left(\vec{v}_N^T x + \theta_N\right)\right)^T$$

- Matrix of inner-layer weights: $V = \left(\vec{v}_1 \quad \ldots \quad \vec{v}_N\right) \in R^{n\times N}$
- Matrix of output-layer weights: $W = \left(\vec{w}_1 \quad \ldots \quad \vec{w}_m\right) \in R^{N\times m}$
- Vector of output biases $c \in R^m$ and thresholds $\theta \in R^N$
- k^{th} output:

$$NN_k(x) = \vec{w}_k^T \sigma\left(\vec{v}_k^T x + \theta_k\right) + c_k = \sum_{j=1}^{N} w_{jk} \; \sigma\left(\sum_{i=1}^{n} v_{ik} \, x_i + \theta_k\right) + c_k$$

Sigmoidal NN, (continued)

- Universal Approximation Property
  - large class of functions can be approximated by sigmoidal SHL NN-s within any given tolerance, on compacted domains

$$\forall f(x): R^n \to R^m \quad \forall \varepsilon > 0 \quad \exists N, W, b, V, \theta \quad \forall x \in X \subset R^n$$

$$\left\| f(x) - W^T \vec{\sigma}\left(V^T \begin{pmatrix} x \\ 1 \end{pmatrix}\right) - b \right\| \le \varepsilon = O\left(\frac{1}{\sqrt{N}}\right)$$

- Introduce: $W \triangleq \begin{bmatrix} W^T & b \end{bmatrix}^T, \quad V \triangleq \begin{bmatrix} V^T & \theta \end{bmatrix}^T, \quad \vec{\sigma} \triangleq \begin{bmatrix} \vec{\sigma} \\ 1 \end{bmatrix}, \quad \mu \triangleq \begin{bmatrix} x \\ 1 \end{bmatrix}$

- Then: ⇨ $NN(x) = W^T \vec{\sigma}\left(V^T \mu\right)$

Sigmoidal SHL NN: Summary

- A very large class of functions can be approximated using *linear combinations of shifted and scaled sigmoids*
- NN approximation error decreases as the number of hidden-layer neurons *N* increases:

$$\left\| f\left(x\right) - NN\left(x\right) \right\| = \mathrm{O}\left(N^{-\frac{1}{2}} \right)$$

- Inclusion of biases and thresholds into NN weight matrices simplifies bookkeeping

$$NN\left(x\right) = W^T \vec{\sigma}\left(V^T \mu\right)$$

- Function approximation using sigmoidal NN means finding connection weights *W* and *V*

RBF NN

- Matrix form: $NN(x) = W^T \Phi(x) + b$
- Vector of RBF-s: $\Phi(x) = \begin{pmatrix} e^{\frac{-\|x - C_1\|^2}{2\sigma_1^2}} & \dots & e^{\frac{-\|x - C_N\|^2}{2\sigma_N^2}} \end{pmatrix}^T$
- Matrix of RBF centers: $C \triangleq \begin{bmatrix} \vec{C}_1 & \dots & \vec{C}_N \end{bmatrix} \in R^{n \times N}$
- Vector of RBF widths: $\vec{\sigma} \triangleq (\sigma_1 \quad \dots \quad \sigma_N)^T \in R^N$
- Matrix of output weights: $W = (\vec{w}_1 \quad \dots \quad \vec{w}_m) \in R^{N \times m}$
- Vector of output biases: $b \in R^m$
- k^{th} output: $NN_k(x) = \vec{w}_k^T \Phi(x) + b_k = \sum_{j=1}^{N} w_{jk} \, e^{\frac{-\|x - C_j\|^2}{2\sigma_j^2}} + b_k$

RBF NN, (continued)

- Universal Approximation Property
  - large class of functions can be approximated by RBF NN-s within any given tolerance, on compacted domains

$$\forall f(x): R^n \to R^m \quad \forall \varepsilon > 0 \quad \exists N, W, \vec{C}, \vec{\sigma} \quad \forall x \in X \subset R^n$$

$$\left\| f(x) - W^T \Phi(x) - b \right\| \le \varepsilon = \mathrm{O}\left(N^{-\frac{1}{n}} \right)$$

- Introduce: $W \triangleq [W \quad b], \quad \Phi(x) \triangleq \begin{bmatrix} \Phi(x) \\ 1 \end{bmatrix}$
- Then: → $NN(x) = W^T \Phi(x)$

RBF NN: Summary

- A very large class of functions can be approximated using *linear combinations of shifted and scaled gaussians*
- NN approximation error decreases as the number of hidden-layer neurons *N* increases:

$$\left\| f(x) - NN(x) \right\| = O\left(N^{-\frac{1}{n}} \right)$$

- Inclusion of biases into NN output weight matrix simplifies bookkeeping

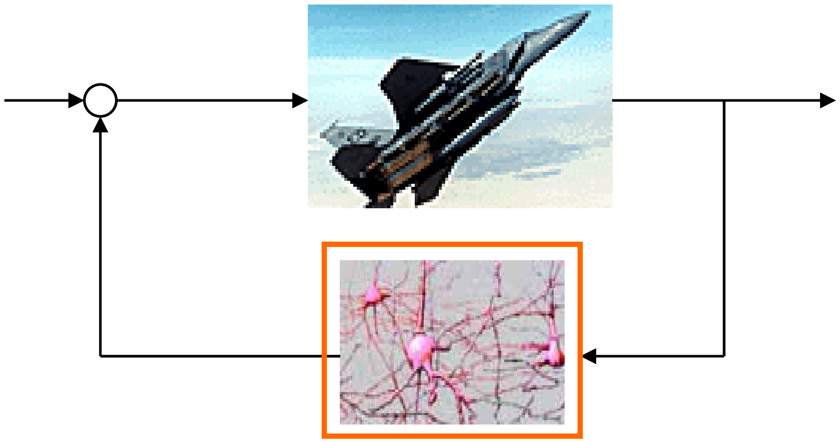
$$NN(x) = W^T \Phi(x)$$

- Function approximation using RBF NN means finding output weights *W*, centers *C*, and widths $\vec{\sigma}$

What is Next?

- Use SHL FNN-s in the context of MRAC systems
  - off-line / on-line approximation of uncertain nonlinearities in system dynamics
    - modeling errors, (aerodynamics)
    - battle damage
    - control failures
- Start with fixed widths RBF NN architectures, (linear in unknown parameters)
- Generalize to using sigmoidal NN-s

Adaptive NeuroControl



n^{th} Order Systems with Matched Uncertainties

- System Dynamics: $\dot{x} = A x + B \Lambda \left(u - f(x)\right), \quad x \in R^n, \quad u \in R^m$
  - $A \in R^{n \times n}, \Lambda = \operatorname{diag}\left(\lambda_1 \quad \ldots \quad \lambda_m\right) \in R^{m \times m}$ are constant *unknown* matrices
  - $B \in R^{n \times m}$ is *known* constant matrix
  - $\forall i = 1, \ldots, m \quad \operatorname{sgn}(\lambda_i)$ is *known*
- ***Approximation of uncertainty***: $f(x) = \Theta^T \Phi(x) + \varepsilon_f(x)$
  - *matrix* of constant *unknown* parameters: $\Theta \in R^{m \times N}$
  - vector of N *fixed* RBF-s: $\Phi(x) = \left(\varphi_1(x) \quad \ldots \quad \varphi_N(x)\right)^T$
  - function approximation tolerance: $\varepsilon_f(x) \in R^m$

n^{th} Order Systems with Matched Uncertainties, (continued)

- *Assumption*: Number of RBF-s, true (unknown) output weights *W* and widths $\vec{\sigma}$ are such that RBF NN approximates the nonlinearity within given tolerance:

$$\left\| \varepsilon_f(x) \right\| = \left\| f(x) - \Theta^T \Phi(x) \right\| \le \varepsilon, \quad \forall x \in X \subset R^n$$

- RBF NN estimator: $\hat{f}(x) = \hat{\Theta}^T \Phi(x)$
- Estimation error:

$$NN(x) - f(x) = \underbrace{\left(\hat{\Theta} - \Theta\right)}_{\Delta\Theta}^T \Phi(x) - \varepsilon_f(x) = \Delta\Theta^T \Phi(x) - \varepsilon_f(x)$$

n^{th} Order Systems with Matched Uncertainties, (continued)

- Stable Reference Model: $\dot{x}_m = A_m\, x_m + B_m\, r, \quad (A_m \text{ is Hurwitz})$

  $r \in R^m, \quad A_m \in R^{n\times n}, \quad B_m \in R^{m\times m}$
- **Control Goal**
  - bounded tracking: $\lim\limits_{t\to\infty} \left\| x(t) - x_m(t) \right\| \le \varepsilon_x$
- MRAC Design Process
  - choose *N* and vector of widths $\vec{\sigma}$
    - can be performed off-line in order to incorporate any prior knowledge about the uncertainty
  - design MRAC and evaluate closed-loop system performance
  - repeat previous two steps, if required

n^{th} Order Systems with Matched Uncertainties, (continued)

- Control Feedback: $u = \hat{K}_x^T x + \hat{K}_r^T r + \hat{\Theta}^T \Phi(x)$
  - ($m\,n + m^2 + m\,N$) - parameters to estimate: $\hat{K}_x, \quad \hat{K}_r, \quad \hat{\Theta}$
- Closed-Loop: $\dot{x} = \left(A + B\Lambda \hat{K}_x^T\right) x + B\Lambda\left(\hat{K}_r^T r + \Delta\Theta^T \Phi(x) - \varepsilon_f(x)\right)$
- Desired Dynamics: $\dot{x}_m = A_m x_m + B_m r$
- Model Matching Conditions
  - there exist *ideal* gains (K_x, K_r) such that: $\Rightarrow$ $A + B\Lambda K_x^T = A_m$, $B\Lambda K_r^T = B_m$
  - *Note*: knowledge of the ideal gains is not required

$$A + B\Lambda \hat{K}_x^T - A_m = A + B\Lambda \hat{K}_x^T - A - B\Lambda K_x^T = B\Lambda\left(\hat{K}_x - K_x\right)^T = B\Lambda \Delta K_x^T$$

$$B\Lambda \hat{K}_r^T - B_m = B\Lambda \hat{K}_r^T - B\Lambda K_r^T = B\Lambda\left(\hat{K}_r - K_r\right)^T = B\Lambda \Delta \hat{K}_r^T$$

n^{th} Order Systems with Matched Uncertainties, (continued)

- Tracking Error: $e(t) = x(t) - x_m(t)$
- Error Dynamics:

$$\dot{e}(t) = \dot{x}(t) - \dot{x}_m(t) =$$
$$\left(A + B\Lambda \hat{K}_x^T\right) x + B\Lambda\left(\hat{K}_r^T r + \Delta\Theta^T \Phi(x) - \varepsilon_f(x)\right) - A_m x_m - B_m r \pm A_m x$$
$$= A_m (x - x_m) + \left(A + B\Lambda \hat{K}_x^T - A_m\right) x + B\Lambda\left(\hat{K}_r - K_r\right)^T r + B\Lambda\left(\Delta\Theta^T \Phi(x) - \varepsilon_f(x)\right)$$
$$= A_m e + B\Lambda\left(\Delta K_x^T x + \Delta K_r^T r + \Delta\Theta^T \Phi(x) - \varepsilon_f(x)\right)$$

- Remarks
  – estimation error $\varepsilon_f(x)$ is bounded, as long as $x \in X$
  – need to keep x within X

n^{th} Order Systems with Matched Uncertainties, (continued)

- Lyapunov Function Candidate

$$V\left(e, \Delta K_x, \Delta K_r, \Delta\Theta\right) = e^T P\, e + \mathrm{trace}\left(\Delta K_x^T\, \Gamma_x^{-1}\, \Delta K_x \left|\Lambda\right|\right) + \mathrm{trace}\left(\Delta K_r^T\, \Gamma_r^{-1}\, \Delta K_r \left|\Lambda\right|\right) + \mathrm{trace}\left(\Delta\Theta^T\, \Gamma_\Theta^{-1}\, \Delta\Theta \left|\Lambda\right|\right)$$

  - where: $\mathrm{trace}(S) \triangleq \sum s_{ii}$
  - $\left|\Lambda\right| \triangleq \mathrm{diag}\left(\left|\lambda_1\right| \quad \ldots \quad \left|\lambda_m^i\right|\right)$ is diagonal matrix with positive elements
  - $\Gamma_x = \Gamma_x^T > 0, \quad \Gamma_r = \Gamma_r^T > 0, \quad \Gamma_\Theta = \Gamma_\Theta^T > 0$ are symmetric positive definite matrices
  - $P = P^T > 0$ is unique symmetric positive definite solution of the algebraic Lyapunov equation $P\,A + A^T\,P = -Q$
    - $Q = Q^T > 0$ is any symmetric positive definite matrix

n^{th} Order Systems with Matched Uncertainties, (continued)

- Time-derivative of the Lyapunov function

$$
\begin{aligned}
\dot{V} &= \dot{e}^T P e + e^T P \dot{e} \\
&+ 2\,\mathrm{trace}\left(\Delta K_x^T \Gamma_x^{-1} \dot{\hat{K}}_x |\Lambda|\right) + 2\,\mathrm{trace}\left(\Delta K_r^T \Gamma_r^{-1} \dot{\hat{K}}_r |\Lambda|\right) + 2\,\mathrm{trace}\left(\Delta \Theta^T \Gamma_\Theta^{-1} \dot{\hat{\Theta}} |\Lambda|\right) \\
&= \left(A_m e + B\Lambda\left(\Delta K_x^T x + \Delta K_r^T r + \Delta\Theta^T \Phi(x) - \varepsilon_f(x)\right)\right)^T P e \\
&+ e^T P\left(A_m e + B\Lambda\left(\Delta K_x^T x + \Delta K_r^T r + \Delta\Theta^T \Phi(x) - \varepsilon_f(x)\right)\right) \\
&+ 2\,\mathrm{trace}\left(\Delta K_x^T \Gamma_x^{-1} \dot{\hat{K}}_x |\Lambda|\right) + 2\,\mathrm{trace}\left(\Delta K_r^T \Gamma_r^{-1} \dot{\hat{K}}_r |\Lambda|\right) + 2\,\mathrm{trace}\left(\Delta \Theta^T \Gamma_\Theta^{-1} \dot{\hat{\Theta}} |\Lambda|\right) \\
&= e^T \left(A_m P + P A_m\right) e \\
&+ 2 e^T P B \Lambda\left(\Delta K_x^T x + \Delta K_r^T r + \Delta\Theta^T \Phi(x) - \varepsilon_f(x)\right) \\
&+ 2\,\mathrm{trace}\left(\Delta K_x^T \Gamma_x^{-1} \dot{\hat{K}}_x |\Lambda|\right) + 2\,\mathrm{trace}\left(\Delta K_r^T \Gamma_r^{-1} \dot{\hat{K}}_r |\Lambda|\right) + 2\,\mathrm{trace}\left(\Delta \Theta^T \Gamma_\Theta^{-1} \dot{\hat{\Theta}} |\Lambda|\right)
\end{aligned}
$$

n^{th} Order Systems with Matched Uncertainties, (continued)

- Time-derivative of the Lyapunov function

$$\begin{aligned}\dot{V} &= -e^T Q e - 2 e^T P B \Lambda \varepsilon_f(x) \\ &+ 2 e^T P B \Lambda \Delta K_x^T x + 2\,\text{trace}\left(\Delta K_x^T \Gamma_x^{-1} \dot{\hat{K}}_x |\Lambda|\right) \\ &+ 2 e^T P B \Lambda \Delta K_r^T r + 2\,\text{trace}\left(\Delta K_r^T \Gamma_r^{-1} \dot{\hat{K}}_r |\Lambda|\right) \\ &+ 2 e^T P B \Lambda \Delta \Theta^T \Phi(x) + 2\,\text{trace}\left(\Delta \Theta^T \Gamma_\Theta^{-1} \dot{\hat{\Theta}} |\Lambda|\right)\end{aligned}$$

- Using trace identity: $a^T b = \text{trace}\left(b\, a^T\right)$

- Example: $\underbrace{e^T P B \Lambda}_{a^T} \underbrace{\Delta K_x^T x}_{b} = \text{trace}\left(\underbrace{\Delta K_x^T x}_{b} \underbrace{e^T P B \Lambda}_{a^T}\right)$

n^{th} Order Systems with Matched Uncertainties, (continued)

- Time-derivative of the Lyapunov function

$$
\begin{aligned}
\dot{V} &= -e^T Q e - 2 e^T P B \Lambda \varepsilon_f(x) \\
&+ 2\,\mathrm{trace}\left(\Delta K_x^T \left\{ \Gamma_x^{-1} \dot{\hat{K}}_x + x e^T P B \,\mathrm{sgn}(\Lambda) \right\} |\Lambda| \right) \\
&+ 2\,\mathrm{trace}\left(\Delta K_r^T \left\{ \Gamma_r^{-1} \dot{\hat{K}}_r + r e^T P B \,\mathrm{sgn}(\Lambda) \right\} |\Lambda| \right) \\
&+ 2\,\mathrm{trace}\left(\Delta \Theta^T \left\{ \Gamma_\Theta^{-1} \dot{\hat{\Theta}} + \Phi(x) e^T P B \,\mathrm{sgn}(\Lambda) \right\} |\Lambda| \right)
\end{aligned}
$$

- Problem
  - choose adaptive parameters $\hat{K}_x, \hat{K}_r, \hat{\Theta}$ such that time-derivative $\dot{V}$ becomes negative definite outside of a compact set in the *error* state space, and all parameters remain bounded for all future times

n^{th} Order Systems with Matched Uncertainties, (continued)

- Suppose that we choose adaptive laws:

$$\begin{aligned}\dot{\hat{K}}_x &= -\Gamma_x \, x e^T P B \operatorname{sgn}(\Lambda) \\ \dot{\hat{K}}_r &= -\Gamma_r \, r e^T P B \operatorname{sgn}(\Lambda) \\ \dot{\hat{\Theta}} &= -\Gamma_\Theta \, \Phi(x) e^T P B \operatorname{sgn}(\Lambda)\end{aligned}$$

- Then we get:

$$\dot{V} = -e^T Q e - 2 e^T P B \Lambda \varepsilon_f(x) \le -\lambda_{\min}(Q)\|e\|^2 + 2\|e\|\|P B\|\lambda_{\max}(\Lambda)\varepsilon$$

- Consequently, $\dot{V} < 0$ outside of the compact set

$$E \triangleq \left\{ e : \|e\| \le \frac{2\|P B\|\lambda_{\max}(\Lambda)\varepsilon}{\lambda_{\min}(Q)} \right\}$$

- *Unfortunately*, inside *E* parameter errors may grow out of bounds, (for $e \in E, \quad \dot{V}$ IS NOT necessarily negative!)

How to Keep Adaptive Parameters Bounded?

- σ - modification:

$$\begin{aligned}\dot{\hat{K}}_x &= -\Gamma_x \left(x e^T P B + \sigma_x \hat{K}_x \right) \mathrm{sgn}(\Lambda) \\ \dot{\hat{K}}_r &= -\Gamma_r \left(r e^T P B + \sigma_r \hat{K}_r \right) \mathrm{sgn}(\Lambda) \\ \dot{\hat{\Theta}} &= -\Gamma_\Theta \left(\Phi(x) e^T P B + \sigma_\Theta \hat{\Theta} \right) \mathrm{sgn}(\Lambda)\end{aligned}$$

- e - modification:

$$\begin{aligned}\dot{\hat{K}}_x &= -\Gamma_x \left(x e^T P B + \sigma_x \left\| e^T P B \right\| \hat{K}_x \right) \mathrm{sgn}(\Lambda) \\ \dot{\hat{K}}_r &= -\Gamma_r \left(r e^T P B + \sigma_r \left\| e^T P B \right\| \hat{K}_r \right) \mathrm{sgn}(\Lambda) \\ \dot{\hat{\Theta}} &= -\Gamma_\Theta \left(\Phi(x) e^T P B + \sigma_\Theta \left\| e^T P B \right\| \hat{\Theta} \right) \mathrm{sgn}(\Lambda)\end{aligned}$$

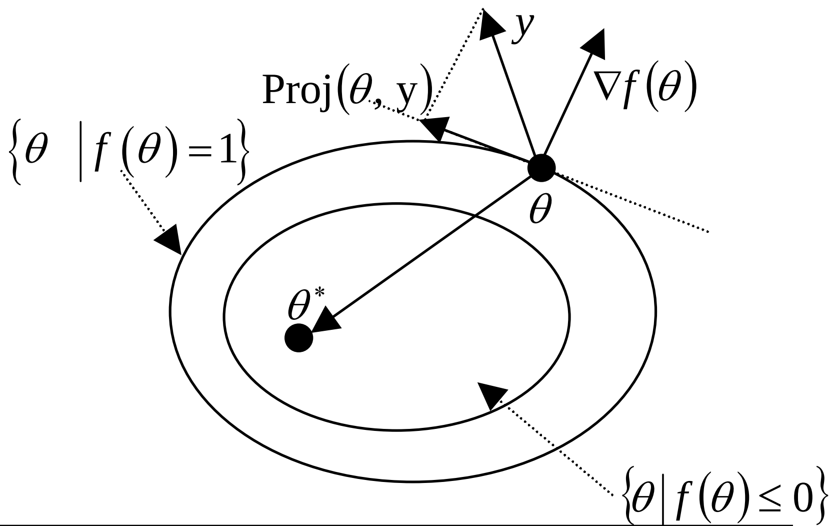
- Modifications add <u>*damping*</u> to adaptive laws
  - damping controlled by choosing $\boxed{\sigma_x, \sigma_r, \sigma_\Theta > 0}$
  - there is a <u>*trade off*</u> between adaptation rate and damping

Introducing Projection Operator

- Requires no damping terms
- Designed to keep NN weights within *pre-specified* bounds
- Maintains negative values of the Lyapunov function time-derivative outside of compact subset: $$E \triangleq \left\{ e : \|e\| \le \frac{2\|P\,B\|\,\lambda_{\max}(\Lambda)\,\varepsilon}{\lambda_{\min}(Q)} \right\}$$
  - the size of *E* defines tracking tolerance
  - the size of *E* can be controlled!

Projection Operator

- Function $f(\theta)$ defines pre-specified parameter domain boundary
- Example:



$$f(\theta) = \frac{\|\theta\|^2 - \theta_{\max}^2}{\varepsilon_\theta \, \theta_{\max}^2}$$

$$\{f(\theta) \le 0\} \Rightarrow \{\|\theta\| \le \theta_{\max}\} \Rightarrow \theta \text{ is within bounds}$$

$$\{0 < f(\theta) \le 1\} \Rightarrow \left\{\|\theta\| \le \sqrt{1+\varepsilon_\theta}\,\theta_{\max}\right\} \Rightarrow \theta \text{ is within } \left(\sqrt{1+\varepsilon_\theta}\right)\% \text{ of bounds}$$

$$\{f(\theta) > 1\} \Rightarrow \left\{\|\theta\| > \sqrt{1+\varepsilon_\theta}\,\theta_{\max}\right\} \Rightarrow \theta \text{ is outside of bounds}$$

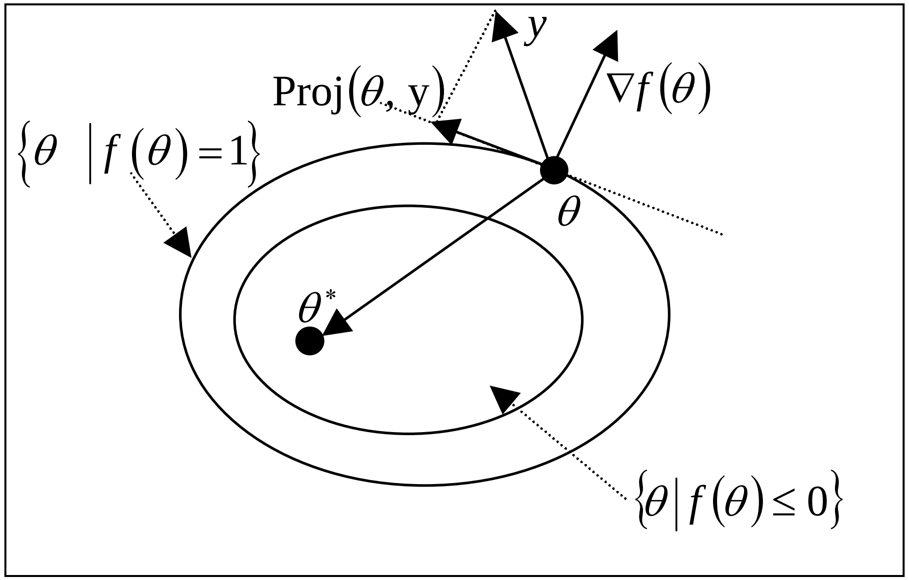
- $\theta_{\max}$ specifies boundary
- ε_θ specifies boundary tolerance

Projection Operator, (continued)

- ***Definition***:

$$\mathrm{Proj}(\theta, y) = \begin{cases} y - \dfrac{\nabla f(\theta)(\nabla f(\theta))^T}{\|\nabla f(\theta)\|^2} y\, f(\theta), & \text{if } f(\theta) > 0 \text{ and } y^T \nabla f(\theta) > 0 \\ y, & \text{if not} \end{cases}$$

- Depends on (θ, y)
- Does not alter *y* if θ is within the pre-specified bounds: $\|\theta\| \le \theta_{\max}$
- Gradient: $\nabla f(\theta) = \dfrac{2\theta}{\varepsilon_\theta \theta_{\max}^2}$
- In $\{0 \le f(\theta) \le 1\}$ the operator subtracts gradient vector $\nabla f(\theta)$ (normal to the boundary) from *y*
  - get a *smooth* transition from *y* for $\lambda = 0$ to a tangent vector field for $\lambda = 1$



- ***Important Property*** $\rightarrow$ $(\theta - \theta^*)^T (\mathrm{Proj}(\theta, y) - y) \le 0$

Lyapunov Function Time-Derivative with Projection Operator

- Make trace terms semi-negative ***AND*** keep parameters bounded:

$$
\begin{aligned}
\dot{V} &= -e^T Q e - 2 e^T P B \Lambda \varepsilon_f(x) \\
&+ 2\,\mathrm{trace}\left(\Delta K_x^T \left\{ \underbrace{\Gamma_x^{-1} \dot{\hat{K}}_x}_{\mathrm{Proj}(\hat{K}_x, y)} + \underbrace{x e^T P B \,\mathrm{sgn}(\Lambda)}_{-y} \right\} |\Lambda| \right) \\
&+ 2\,\mathrm{trace}\left(\Delta K_r^T \left\{ \underbrace{\Gamma_r^{-1} \dot{\hat{K}}_r}_{\mathrm{Proj}(\hat{K}_r, y)} + \underbrace{r e^T P B \,\mathrm{sgn}(\Lambda)}_{-y} \right\} |\Lambda| \right) \\
&+ 2\,\mathrm{trace}\left(\Delta \Theta^T \left\{ \underbrace{\Gamma_\Theta^{-1} \dot{\hat{\Theta}}}_{\mathrm{Proj}(\hat{\Theta}, y)} + \underbrace{\Phi(x) e^T P B \,\mathrm{sgn}(\Lambda)}_{-y} \right\} |\Lambda| \right)
\end{aligned}
$$

Adaptation with Projection

- Modified adaptive laws:

$$\begin{aligned}\dot{\hat{K}}_x &= \Gamma_x \operatorname{Proj}\left(\hat{K}_x, -x\, e^T P\, B \operatorname{sgn}(\Lambda)\right) \\ \dot{\hat{K}}_r &= \Gamma_r \operatorname{Proj}\left(\hat{K}_r, -r\, e^T P\, B \operatorname{sgn}(\Lambda)\right) \\ \dot{\hat{\Theta}} &= \Gamma_\Theta \operatorname{Proj}\left(\hat{\Theta}, -\Phi(x)\, e^T P\, B \operatorname{sgn}(\Lambda)\right)\end{aligned}$$

- Projection Operator, its bounds and tolerances are defined ***column-wise***
- Lyapunov function time-derivative:

$$\dot{V} \le -e^T Q\, e - 2\, e^T P\, B\, \Lambda\, \varepsilon_f(x) \le -\lambda_{\min}(Q)\|e\|^2 + 2\|e\|\|P\, B\|\lambda_{\max}(\Lambda)\,\varepsilon$$

- Adaptive parameters stay within the pre-specified bounds, while $\dot{V} < 0$ outside of the compact set:

$$E \triangleq \left\{ e : \|e\| \le \frac{2\|P\, B\|\lambda_{\max}(\Lambda)\,\varepsilon}{\lambda_{\min}(Q)} \right\}$$

Example: Projection Operator, (scalar case)

- Scalar adaptive gain: $\dot{\hat{k}} = \gamma \, \mathrm{Proj}\left(\hat{k}, -x\,e\,\mathrm{sgn}(b)\right)$
- Pre-specified parameter domain boundary:
  - using function: $f\left(\hat{k}\right) = \dfrac{\hat{k}^2 - k_{\max}^2}{\varepsilon\, k_{\max}^2}$ ⟹ $\nabla f\left(\hat{k}\right) = f'\left(\hat{k}\right) = \dfrac{2\,\hat{k}}{\varepsilon\, k_{\max}^2}$

$$\left\{ f\left(\hat{k}\right) \le 0 \right\} \Rightarrow \left\{ \left|\hat{k}\right| \le k_{\max} \right\} \Rightarrow \hat{k} \text{ is within bounds}$$

$$\left\{ 0 < f\left(\hat{k}\right) \le 1 \right\} \Rightarrow \left\{ \left|\hat{k}\right| \le \sqrt{1+\varepsilon}\, k_{\max} \right\} \Rightarrow \hat{k} \text{ is within } \left(\sqrt{1+\varepsilon}\right)\% \text{ of bounds}$$

$$\left\{ f\left(\hat{k}\right) > 1 \right\} \Rightarrow \left\{ \left|\hat{k}\right| > \sqrt{1+\varepsilon}\, k_{\max} \right\} \Rightarrow \hat{k} \text{ is outside of bounds}$$

  - Projection Operator:

$y = -x\,e\,\mathrm{sgn}(b)$ ⟹

$$\mathrm{Proj}\left(\hat{k}, y\right) = \begin{cases} y\left(1 - f\left(\hat{k}\right)\right), & \text{if } f\left(\hat{k}\right) > 0 \text{ and } y\, f'\left(\hat{k}\right) > 0 \\ y, & \text{if not} \end{cases}$$

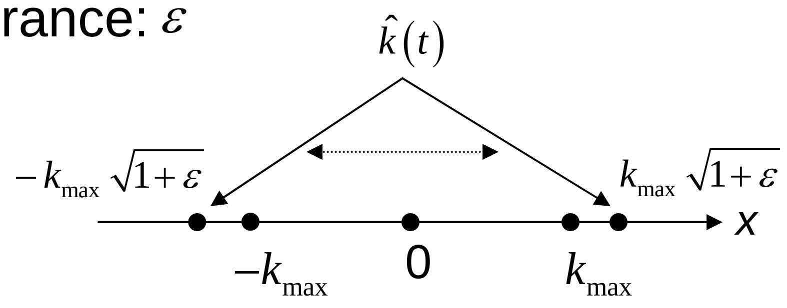
Example: Projection Operator, (scalar case) (continued)

- Adaptive Law, ($b > 0$):

$$\dot{\hat{k}} = \begin{cases} -x\,e\left(1 - f\left(\hat{k}\right)\right), & \text{if } \left[f\left(\hat{k}\right) > 0 \text{ and } x\,e\,f'\left(\hat{k}\right) \right] < 0 \\ -x\,e, & \text{if not} \end{cases}$$

$$\text{where: } f\left(\hat{k}\right) = \frac{\hat{k}^2 - k_{\max}^2}{\varepsilon\, k_{\max}^2}$$

- Geometric Interpretation
  - adaptive parameter $\hat{k}(t)$ changes within the pre-specified interval
  - interval bound: $k_{\max}$
  - Bound tolerance: ε



Adaptive *Augmentation* Design

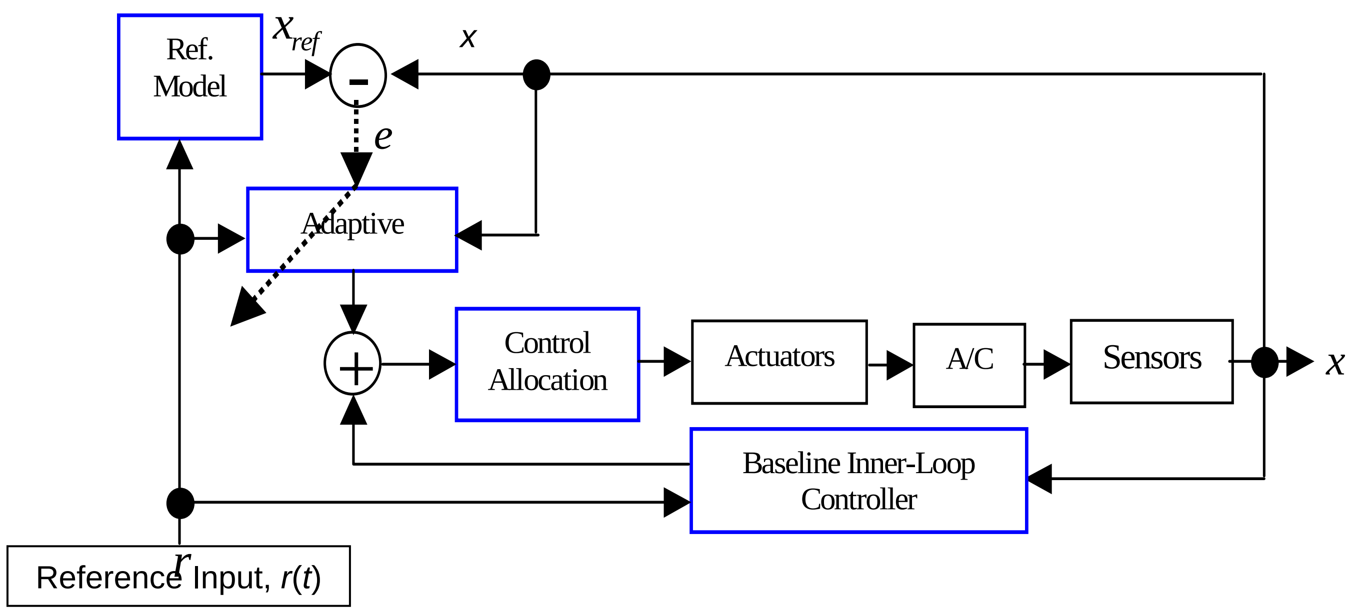
- Nominal Control: $u_{nom} = F_x^T\, x + F_r^T\, r$
- Adaptive Control: $u = \hat{K}_x^T\, x + \hat{K}_r^T\, r + \hat{\Theta}^T\, \Phi(x)$
- Augmentation:

$$
\begin{aligned}
u &= \hat{K}_x^T\, x + \hat{K}_r^T\, r + \hat{\Theta}^T\, \Phi(x) \pm u_{nom} \\
&= u_{nom} + \underbrace{\left(\hat{K}_x - F_x\right)}_{\hat{D}_x}^T x + \underbrace{\left(\hat{K}_r - F_r\right)}_{\hat{D}_r}^T r + \hat{\Theta}^T\, \Phi(x) \\
&= u_{nom} + \hat{D}_x^T\, x + \hat{D}_r^T\, x + \hat{\Theta}^T\, \Phi(x)
\end{aligned}
$$

- *Incremental* Adaptation:

$$
\begin{aligned}
\dot{\hat{D}}_x &= \Gamma_x \operatorname{Proj}\left(\hat{D}_x,\; -x\, e^T P\, B \operatorname{sgn}(\Lambda)\right), \quad \hat{D}_x = 0_{n\times m} \\
\dot{\hat{D}}_r &= \Gamma_r \operatorname{Proj}\left(\hat{D}_r,\; -r\, e^T P\, B \operatorname{sgn}(\Lambda)\right), \quad \hat{D}_r = 0_{m\times m} \\
\dot{\hat{\Theta}} &= \Gamma_\Theta \operatorname{Proj}\left(\hat{\Theta}, -\Phi(x)\, e^T P\, B \operatorname{sgn}(\Lambda)\right), \quad \hat{\Theta} = 0_{N\times m}
\end{aligned}
$$

Adaptive *Augmentation* Block-Diagram



- Reference Model provides desired response
- Nominal Baseline Controller
- Adaptive Augmentation
  - *Dead-Zone* modification prevents adaptation from changing nominal closed-loop dynamics
  - *Projection Operator* bounds adaptation parameters / gains

Adaptive Control using *Sigmoidal* NN

- System Dynamics: $\dot{x} = A\,x + B\,\Lambda\left(u - f\left(x\right)\right), \quad x \in R^n, \quad u \in R^m$
  - $A \in R^{n\times n}, \Lambda = \mathrm{diag}\left(\lambda_1 \quad \ldots \quad \lambda_m\right) \in R^{m\times m}$ are constant ***unknown*** matrices
  - $B \in R^{M\times m}$ is *known* constant matrix, and $M \ge m$
  - $\forall\, i = 1, \ldots, m \quad \mathrm{sgn}\left(\lambda_i\right)$ is *known*
- ***Approximation of uncertainty***:

$$f\left(x\right) = W^T\,\vec{\sigma}\left(V^T\,\mu\right) + \varepsilon_f\left(x\right), \quad \mu = \left(x^T \quad 1\right)^T, \quad \varepsilon_f\left(x\right) \in R^m$$

  - *matrix* of constant *unknown* *Inner-Layer* weights: $V = \begin{bmatrix} \vec{v}_1 & \ldots & \vec{v}_N \\ \theta_1 & \ldots & \theta_N \end{bmatrix} \in R^{(n+1)\times N}$
  - *matrix* of constant *unknown* *Outer-Layer* weights: $W = \begin{bmatrix} \vec{w}_1 & \ldots & \vec{w}_m \\ c_1 & \ldots & c_m \end{bmatrix} \in R^{(N+1)\times m}$
  - vector of *N* *sigmoids* and a unity:

$$\vec{\sigma}\left(V^T\mu\right) = \left(\sigma\left(v_1^T\,x + \theta_1\right) \quad \ldots \quad \sigma\left(\vec{v}_N^T\,x + \theta_N\right) \quad 1\right)^T, \quad \text{where: } \sigma\left(s\right) = \frac{1}{1+e^{-s}}$$

Adaptive Control using *Sigmoidal* NN

- Control Feedback: $u = \hat{K}_x^T x + \hat{K}_r^T r + \hat{W}^T \vec{\sigma}\left(\hat{V}^T \mu\right)$
  - $(m\,n + m^2 + (n+1)\,N + (N+1)\,m)$ - parameters to estimate: $\hat{K}_x,\ \hat{K}_r,\ \hat{W},\ \hat{V}$
- Adaptation with Projection, $(\Lambda > 0)$:

$$
\begin{cases}
\dot{\hat{K}}_x = \Gamma_x \operatorname{Proj}\left(\hat{K}_x,\ -x\,e^T P\,B\right) \\
\dot{\hat{K}}_u = \Gamma_u \operatorname{Proj}\left(\hat{K}_u,\ -r\,e^T P\,B\right) \\
\dot{\hat{W}} = \Gamma_W \operatorname{Proj}\left(\hat{W},\ \left(\vec{\sigma}\left(\hat{V}^T \mu\right) - \vec{\sigma}'\left(\hat{V}^T \mu\right)\hat{V}^T \mu\right) e^T P\,B\right) \\
\dot{\hat{V}} = \Gamma_V \operatorname{Proj}\left(\hat{V},\ \mu\, e^T P\,B\,\hat{W}^T \vec{\sigma}'\left(\hat{V}^T \mu\right)\right)
\end{cases}
$$

- Provides ***bounded*** tracking

Design Example
Adaptive Reconfigurable Flight Control using RBF NN-s

Aircraft Model

- Flight Dynamics Approximation, (constant speed):

$$\dot{x}_p = A_p\, x_p + \underbrace{B\, G}_{B_p} \Lambda\left(\delta + K_0\left(x_p\right)\right) = A_p\, x_p + B_p\, \Lambda\left(\delta + K_0\left(x_p\right)\right)$$

  - State: $x_p = \left(\alpha \quad \beta \quad p \quad q \quad r\right)^T$
  - Control allocation matrix *G*
  - *Virtual* Control Input: $\delta \in R^3$
  - Modeling control uncertainty / failures by $\Lambda \in R^{3\times 3}$ diagonal matrix with positive elements
  - Vector of actual control inputs:

    $G\,\Lambda\,\delta = \left(\delta_{LOB} \quad \delta_{LMB} \quad \delta_{LIB} \quad \delta_{RIB} \quad \delta_{RMB} \quad \delta_{ROB} \quad \delta_{Tvec}\right)^T \in R^7$
  - A_p, B_p are *known* matrices
    - represent nominal system dynamics
  - *Matched* unknown nonlinear effects: $K_0\left(x_p\right) \in R^3$

Baseline Inner-Loop Controller

- Dynamics: $\dot{x}_c = A_c\, x_c + B_{1\,c}\, x_p + B_{2\,c}\, u$
- States: $x_c = \begin{pmatrix} q_I & p_I & r_I & r_w \end{pmatrix}^T \in R^4$
- Inner-loop commands, (reference input):

$$u = \begin{pmatrix} a_z^{cmd} & \beta^{cmd} & p^{cmd} & r^{cmd} \end{pmatrix}^T$$

- System output: $a_z = C_p\, x_p + \underbrace{D\,G}_{D_p}\Lambda\left(\delta + K_0\left(x_p\right)\right) = C_p\, x_p + D_p\,\Lambda\left(\delta + K_0\left(x_p\right)\right)$
- Augmented system dynamics:

$$\underbrace{\begin{pmatrix} \dot{x}_p \\ \dot{x}_c \end{pmatrix}}_{\dot{x}} = \underbrace{\begin{pmatrix} A_p & 0 \\ B_{1c} & A_c \end{pmatrix}}_{A}\underbrace{\begin{pmatrix} x_p \\ x_c \end{pmatrix}}_{x} + \underbrace{\begin{pmatrix} B_p \\ 0 \end{pmatrix}}_{B_1}\Lambda\left(\delta + K_0\left(x_p\right)\right) + \underbrace{\begin{pmatrix} 0 \\ B_{2c} \end{pmatrix}}_{B_2} u$$

$$\dot{x} = A\,x + B_1\,\Lambda\left(\delta + K_0\left(x_p\right)\right) + B_2\,u$$

- Inner-Loop Control: $\delta_L = K_x^T\, x + K_u^T\, u$

Reference Model

- Assuming nominal data, $\left(\Lambda = I_{3\times 3}, \quad K_0\left(x_p\right) = 0_{3\times 1}\right)$, and using baseline controller:

$$\dot{x}_{ref} = \underbrace{\left(A + B_1 K_x^T\right)}_{A_{ref}} x_{ref} + \underbrace{\left(B_2 + B_1 K_u^T\right)}_{B_{ref}} u = A_{ref}\, x_{ref} + B_{ref}\, u$$

- ***Assumption***: Reference model matrix A_{ref} is Hurwitz, (i.e., baseline controller stabilizes nominal system)

Inner-Loop Control Objective
(Bounded Tracking)

- Design virtual control input such that, despite system uncertainties, the system state tracks the state of the reference model, while all closed-loop signals remain bounded
- Solution
  - Incremental, (i.e., adaptive augmentation), MRAC system with RBF NN, Dead-Zone, and Projection Operator

Adaptive Augmentation

- Total control input:

$$\delta = \underbrace{\hat{K}_x^T\, x + \hat{K}_u^T\, u - \hat{K}_0\left(x_p\right)}_{\text{Total Adaptive Control}} \pm \underbrace{\delta_L\left(x,u\right)}_{\text{Nominal Baseline}}$$

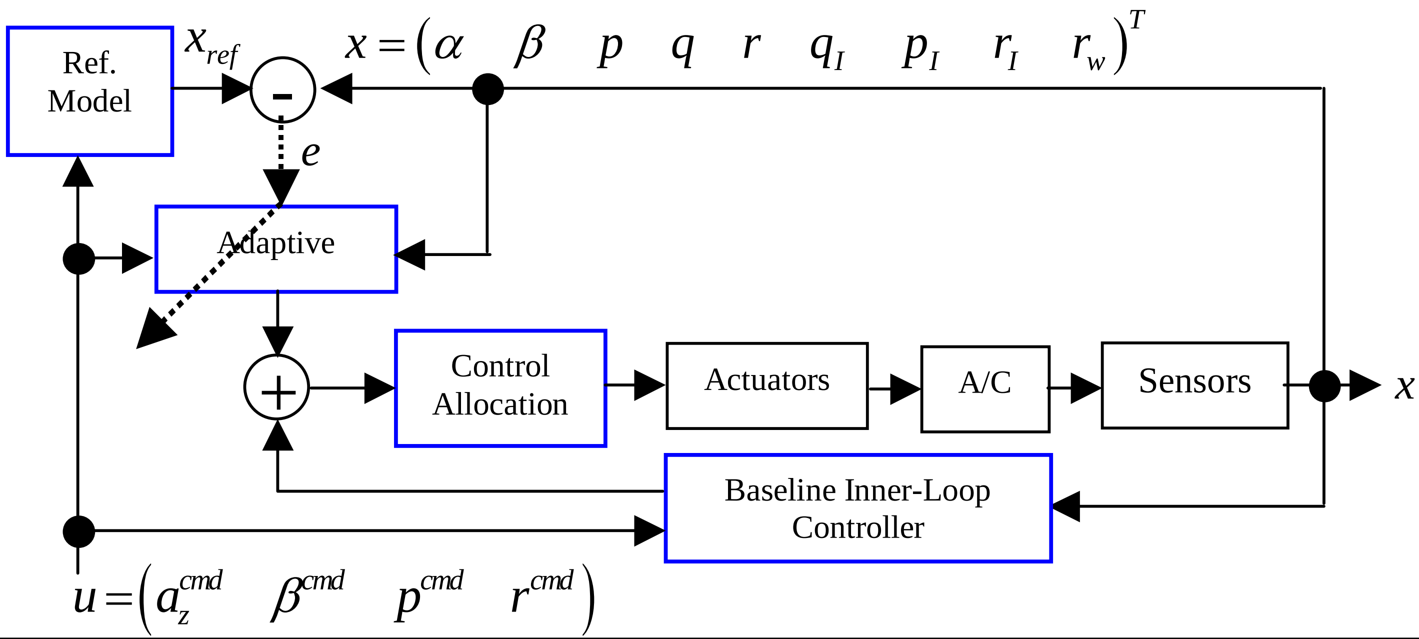
$$= \underbrace{\delta_L\left(x,u\right)}_{\delta_L\left(x,u\right)} + \underbrace{\left(\hat{K}_x - K_x\right)}_{\hat{k}_x}{}^T\, x + \underbrace{\left(\hat{K}_u - K_u\right)}_{\hat{k}_u}{}^T\, u - \underbrace{\hat{K}_0\left(x_p\right)}_{\hat{\Theta}^T\,\Phi\left(x_p\right)}$$

$$= \underbrace{\delta_L\left(x_p, x_c, u\right)}_{\text{Nominal Baseline}} + \underbrace{\Delta\hat{K}_x^T\, x + \Delta\hat{K}_u^T\, u - \hat{\Theta}^T\,\Phi\left(x_p\right)}_{\text{Incremental Adaptive Control}}$$

- Incremental adaptation with projection:

$$\begin{cases} \Delta\dot{\hat{K}}_x = \Gamma_x\ \operatorname{Proj}\left(\Delta\hat{K}_x,\ -x\,e^T\,P\,B_1\right), & \Delta\hat{K}_x\left(0\right) = 0_{n\times 3} \\ \Delta\dot{\hat{K}}_u = \Gamma_u\ \operatorname{Proj}\left(\Delta\hat{K}_u,\ -u\,e^T\,P\,B_1\right), & \Delta\hat{K}_u\left(0\right) = 0_{n\times 4} \\ \dot{\hat{\Theta}} = \Gamma_\Theta\ \operatorname{Proj}\left(\hat{\Theta},\ \Phi\left(x_p\right)e^T\,P\,B_1\right), & \hat{\Theta}\left(0\right) = 0_{N\times m} \end{cases}$$

Inner-Loop Block-Diagram



- Reference Model provides desired response
- Nominal Baseline Inner-Loop Controller
- Adaptive Augmentation
  - *Dead-Zone* modification prevents adaptation from changing nominal closed-loop dynamics
  - *Projection Operator* bounds adaptation parameters / gains

Adaptive Backstepping

Why?

- MRAC requires model matching conditions

$$A + B\Lambda K_x^T = A_m$$
$$B\Lambda K_r^T = B_m$$

- Example that violates matching
  - System: $\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \end{pmatrix} = \underbrace{\begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}}_{A} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \underbrace{\begin{pmatrix} 0 \\ 1 \end{pmatrix}}_{b} u$
  - Reference model: $\begin{pmatrix} \dot{x}_1^m \\ \dot{x}_2^m \end{pmatrix} = \underbrace{\begin{pmatrix} -1 & 1 \\ 0 & -2 \end{pmatrix}}_{A_m} \begin{pmatrix} x_1^m \\ x_2^m \end{pmatrix} + \begin{pmatrix} 0 \\ 1 \end{pmatrix} r$

Matching conditions don't hold

$$A - A_m = \begin{pmatrix} 1 & 0 \\ 0 & 2 \end{pmatrix} \neq b\, k_x^T = \begin{pmatrix} 0 & 0 \\ * & * \end{pmatrix}$$

Control Tracking Problem

- Consider 2nd order ***cascaded*** system

$$\dot{x}_1 = f_1(x_1) + g_1(x_1)\,x_2$$
$$\dot{x}_2 = f_2(x_1, x_2) + g_2(x_1, x_2)\,u$$

- Control goal
  - Choose *u* such that: $x_1(t) \to x_1^{com}(t), \quad \text{as } t \to \infty$
- Assumptions
  - All functions are known
  - $g_i \neq 0$ does not cross zero
- Example: AOA tracking

$$\begin{cases} \dot{\alpha} = \underbrace{-L_\alpha(\alpha)}_{f_1}\overbrace{\alpha}^{x_1} + \underbrace{1}_{g_1}\overbrace{q}^{x_2} \\ \dot{q} = \underbrace{M_0(\alpha, q)}_{f_2} + \underbrace{1}_{g_2}\underbrace{\dot{q}_{cmd}}_{u} \end{cases}$$

Backstepping Design

- Introduce pseudo control: $x_2^{com} = x_2^{com}(t)$
- Rewrite the 1st equation:

$$\dot{x}_1 = f_1(x_1) + g_1(x_1)\, x_2^{com} + g_1(x_1)\underbrace{\left(x_2 - x_2^{com}\right)}_{\Delta x_2}$$

- Dynamic inversion using pseudo control:

$$x_2^{com} = \frac{1}{g_1(x_1)}\left(\dot{x}_1^{com} - f_1(x_1) - k_1\,\Delta x_1\right)$$

- 1st state error dynamics: $\Delta\dot{x}_1 = -k_1\,\Delta x_1 + g_1(x_1)\,\Delta x_2$

Backstepping Design
(continued)

- Dynamic inversion using actual control

$$u = \frac{1}{g_2(x_1, x_2)}\left(\dot{x}_2^{com} - f_2(x_1, x_2) - k_2\,\Delta x_2 - g_1(x_1)\Delta x_1\right)$$

- 2[nd] state error dynamics

$$\Delta\dot{x}_2 = -k_2\,\Delta x_2 - g_1(x_1)\Delta x_1$$

- ***Asymptotically stable*** error dynamics

$$\begin{pmatrix}\Delta\dot{x}_1\\ \Delta\dot{x}_2\end{pmatrix} = \begin{pmatrix}-k_1 & g_1(x_1)\\ -g_1(x_1) & -k_2\end{pmatrix}\begin{pmatrix}\Delta x_1\\ \Delta x_2\end{pmatrix}$$

nonlinear system

- Conclusion: $x_i(t) \to x_i^{com}(t), \quad \text{as } t \to \infty$

Adaptive Backstepping Design

- 1st state dynamics: $\dot{x}_1 = \hat{f}_1 + \hat{g}_1\, x_2^{com} + \hat{g}_1\, \Delta x_2 - \Delta f_1 - \Delta g_1\, u$
  - Function estimation errors:

$$\Delta f_1 \triangleq \hat{f}_1 - f_1, \quad \Delta g_1 \triangleq \hat{g}_1 - g_1$$

- Dynamic inversion using pseudo control and estimated functions:

$$x_2^{com} = \frac{1}{\hat{g}_1(x_1)}\left(\dot{x}_1^{com} - \hat{f}_1(x_1) - k_1\, \Delta x_1\right)$$

- 1st state error dynamics:

$$\Delta \dot{x}_1 = -k_1\, \Delta x_1 + \hat{g}_1\, \Delta x_2 - \Delta f_1 - \Delta g_1\, u$$

Adaptive Backstepping Design (continued)

- 2nd state dynamics: $\dot{x}_2 = \hat{f}_2 + \hat{g}_2\, u - \Delta f_2 - \Delta g_2\, u$
  - Function estimation errors:

$$\Delta f_2 \triangleq \hat{f}_2 - f_2, \quad \Delta g_2 \triangleq \hat{g}_2 - g_2$$

- Dynamic inversion using actual control and estimated functions:

$$u = \frac{1}{\hat{g}_2(x_1, x_2)}\left(\dot{x}_2^{com} - \hat{f}_2(x_1, x_2) - k_2\, \Delta x_2 - \hat{g}_1(x_1)\, x_1\right)$$

- 2nd state error dynamics:

$$\Delta\dot{x}_2 = -k_2\, \Delta x_2 - \hat{g}_1\, \Delta x_1 - \Delta f_2 - \Delta g_2\, u$$

Adaptive Backstepping Design
(continued)

- Combined error dynamics:

$$\begin{pmatrix} \Delta\dot{x}_1 \\ \Delta\dot{x}_2 \end{pmatrix} = \begin{pmatrix} -k_1 & \hat{g}_1(x_1) \\ -\hat{g}_1(x_1) & -k_2 \end{pmatrix} \begin{pmatrix} \Delta x_1 \\ \Delta x_2 \end{pmatrix} + \begin{pmatrix} -\Delta f_1 - \Delta g_1 u \\ -\Delta f_2 - \Delta g_2 u \end{pmatrix}$$

- Uncertainty parameterization, function and parameter estimation errors:

$$\Delta f_i = \Delta\theta_{f_i}^T \Phi_f(x_1, x_2) - \varepsilon_{f_i}$$
$$\Delta g_i = \Delta\theta_{g_i}^T \Phi_g(x_1, x_2) - \varepsilon_{g_i}$$

$$\Delta\theta_{f_i} \triangleq \hat{\theta}_{f_i} - \theta_{f_i}$$
$$\Delta\theta_{g_i} \triangleq \hat{\theta}_{g_i} - \theta_{g_i}$$

Adaptive Backstepping Design (continued)

- Tracking error dynamics:

$$\underbrace{\begin{pmatrix}\Delta\dot{x}_1\\ \Delta\dot{x}_2\end{pmatrix}}_{\dot{e}} = \underbrace{\begin{pmatrix}-k_1 & \hat{g}_1\\ -\hat{g}_1 & -k_2\end{pmatrix}}_{A}\underbrace{\begin{pmatrix}\Delta x_1\\ \Delta x_2\end{pmatrix}}_{e} - \underbrace{\begin{pmatrix}\Delta\theta_{f_1}^T & \Delta\theta_{g_1}^T\\ \Delta\theta_{f_2}^T & \Delta\theta_{g_2}^T\end{pmatrix}}_{\Delta\Theta^T}\underbrace{\begin{pmatrix}\Phi_f\\ \Phi_g\, u\end{pmatrix}}_{\Phi} + \underbrace{\begin{pmatrix}\varepsilon_{f_1} + \varepsilon_{g_1} u\\ \varepsilon_{f_2} + \varepsilon_{g_2} u\end{pmatrix}}_{\varepsilon}$$

$$\Updownarrow$$

$$\dot{e} = A e - \Delta\Theta^T\,\Phi + \varepsilon$$

- Stable robust adaptive laws:

$$\dot{\hat{\Theta}} = \Gamma\,\mathrm{Proj}\left(\hat{\Theta},\quad \Phi\, e^T\right)$$

- *Conclusion*: Bounded tracking

Adaptive Control in the Presence of Actuator Constraints*

*E. Lavretsky and N. Hovakimyan, "Positive μ – modification for stable adaptation in the presence of input constraints," ACC, 2004.

Current Status

- Problem: Enforce robustness and assure stability of an adaptive control system in the presence of unmodelled dynamics, position, and rate saturation constraints that are due to physical limitations of an actuator
- Solutions (state of the art)
  - Ad-hoc
  - Proof-by-simulation
  - Lyapunov based
    - limited class of systems
    - sufficient conditions are often hard-to-verify

not acceptable

preferred

- Need: *Theoretically justified* and *verifiable* conditions for stable adaptation and control design in the presence of actuator dynamics, position, and rate limits
- Design Solutions include modifications, (adaptive / fixed gain) to:
  - control input
  - tracking error
  - reference model

Design Solutions

- R. Monopoli, (1975)
  - adaptive modifications: tracking error and reference input
  - no theoretical stability proof
- S.P. Karason, A.M. Annaswamy, (1994)
  - adaptive modifications: reference input
  - rigorous stability proof
- E.N. Johnson, A.J. Calise, (2003)
  - pseudo control hedging (PCH)
    - fixed gain modification: reference input
- E. Lavretsky, H. Hovakimyan, (2004)
  - positive μ – modification
    - adaptive modification: reference input
    - rigorous stability proof
    - avoids control saturation

Adaptive Control in the Presence of Input Constraints: Problem Formulation

- System dynamics: $\dot{x}(t) = A x(t) + b\,\lambda\, u(t), \quad x \in R^n, u \in R$
  - *A* is ***unknown*** matrix
  - *b* is ***known*** control direction
  - $\lambda > 0$ is ***unknown*** positive constant
- Amplitude limited control input

$$u(t) = u_{\max}\,\text{sat}\left(\frac{u_c}{u_{\max}}\right) = \begin{cases} u_c(t), & |u_c(t)| \le u_{\max} \\ u_{\max}\,\text{sgn}(u_c(t)), & |u_c(t)| \ge u_{\max} \end{cases}$$

amplitude saturation

commanded input

- ***Ideal*** Reference model dynamics:

$$\dot{x}_m^*(t) = A_m\, x_m^*(t) + b_m\, r(t), \quad x_m^* \in R^n, r \in R$$

Hurwitz

bounded reference input

Problem Formulation

- Control Design Goals:
  - Find commanded input $u_c(t)$ and, if necessary, augment reference input $r(t)$ such that system state $x(t)$ asymptotically tracks new reference input, while all the signals remain bounded
  - Avoid control saturation at all times
  - Provide theoretical proof of stability and verifiable sufficient conditions

Preliminaries

- Rewrite system dynamics: $\dot{x} = A x + b \lambda (u_c + \Delta u), \quad \Delta u = u - u_c$

  control deficiency

- Define: $u_{\max}^{\delta} = u_{\max} - \delta, \quad \text{where: } 0 < \delta < u_{\max}$
- Commanded control deficiency: → $\Delta u_c = u_{\max}^{\delta} \operatorname{sat}\left(\frac{u_c}{u_{\max}^{\delta}} \right) - u_c$
- Adaptive control with μ – modification, (*implicit* form):

$$u_c = \underbrace{k_x^T x + k_r r}_{u_{lin}} + \boxed{\mu \Delta u_c}$$

linear feedback / feedforward component

control deficiency feedback

Positive μ – modification

- Commanded control with μ – mod is given by *convex combination* of u_{lin} and $u_{\max}^{\delta}\,\mathrm{sat}\left(\dfrac{u_{lin}}{u_{\max}^{\delta}}\right)$

$$u_c = \frac{1}{1+\mu}\left(u_{lin} + \mu\, u_{\max}^{\delta}\,\mathrm{sgn}\left(\frac{u_{lin}}{u_{\max}^{\delta}}\right)\right) = \begin{cases} u_{lin}, & \left|u_{lin}\right| \le u_{\max}^{\delta} \\ \dfrac{1}{1+\mu}\left(u_{lin} + \mu\, u_{\max}^{\delta}\right), & u_{lin} > u_{\max}^{\delta} \\ \dfrac{1}{1+\mu}\left(u_{lin} - \mu\, u_{\max}^{\delta}\right), & u_{lin} < -u_{\max}^{\delta} \end{cases}$$

continuous in time but *not* continuously differentiable

Closed-Loop Dynamics

- μ – mod control: $u_c = u_{lin} + \mu\,\Delta u_c$
- System dynamics: $\dot{x} = A\,x + b\,\lambda\,u_c + b\,\lambda\,\Delta u$
- Closed-loop system:

linear control deficiency

$$\dot{x} = A\,x + b\,\lambda\,u_{lin} + b\,\lambda\overbrace{\left(\mu\,\Delta u_c + \Delta u\right)}^{\Delta u_{lin}}$$

$$\text{where: } \Delta u_{lin} = u_{\max}\,\text{sat}\left(\frac{u_c}{u_{\max}}\right) - u_{lin}$$

$\Updownarrow$

explicitly doesn't depend on μ

$$\dot{x} = \left(A + b\,\lambda\,k_x^T\right)x + b\,\lambda\,k_r\,r + b\,\lambda\,\Delta u_{lin}$$

Adaptive Reference Model

- Closed-loop system:

$$\dot{x} = \left(A + b\,\lambda\,k_x^T\right)x + b\,\lambda\left(k_r\,r + \Delta u_{lin}\right)$$

- Leads to consideration of ***adaptive*** reference model:

adaptive augmentation

$$\dot{x}_m = A_m\,x_m + b_m\left(r(t) + k_u\,\Delta u_{lin}\right), \quad \left|r(t)\right| \le r_{\max}$$

reference input

- Matching conditions:

$$\forall\,\lambda > 0\ \exists\left(k_x^* \in R^n,\quad k_r^* \in R,\quad k_u^* \in R\right) \Longrightarrow \begin{cases} A + b\,\lambda\left(k_x^T\right)^* = A_m \\ \left.\begin{array}{l} b\,\lambda\,k_r^* = b_m \\ b\,\lambda = b_m\,k_u^* \end{array}\right\} \Rightarrow k_u^*\,k_r^* = 1 \end{cases}$$

Adaptive Laws Derivation

- Tracking error: $e = x - x_m$
- Parameter errors: $\begin{cases} \Delta k_x = k_x - k_x^* \\ \Delta k_r = k_r - k_r^* \\ \Delta k_u = k_u - k_u^* \end{cases}$
- Tracking error dynamics:

$$\dot{e} = A_m\, e + b\,\lambda\left(\Delta k_x^T\, x + \Delta k_r\, r\right) - b_m\, \Delta k_u\, \Delta u_{lin}$$

- Lyapunov function:

$$V\left(e, \Delta k_x, \Delta k_r, \Delta k_u\right) = e^T\, P\, e + \lambda\left(\Delta k_x^T\, \Gamma_x^{-1}\, \Delta k_x + \gamma_r^{-1}\, \Delta k_r^2 + \gamma_u^{-1}\, \Delta k_u^2\right)$$

where: $P\, A_m + A_m\, P = -Q < 0$

Stable Parameter Adaptation

- Adaptive laws derived to yield stability:

$$\begin{aligned} \dot{k}_x &= -\Gamma_x\, x\, e^T P\, b \\ \dot{k}_r &= -\gamma_r\, r(t)\, e^T P\, b \\ \dot{k}_u &= \gamma_u\, \Delta u_{lin}\, e^T\, P\, b_m \end{aligned} \Leftrightarrow \dot{V} = -e^T Q\, e < 0 \Rightarrow \dot{V}\left(e, \Delta k_x, \Delta k_r, \Delta k_u\right) \le 0$$

for open-loop unstable systems this is not sufficient for boundedness of all signals

- For open-loop unstable systems verifiable sufficient conditions established:
    - upper bound on $r_{\max}$
    - lower bound on μ
    - upper bounds on initial conditions of the system state $x(0)$ and the Lyapunov function $V(0)$

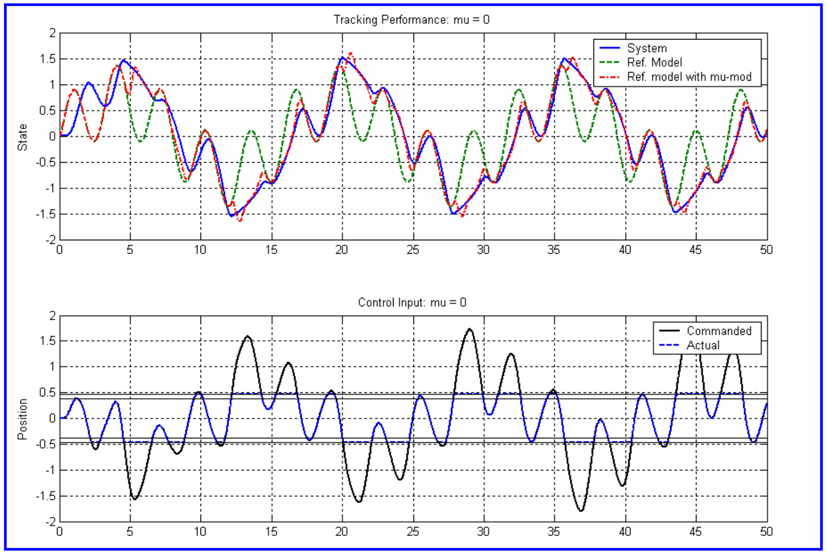
Simulation Example

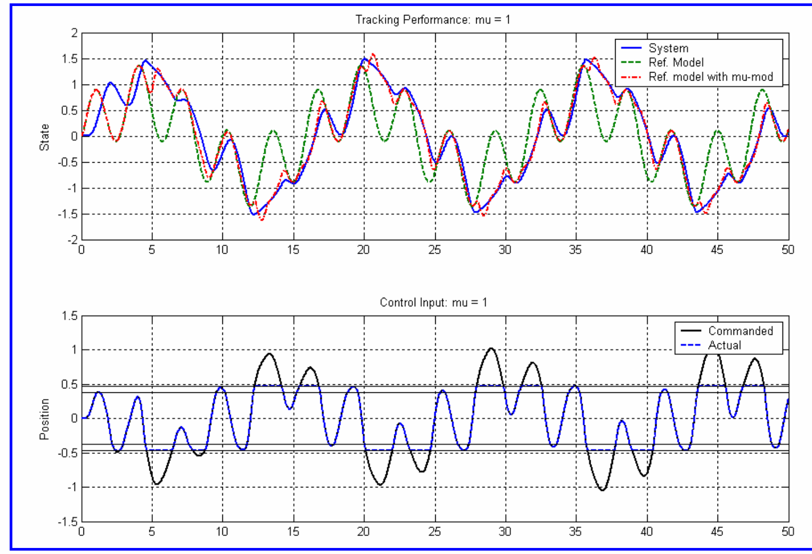
- Unstable open-loop system:

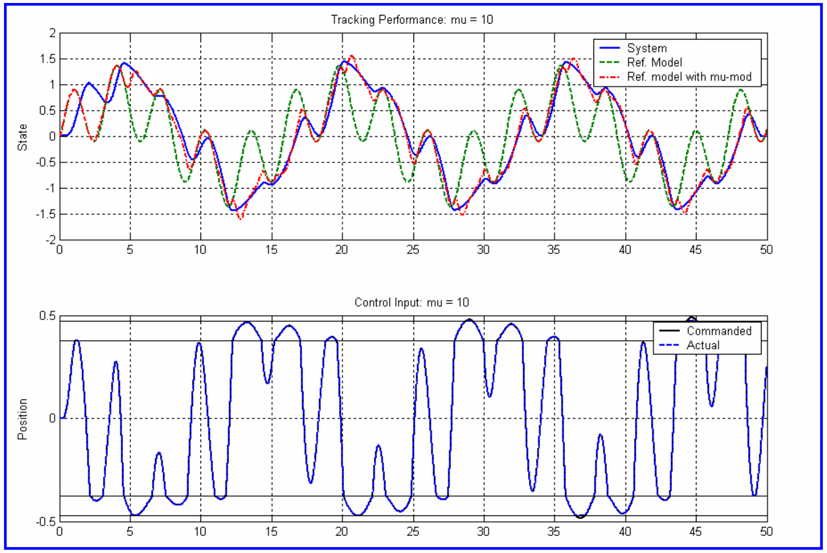
$$\dot{x} = a\,x + b\,u_{\max}\,\mathrm{sat}\left(\frac{u_c}{u_{\max}}\right),\ \text{where: } a = 0.5,\ b = 2,\ u_{\max} = 0.47$$

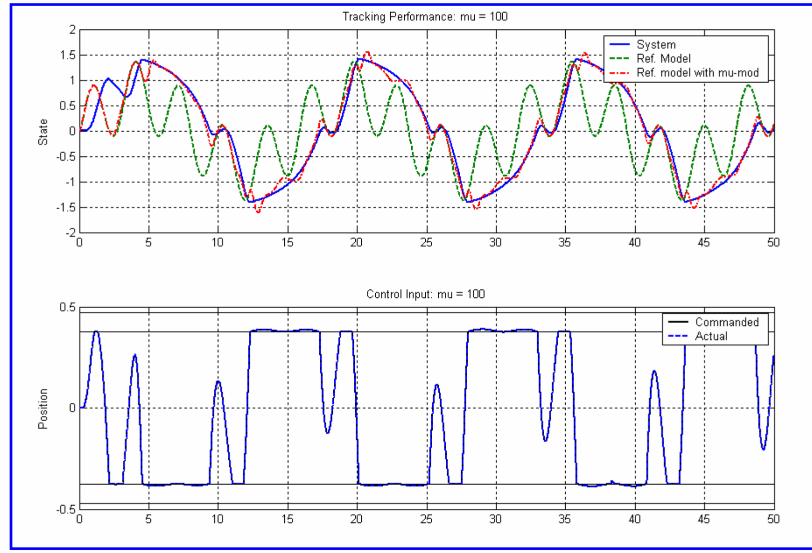
- Choose: $\delta = 0.2\,u_{\max}$ → $u_{\max}^{\delta} = u_{\max} - \delta = 0.8\,u_{\max}$
- Ideal reference model: $\dot{x}_m = -6\left(x_m - r(t)\right)$
- Reference input: $r(t) = 0.7\left(\sin(2t) + \sin(0.4t)\right)$
- Adaptation rates set to unity
- System and reference model start at zero

Simulation Data









Control Design Summary

- Choose "safety zone": $0 < \delta < u_{\max}$
- Define *virtual* control constraint: $u_{\max}^{\delta} = u_{\max} - \delta$
- Calculate *linear* component of adaptive control signal: $u_{lin} = k_x^T x + k_r r(t)$
- *Total* adaptive control with $\mu -$ mod:

$$u_c = \frac{1}{1+\mu}\left(u_{lin} + \mu\, u_{\max}^{\delta} \operatorname{sgn}\left(\frac{u_{lin}}{u_{\max}^{\delta}} \right) \right)$$

$$\begin{cases} \dot{k}_x = -\Gamma_x\, x\, e^T P\, b \\ \dot{k}_r = -\gamma_r\, r(t)\, e^T P\, b \\ \dot{k}_u = \gamma_u\, \Delta u_{lin}\, e^T\, P\, b_m \end{cases}$$

adaptive laws

- *Modified* reference model:

$$\dot{x}_m = A_m\, x_m + b_m \left(r + k_u \left(u_{\max} \operatorname{sat}\left(\frac{u_c}{u_{\max}} \right) - u_{lin} \right) \right)$$

Conclusions

- Design modification for stable adaptation in the presence of input constraints, (μ – mod)
  - Theoretically justified
  - Lyapunov based
  - Verifiable sufficient conditions derived
- Future Work
  - MIMO systems
  - Dynamic actuators
  - Nonaffine-in-control dynamics
  - Flight control applications

Adaptive Flight Control Applications, Open Problems, and Future Work

Autonomous Formation Flight, (AFF)

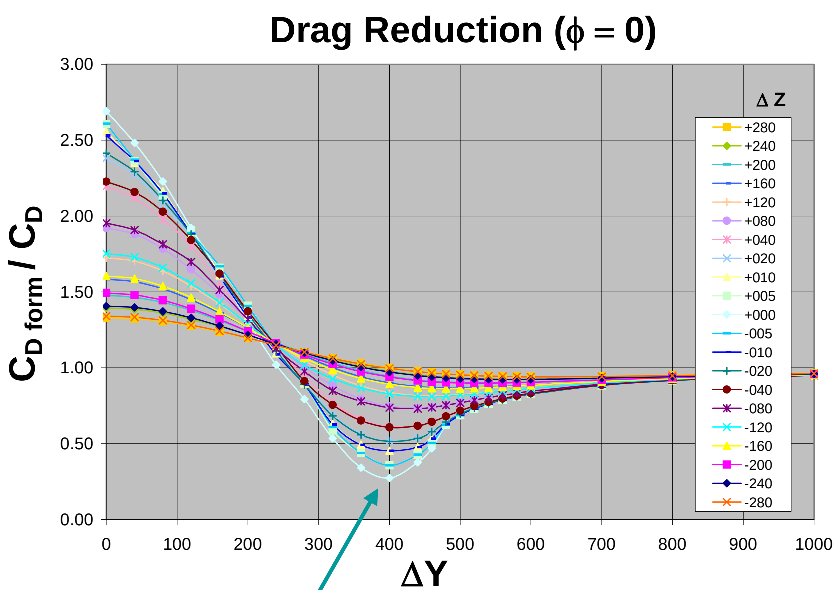
References:

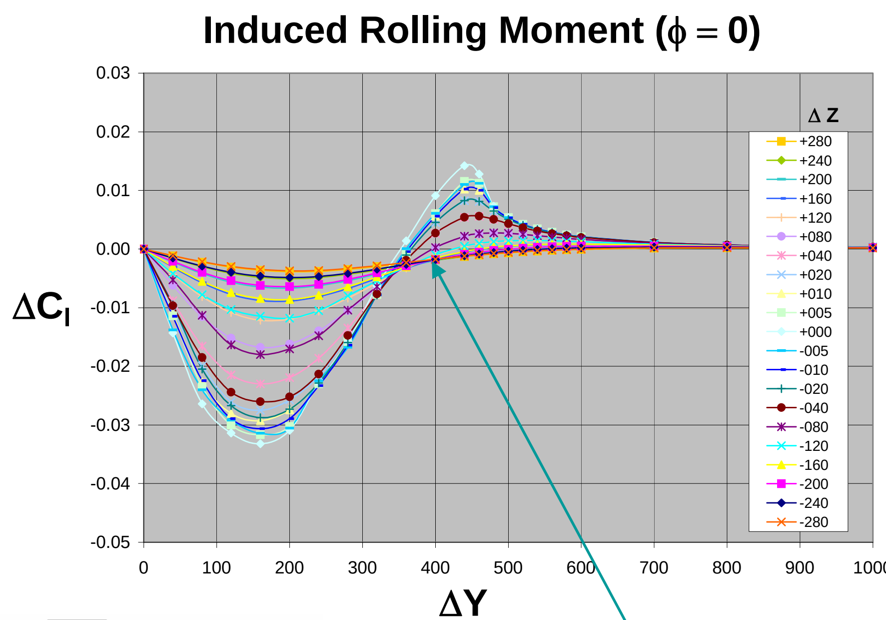
- Lavretsky, E. "F/A-18 Autonomous Formation Flight Control System Design", *AIAA GN&C Conference, Monterey, CA, 2002.*
- Lavretsky, E., Hovakimyan, N., Calise, A., Stepanyan, V. *"Adaptive Vortex Seeking Formation Flight Neurocontrol", AIAA-2002-4757*, *AIAA GN&C Conference,St. Antonio, TX, 2003.*

AFF: Program Overview

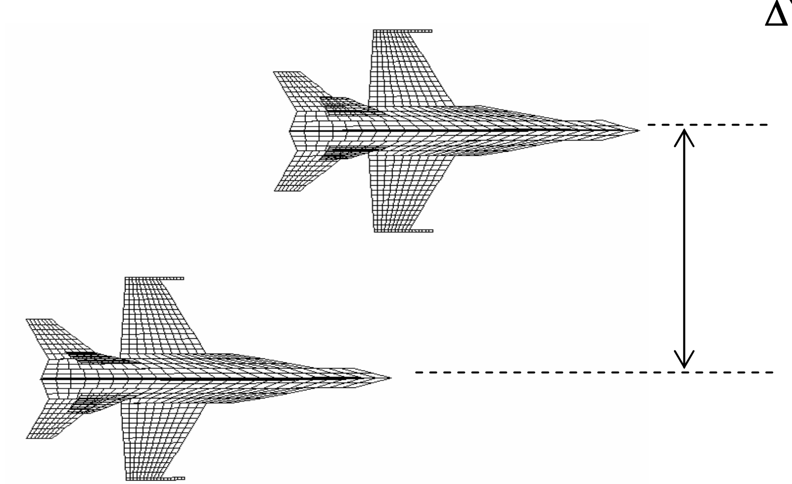
- **Program participants**:
  - NASA Dryden
  - Boeing - Phantom Works
  - UCLA
- **Flight test program**
  - Completed in December of 2001
  - 2 F/A-18 Hornets, 45 flights
  - Demonstrated up to 20% induced aerodynamic drag reduction
- **AFF Autopilot**
  - **Baseline** linear classical design to meet stability margins
  - **Adaptive** incremental system to counteract unknown vortex effects and environmental disturbances
  - **On-line extremum seeking** command generation

AFF: Lead Aircraft Wingtip Vortex Effects
Induced Drag Ratio & Rolling Moment Coefficient





"sweet" spot



close to vortex induced roll reversal

AFF: Trailing Aircraft Dynamics in Formation

- Trailing Aircraft:

Longitudinal Dynamics

$$\begin{cases} \dot{l} = V \\ \dot{V} = -g\sin\gamma + \dfrac{\rho V^2}{2m} S\left(C_{T_{\delta_T}} \delta_T - C_D(M,\alpha)\,\eta(y,\phi)\right) \end{cases}$$

Throttle Input

Lateral Dynamics

$$\begin{cases} \ddot{y} = -g\, n_z \sin\phi \\ \dot{\phi} = p \\ \dot{p} = -a_p\, p + b_{\delta_a} \delta_a + \xi(y,\phi) \end{cases}$$

Aileron Input

- Trailing Aircraft Modeling Assumptions
  - SCAS yields 1st order roll dynamics & turn coordination
  - $a_p, b_{\delta_a}, C_{T_{\delta_T}}$ are *unknown positive* constants
  - $C_D(M,\alpha),\ \eta(y,\phi),\ \xi(y,\phi)$ are *unknown bounded* functions of known arguments and shapes
- Lead aircraft trimmed for level flight

AFF: Vortex Seeking Formation Flight Control

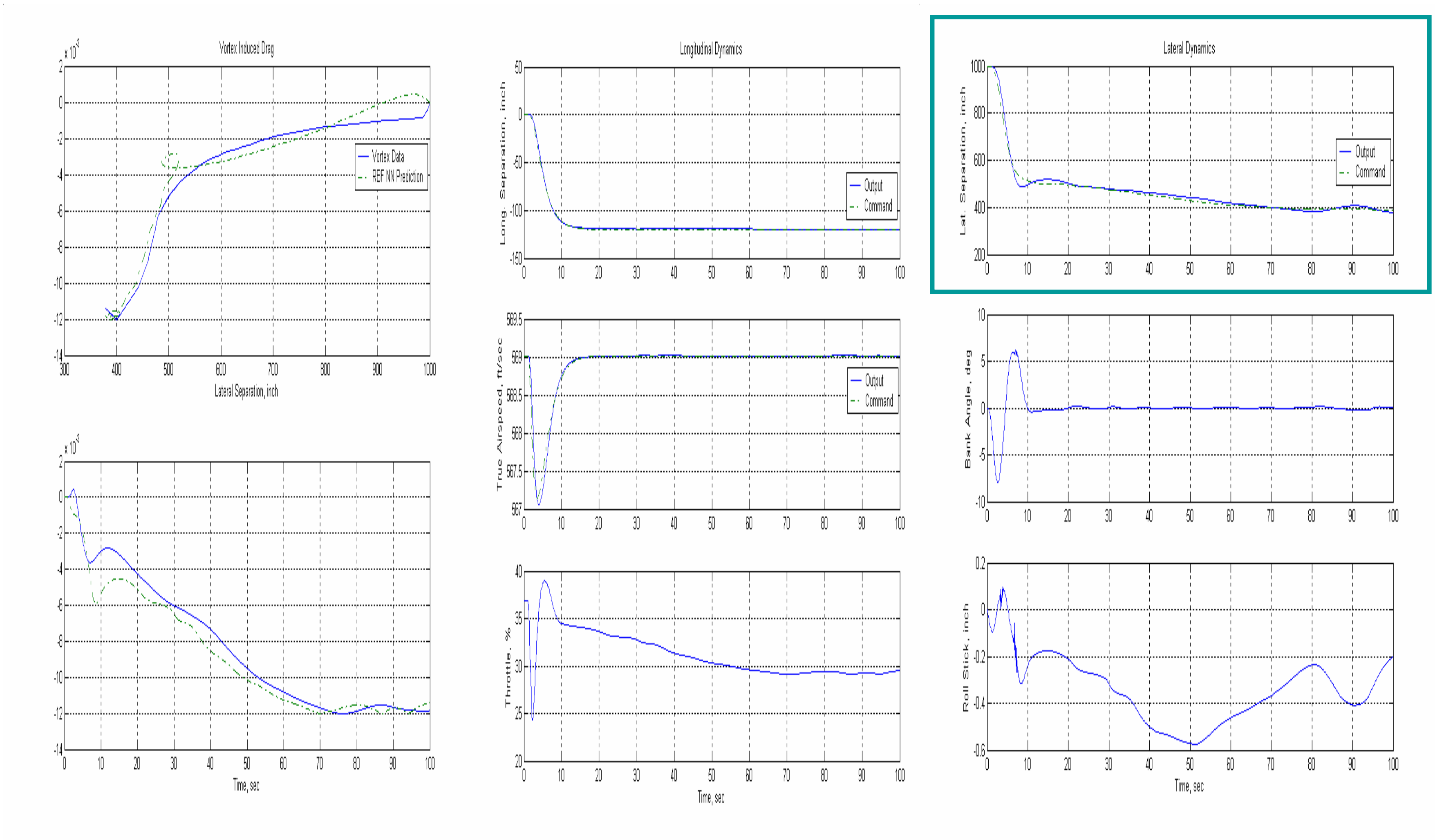
- **Problem**: Using *throttle* and *aileron* inputs
  - Track desired longitudinal displacement command l_c
  - Generate on-line and track lateral separation command y_c in order to:
    - Minimize unknown vortex induced drag coefficient $\eta(y,\phi)$ with respect to its 1st argument, (lateral separation)

$$\dot{V} = -g\sin\gamma + \frac{\rho V^2}{2m} S\left(C_{T_{\delta_T}} \delta_T - C_D(M,\alpha)\eta(y,\phi)\right)$$

- Remarks:
  - Aileron controls lateral separation
  - Throttle controls longitudinal separation
    - depends on lateral separation through unknown function $\eta(y,\phi)$
- **Solution**
  - Using Direct Adaptive Model Reference Control
  - Radial Basis Functions for approximation of uncertainties
  - Extremum Seeking Command Generation

$$\dot{y}_r = -\gamma \left.\frac{\partial \hat{\eta}(y,\phi)}{\partial y}\right|_{y=y_r}, \quad \gamma > 0$$

AFF: Simulation Data

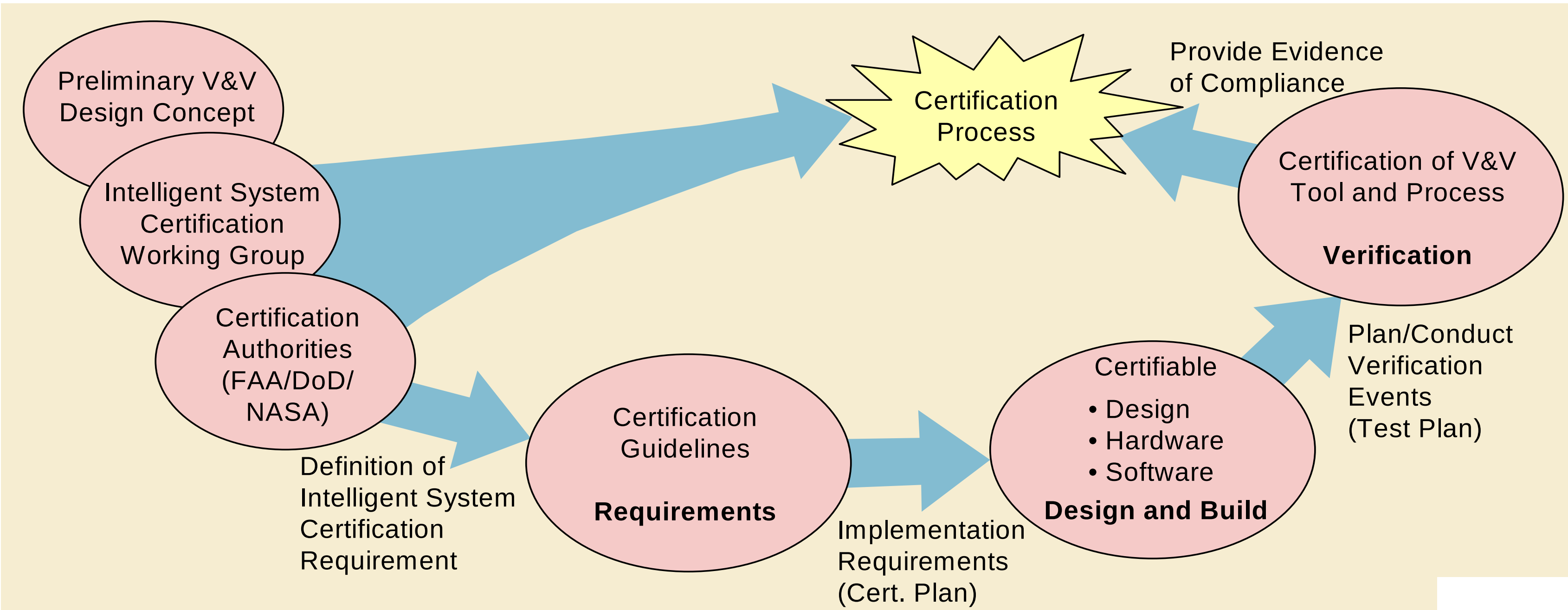


Open Problems and Future Work

Task 1: Validation & Verification (V&V) of Adaptive Systems

- Significant industry effort going into development of adaptive / reconfigurable GN&C systems
- Methods to test and certify flight critical systems are not readily available
- There exists a necessity to develop V&V methods and certification tools that are similar to and extend the current process for conventional, non-adaptive GN&C systems
- Theoretically justified V&V technologies are needed to:
  - provide a standard process against which adaptive GN&C systems can be certified
  - offer certification guidelines during the early design cycle of such systems

Task 1: V&V of Adaptive Systems
Road Map to Solution (Issue Paper)



- **Goal:** Provide *theoretically* justified V&V method and a process-based acceptance procedure to certify current and future intelligent / adaptive GN&C flight critical systems

- **Two Major Tasks**
  - Stability Margins / Robustness Analysis
  - S/W V&V Procedures

Task 1: V&V of Adaptive Systems

Subtask: Theoretical Stability / Robustness Analysis

- Establish adaptive control design guidelines
  - Define rates of adaptation
  - Calculate stability / robustness margins
  - Determine bounds on control parameters that correspond to stability / robustness margins
- Perform system validation using the derived margins
- Incorporate modifications that lead to improvement (if required) in the stability / robustness margins
- Validate closed-loop system tracking performance

Task 2: Integrated Vehicle Health Management (IVHM) and Composite Adaptation

- Aerodynamic parameters are of paramount importance to IVHM system functionality
- Examine different sources of on-line aerodynamic parameter estimation
  - Tracking errors
  - Prediction errors
- Composite Adaptive Flight Control = (Indirect + Direct) MRAC

Task 3: Persistency of Excitation in Flight Mechanics

- Information content from adaptation / estimation processes depends on parameter convergence
  - Requires persistent excitation (PE) of control inputs
- Need numerically stable / on-line verifiable PE conditions for flight mechanics and control
- *Aircraft Example*: Longitudinal dynamics

$$
\begin{cases}
\dot{V} = \dfrac{T\cos\alpha - D}{m} - g\sin(\theta - \alpha) \\
\dot{\alpha} = q - \dfrac{T\sin\alpha + L}{mV} + g\cos(\theta - \alpha) \\
\dot{q} = \dfrac{M}{I_y} \\
\dot{\theta} = q
\end{cases}
$$

$$
\begin{cases}
T = \bar{q}\, S\, C_T \cong \bar{q}\, S\, C_{T_{\delta_T}}\, \delta_T \\
L = \bar{q}\, S\, C_L \cong \bar{q}\, S\, C_L(\alpha, q) \\
D = \bar{q}\, S\, C_D \cong \bar{q}\, S\, C_D(\alpha, q) \\
M = \bar{q}\, S\, \bar{c}\, C_M \cong \bar{q}\, S\, \bar{c}\left(C_{M_0}(\alpha, q) + C_{M_{\delta_e}}(\alpha, \delta_e)\, \delta_e\right)
\end{cases}
$$

Control Inputs

Problem:

- ➢Estimate on-line unknown aerodynamic coefficients
- ➢Find sufficient conditions (PE) that yield convergence of the estimated parameters to their corresponding true (unknown) values