Adaptive Control: Introduction, Overview, and Applications

Eugene Lavretsky, Ph.D.

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

Kevin A. Wise, Ph.D.

E-mail: kevin.a.wise@boeing.com
Phone: 314-232-4549

Course Overview

- **Motivating Example**
- **Review of Lyapunov Stability Theory**
  - Nonlinear systems and equilibrium points
  - Linearization
  - Lyapunov's direct method
  - Barbalat Lemma, Lyapunov-like Lemma, Bounded Stability
- **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**
- **Open Problem: V&V of Adaptive Control Challenge**
- **Adaptive Autopilot for MK-82 Joint Direct Attack Munition (JDAM)**
  - **design, analysis and flight testing**

References

- J-J. E. Slotine and W. Li, *Applied Nonlinear Control*, Prentice-Hall, New Jersey, 1991
- P.A. Ioannou and J. Sun, *Robust Adaptive Control*, Prentice Hall, Upper Saddle River, NJ, 1996, electronic copy available in PDF form at: http://www.rcf.usc.edu/~ioannou/Robust_Adaptive_Control.htm
- S. Sastry and M. Bodson, Adaptive Control: Stability, Convergence, and Robustness, Prentice-Hall, 1989, electronic copy available in PDF form at: http://www.ece.utah.edu/~bodson/acscr
- K.S. Narendra and A.M. Annaswamy, Stable Adaptive Systems, Dover, 2005
- 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
- E. Lavretsky, “Adaptive Control,” Lecture Notes, Caltech, Spring 2005
- E. Lavretsky, “System Identification,” Lecture Notes, Caltech, Spring 2006

The *survival of the fittest* concept: "In the struggle for survival, the fittest win out at the expense of their rivals because they succeed in *adapting* themselves best to their environment”, Charles Robert Darwin (1809–1882), The Origin of Species, The Harvard Classics, Vol. 11, 1909–14.

Motivating Example: Roll Dynamics

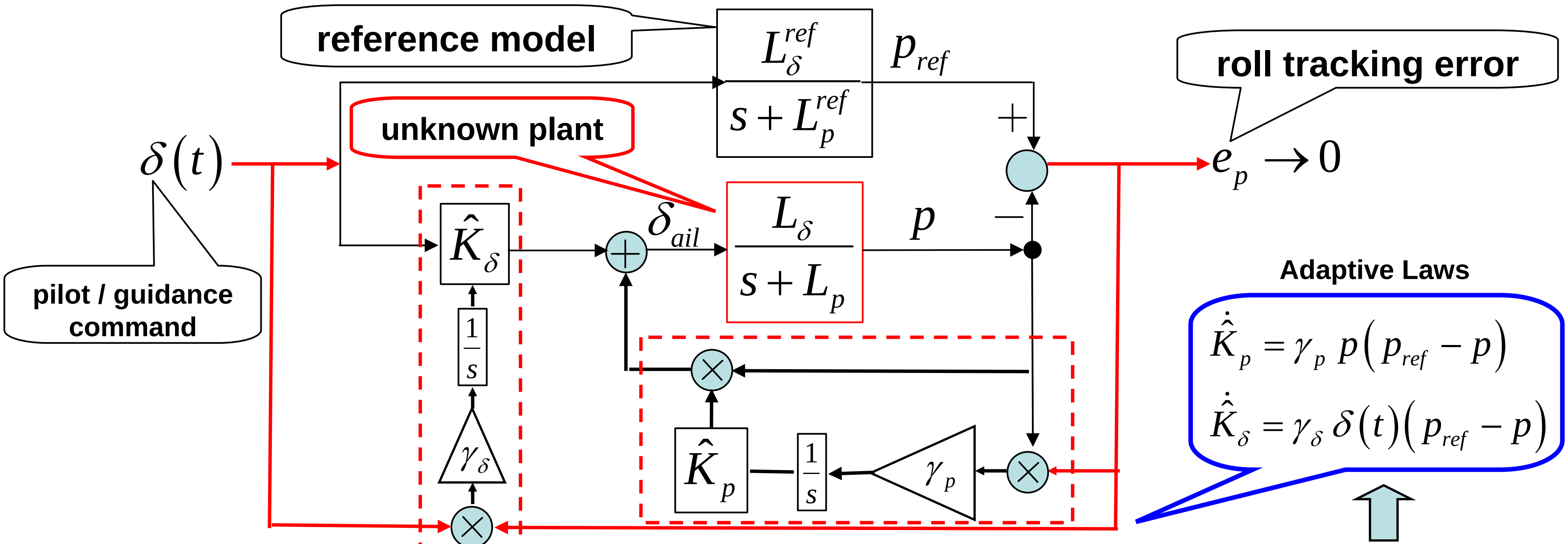
(Model Reference Adaptive Control)

- ***Uncertain*** **Roll dynamics**: $\dot{p} = L_p\, p + L_{\delta_{ail}}\, \delta_{ail}$
  – p is roll rate,
  – δ_{ail} is aileron position
  – $\left(L_p,\, L_{\delta_{ail}}\right)$ are *unknown* damping, aileron effectiveness
- **Flying Qualities Model**: $\dot{p}_{ref} = L_p^{ref}\, p_{ref} + L_\delta^{ref}\, \delta(t)$
  – $\left(L_p^{ref},\, L_\delta^{ref}\right)$ are *desired* damping, control effectiveness
  – $\delta(t)$ is a reference input, (pilot stick, guidance command)
  – roll rate tracking error: $e_p(t) = \left(p(t) - p_{ref}(t)\right) \to 0$
- **Adaptive Roll Control**: $\delta_{ail} = \hat{K}_p\, p + \hat{K}_\delta\, \delta$

$$\begin{cases} \dot{\hat{K}}_p = -\gamma_p\, p\left(p - p_{ref}\right) \\ \dot{\hat{K}}_\delta = -\gamma_{\delta_{ail}}\, \delta(t)\left(p - p_{ref}\right) \end{cases}, \quad \left(\gamma_p,\, \gamma_{\delta_{ail}}\right) > 0$$

parameter adaptation laws

Adaptive Roll Control



$\dot{p} = L_p \, p + L_{\delta_{ail}} \, \delta_{ail}$

$\delta_{ail} = \hat{K}_p \, p + \hat{K}_\delta \, \delta$

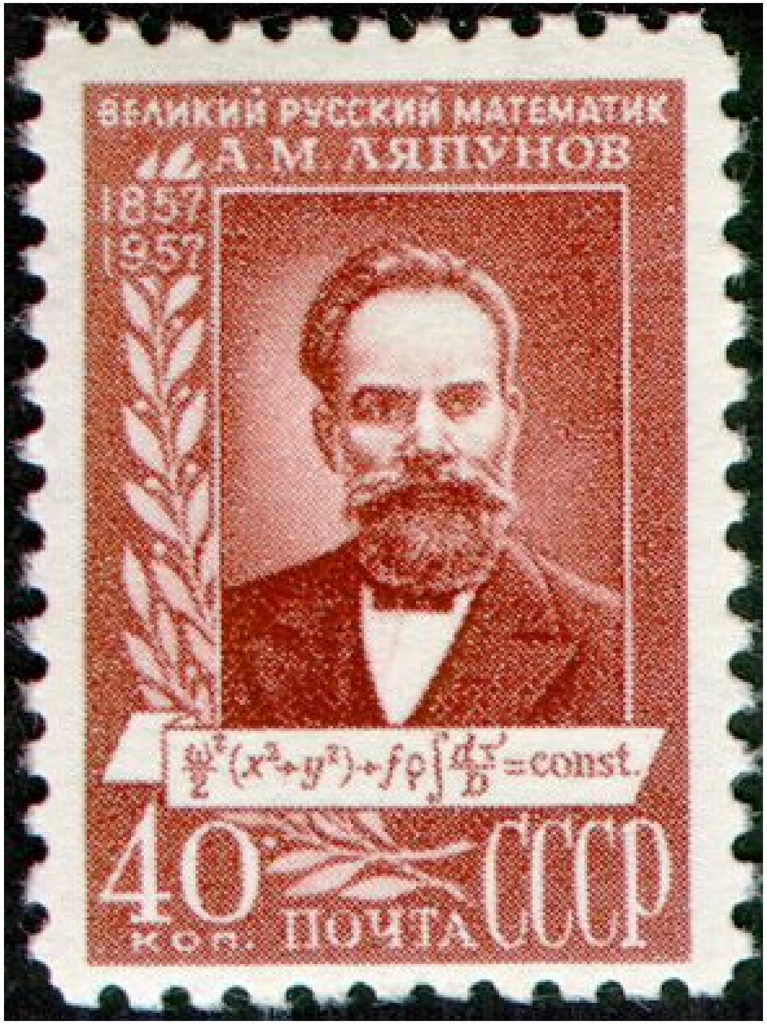
Nonlinear Integral Control !!!

- Design is based on Lyapunov Theorem (2nd method)
- Adaptive control provides stability, (in the sense of Lyapunov)
  - has infinite gain margin
  - time-delay sensitive, (phase margin unknown)
- Yields closed-loop asymptotic tracking with all remaining signals bounded in the presence of system uncertainties $\left(L_\delta, L_p\right)$

Lyapunov Stability Theory

Alexander Michailovich Lyapunov 1857-1918

- Russian mathematician and engineer who laid out foundations 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
- Most (if not all) of nonlinear control design methods are Lyapunov based



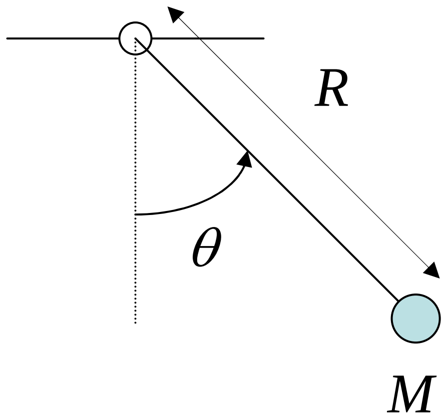
Nonlinear Dynamic Systems and Equilibrium Points

- A nonlinear dynamic system can usually be represented by a set of *n* differential equations in the form: $\boxed{\dot{x} = f(x,t), \quad \text{with } x \in R^n, t \in R}$
  - 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 ***autonomous***: $\boxed{\dot{x} = f(x)}$
- A state x_e is an equilibrium if once $x(t) = x_e$, it remains equal to x_e for all future times: $\boxed{0 = f(x)}$

Example: Equilibrium Points of a Pendulum

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

$$\boxed{\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= -\frac{b}{M\,R^2}\,x_2 - \frac{g}{R}\sin(x_1) \end{aligned}}$$

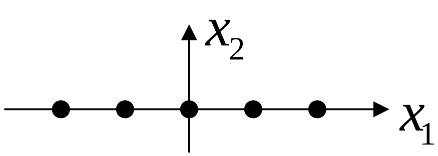


- Equilibrium points:

$$\boxed{\begin{aligned} 0 &= x_2 \\ 0 &= -\frac{b}{M\,R^2}\,x_2 - \frac{g}{R}\sin(x_1) \end{aligned}} \Rightarrow \boxed{x_2 = 0, \quad \sin(x_1) = 0}$$

$$\Downarrow$$

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



Example: Linear Time-Invariant (LTI) Systems

- LTI system dynamics: $\boxed{\dot{x} = A\,x}$
  - has a single equilibrium point (the origin 0) if *A* is nonsingular
  - has an infinity of equilibrium points in the null-space of *A*: $\boxed{A\,x_e = 0}$
- LTI system trajectories: $\boxed{x(t) = \exp\big(A(t - t_0)\big)\,x(t_0)}$
- 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 $\boxed{\dot{x} = f(x)}$
  - the nominal motion trajectory corresponding to initial conditions $x^*(0) = x_0$
- Perturb the initial condition $\boxed{x(0) = x_0 + \delta x_0}$
- Study the stability of the motion error: $\boxed{e(t) = x(t) - x^*(t)}$
- The error dynamics:
  - non-autonomous!

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

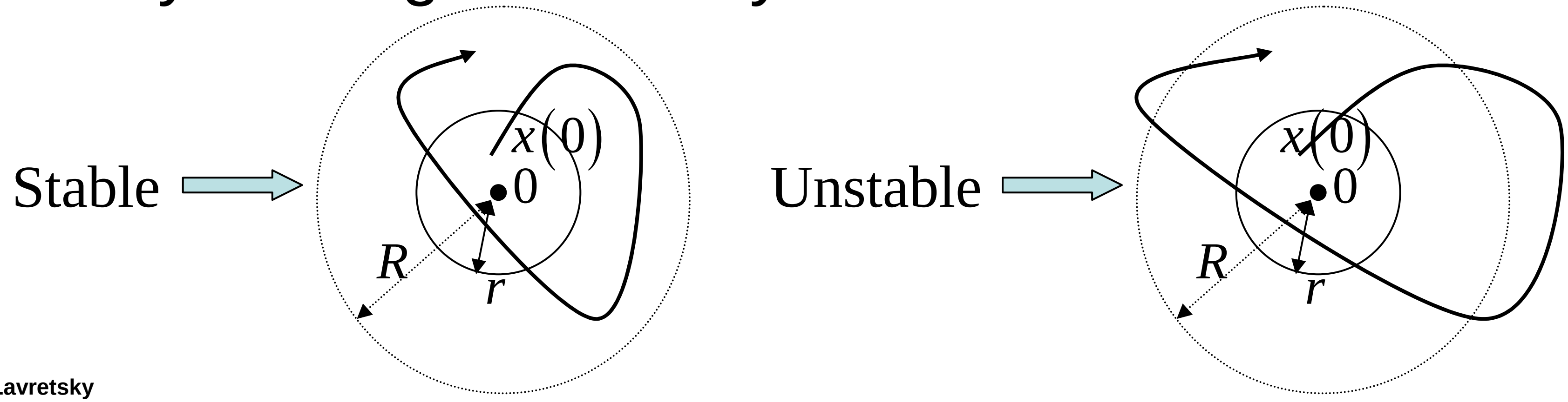
- <u>*Conclusion*</u>: Instead of studying stability of the nominal motion, study stability of the error dynamics w.r.t. the origin

Lyapunov Stability

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

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

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



Asymptotic Stability

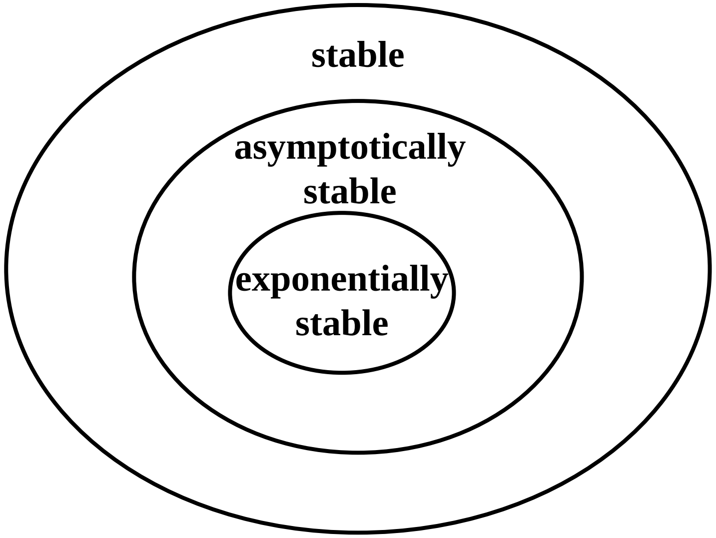
- **Definition**: An equilibrium point 0 is *asymptotically stable* if it is stable and if in addition: $$\boxed{\exists r > 0, \quad \left\{\left\|x(0)\right\| < r\right\} \Rightarrow \left\{\lim_{t\to\infty}\left\|x(t)\right\| = 0\right\}}$$
- Asymptotic stability means that the equilibrium is stable, 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:

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

- 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: $\boxed{\dot{x} = f(x)}$
- Assume that *f*(*x*) is continuously differentiable
- Perform linearization:

$$\boxed{\dot{x} = \underbrace{\left(\frac{\partial f(x)}{\partial x}\right)_{x=0}}_{A} x + \underbrace{f_{h.o.t.}(x)}_{\text{higher-order terms}} \cong A x}$$

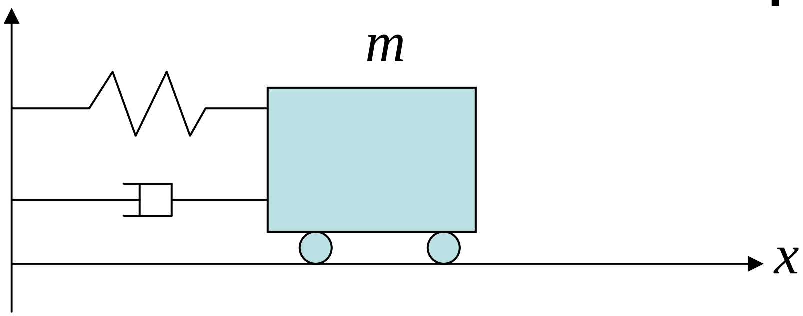
- **Theorem**
  - If A is Hurwitz then the equilibrium is asymptotically stable, (locally!)
  - If A has at least one eigenvalue in right-half complex plane then the equilibrium is unstable
  - If A has at least one eigenvalue on the imaginary axis then one cannot conclude anything from the linear approximation

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



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

- **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

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

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

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

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

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

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

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

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

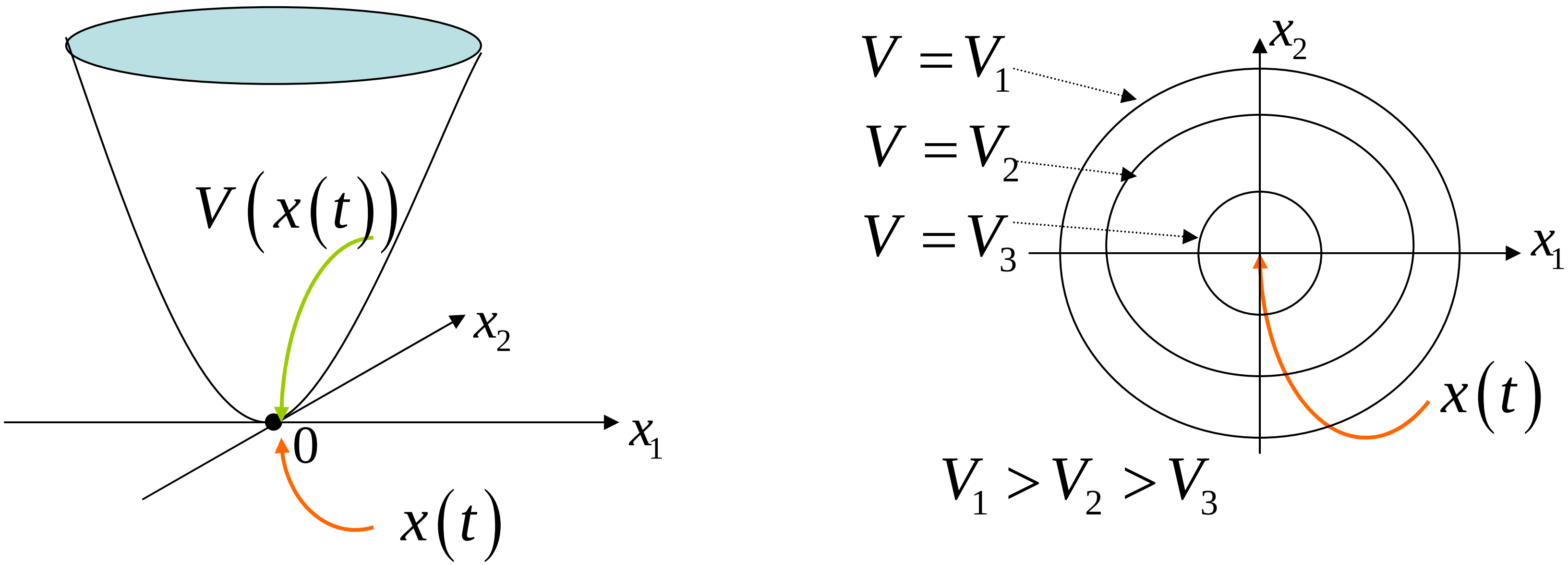
Lyapunov Functions

- **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 $\boxed{\dot{x} = f(x)}$ is negative semi-definite, i.e., $\boxed{\dot{V}(x) \le 0}$ then $V(x)$ is said to be a *Lyapunov function* for the system.

- *Time derivative* of the Lyapunov function

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

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 $\boxed{\forall x \in B_R : \quad V(x) > 0 \wedge \dot{V}(x) \leq 0}$ then the equilibrium point 0 is <u>*stable*</u>
  - If the time derivative is locally negative definite $\boxed{\dot{V}(x) < 0}$ then the stability is <u>*asymptotic*</u>
    - If $V(x)$ is radially unbounded, i.e., $\lim\limits_{\|x\| \to \infty} V(x) = \infty$, then the origin is <u>*globally asymptotically stable*</u>
- $V(x)$ is called the Lyapunov function of the system

Example: Local Stability

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

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

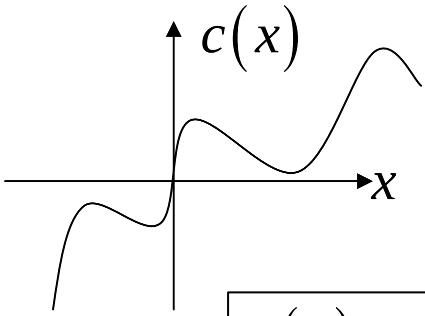
- <u>*Conclusion*</u>: System is locally stable

Example: Asymptotic Stability

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

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

Example: Global Asymptotic Stability



- Nonlinear 1st order system

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

- Lyapunov function candidate: $\boxed{V(x) = x^2}$
  - globally positive definite
  - time-derivative is negative definite

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

- *__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
  - If $\dot{V}\left(x\left(t\right)\right)\leq 0$ then all solutions of an *autonomous* system are bounded and approach *positive limit set*, (asymptotically stable equilibrium, limit cycle, others)
- Unlike Lyapunov Theorem, LaSalle's Theorem does not require $V\left(x\right)$ to be positive-definite.

Example: 2nd Order Nonlinear System

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

$$\boxed{\dot{x}\,b(\dot{x})=0\Leftrightarrow\dot{x}=0\underset{\text{using system dynamics}}{\Rightarrow}\ddot{x}+c(x)\Rightarrow c(x)=0\Leftrightarrow x_e=0}$$

    - 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: $\boxed{\dot{x} = A\,x}$
- Lyapunov function candidate: $\boxed{V(x) = x^T P\, x}$
  - where *P* is symmetric positive definite matrix
  - function V(x) is positive definite
- Time-derivative of *V*(*x*(*t*)) along the system trajectories: $\boxed{\dot{V}(x) = \dot{x}^T P\,x + x^T P\,\dot{x} = x^T \underbrace{\left(A^T P + P\,A\right)}_{-Q} x = -x^T Q\,x < 0}$
  - where Q is symmetric positive definite matrix
  - Lyapunov equation: $\boxed{A^T P + P\,A = -Q}$
- 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 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 Lemma extends Lyapunov stability analysis to *non-autonomous* systems, (such as adaptive model reference control)

- *Uniformly continuous functions*
  - Function $f(t): R \to R$ is uniformly continuous if:

$$\forall \varepsilon > 0 \quad \exists \delta = \delta(\varepsilon) > 0 \quad \forall |t_2 - t_1| \le \delta \Rightarrow |f(t_2) - f(t_1)| \le \varepsilon$$

Barbalat Lemma

- **Lemma (1st formulation)**
  - Suppose $f(t): R \to R$ is differentiable, has a finite limit, as $t \to \infty$, and $\dot{f}(t)$ is *uniformly continuous*. Then $\lim\limits_{t\to\infty} \dot{f}(t) = 0$.
- **Lemma (2nd formulation)**
  - If $f(t): R^+ \to R$ is uniformly continuous for all $t \geq 0$ and if the limit of the integral $\lim\limits_{t\to\infty} \int_0^t f(\tau)\,d\tau$ exists and finite, then $\lim\limits_{t\to\infty} f(t) = 0$.
- **Remarks**
  - Sufficient condition for uniform continuity
    - 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): $f(t) \to \lim \not\Leftrightarrow \dot{f}(t) \to \lim$
    - Counter-examples:

$$\boxed{f(t) = \sin(\ln t) \underset{t\to\infty}{\not\to} \lim \qquad \dot{f}(t) = \frac{\cos(\ln t)}{t} \underset{t\to\infty}{\to} 0}$$

$$\boxed{f(t) = e^{-t}\sin\left(e^{2t}\right) \underset{t\to\infty}{\to} 0 \qquad \dot{f}(t) = -e^{-t}\sin\left(e^{2t}\right) + e^{t}\sin\left(e^{2t}\right) \underset{t\to\infty}{\to} \infty}$$

Example: LTI System

- **Statement**: Output of a stable LTI system is uniformly continuous in time
  - System dynamics: $\boxed{\dot{x} = A\,x + B\,u}$
  - Control input *u* is bounded
  - System output: $\boxed{y = C\,x}$
- **Proof**: Since *u* is bounded and the system is stable then *x* is bounded. Consequently, the output time-derivative $\boxed{\dot{y} = C\,\dot{x} = C\left(A\,x + B\,u\right)}$ is bounded. Thus, (using Barbalat 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: $\boxed{\lim_{t \to \infty} \dot{V}(x,t) = 0}$
- **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 $\boxed{\dot{e} = -e + \theta\, w(t),\ \dot{\theta} = -e\, w(t)}$
  - 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: $\boxed{V(e,\theta) = e^2 + \theta^2}$
    - it is positive definite
    - its time-derivative is negative semi-definite $\boxed{\dot{V}(e,\theta) = 2e(-e + \theta\, w) + 2\theta(-e\, w) = -2e^2 \le 0}$
    - consequently, *e* and θ are bounded
    - since $\boxed{\ddot{V}(e,\theta) = -4e(-e + \theta\, w)}$ is bounded, $\boxed{\dot{V}(e,\theta)}$ is uniformly continuous
    - hence: $\boxed{\lim\limits_{t\to\infty}(-2e^2) = \lim\limits_{t\to\infty}\dot{V}(e,\theta) = 0 \Rightarrow \lim\limits_{t\to\infty} e(t)}$

Stability Analysis Methods: Summary

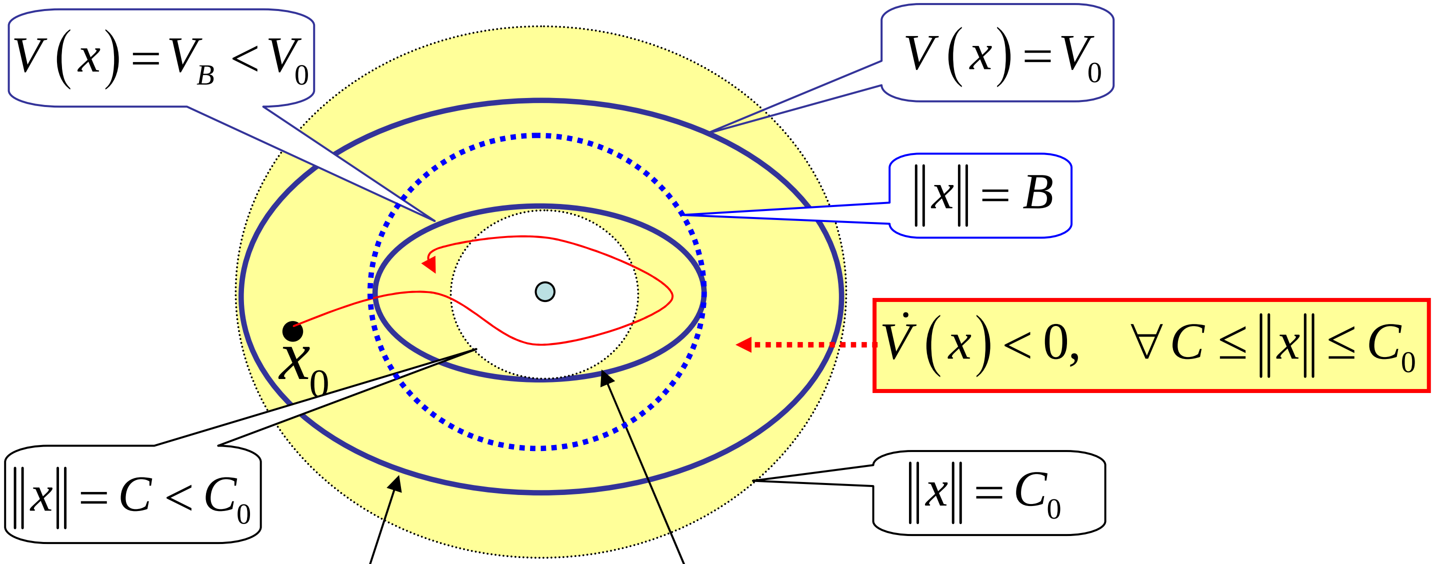
- **Lyapunov Direct Method, (2nd Theorem)**
  - Lyapunov positive-definite function V(x), (system energy)
  - time-derivative of V (system power) must be negative
    - semi-definite ➔ stable
    - strictly definite ➔ asymptotically stable
    - sign indefinite ➔ no conclusions
- **LaSalle Invariant Set Theorems**
  - applicable to autonomous systems
  - time-derivative of V is negative semi-definite
    - analysis of invariant sets ➔ asymptotic stability
- **Barbalat Lemma**
  - applicable to uniformly continuous (u.c.) functions
  - u.c. function has a limit ➔ its derivative tends to 0
- **Lyapunov-like Lemma**
  - applicable to non-autonomous systems, (such as tracking devices)
  - using Barbalat Lemma
  - boundedness of all state components and asymptotic stability of the system output signal

Uniform Ultimate Boundedness

- **Definition**: The solutions of $\dot{x} = f(x,t)$ starting at $x(t_0) = x_0$ are Uniformly Ultimately Bounded (UUB) with ultimate bound *B* if:

$$\boxed{\exists C_0 > 0,\ T = T(C_0, B) > 0: \quad \left(\|x(t_0)\| \le C_0\right) \Rightarrow \left(\|x(t)\| \le B, \quad \forall t \ge t_0 + T\right)}$$

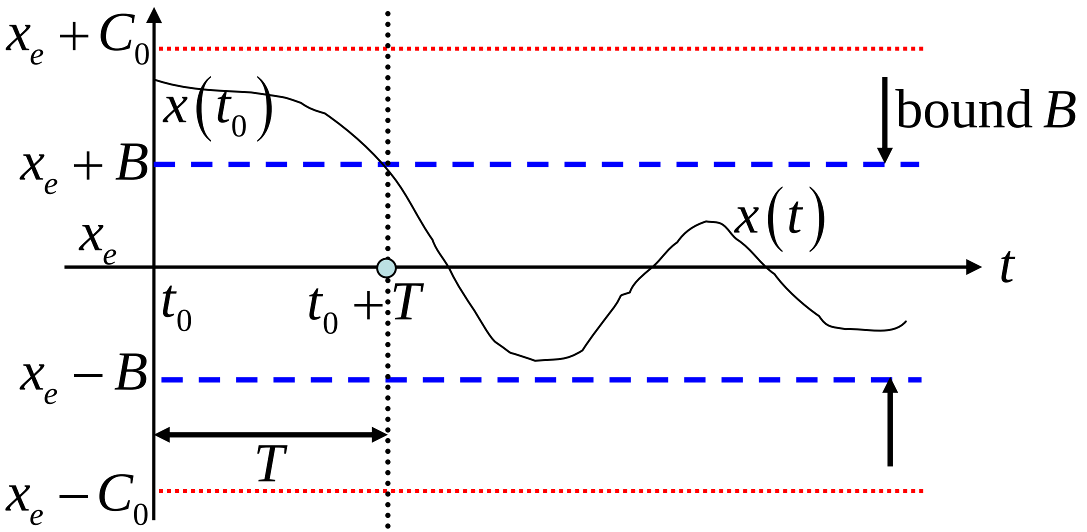
- Lyapunov analysis can be used to show UUB



All trajectories starting in large ellipse enter small ellipse within finite time $T(C_0, B)$

UUB Example : 1st Order System

- The equilibrium point x_e is UUB if there exists a constant C_0 such that for every initial state $x(t_0)$ in an interval $\boxed{|x(t_0)| \le C_0}$ there exists a bound B and a time $\boxed{T(B, x(t_0))}$ such that $\boxed{|x(t) - x_e| \le B}$ for all $\boxed{t \ge t_0 + T}$



UUB by Lyapunov Extension

- Milder form of stability than SISL
- More useful for controller design in practical systems with unknown bounded disturbances:

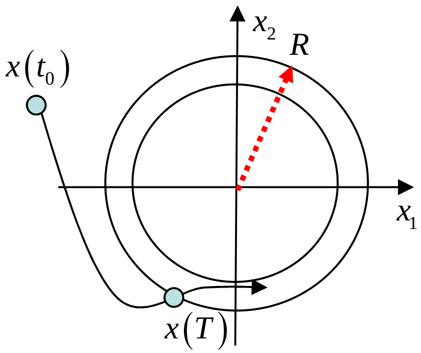
$$\boxed{\dot{x} = f(x) + d(x)}$$

- *Theorem*: Suppose that there exists a function *V*(*x*) with continuous partial derivatives such that for x in a compact set $S \subset R^n$
  - *V*(*x*) is positive definite: $V(x) > 0, \quad \forall \|x\| \neq 0$
  - time derivative of *V*(*x*) is negative definite outside of *S:* $\dot{V}(x) < 0, \quad \forall \|x\| > R, \quad (\|x\| \leq R) \Rightarrow (x \in S)$
  - Then the system is UUB and $\boxed{\|x(t)\| \leq R, \quad \forall t \geq t_0 + T}$

Example: UUB by Lyapunov Extension

- System:

$$\dot{x}_1 = x_1\, x_2^2 - x_1\left(x_1^2 + x_2^2 - 9\right)$$
$$\dot{x}_2 = -x_1^2\, x_2 - x_2\left(x_1^2 + x_2^2 - 9\right)$$



- Lyapunov function candidate:

$$V\left(x_1, x_2\right) = x_1^2 + x_2^2$$

- Time derivative:

$$\dot{V}\left(x_1, x_2\right) = 2\left(x_1\,\dot{x}_1 + x_2\,\dot{x}_2\right) = -2\left(x_1^2 + x_2^2\right)\left(x_1^2 + x_2^2 - 9\right)$$

- Time derivative negative outside of compact set

$$\dot{V}\left(x_1, x_2\right) < 0, \quad \forall\left\{x : x_1^2 + x_2^2 > 9\right\}$$

- ***Conclusion***: All trajectories enter circle of radius $R = 3 + \varepsilon$, in finite time T, (where $\varepsilon > 0$)

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

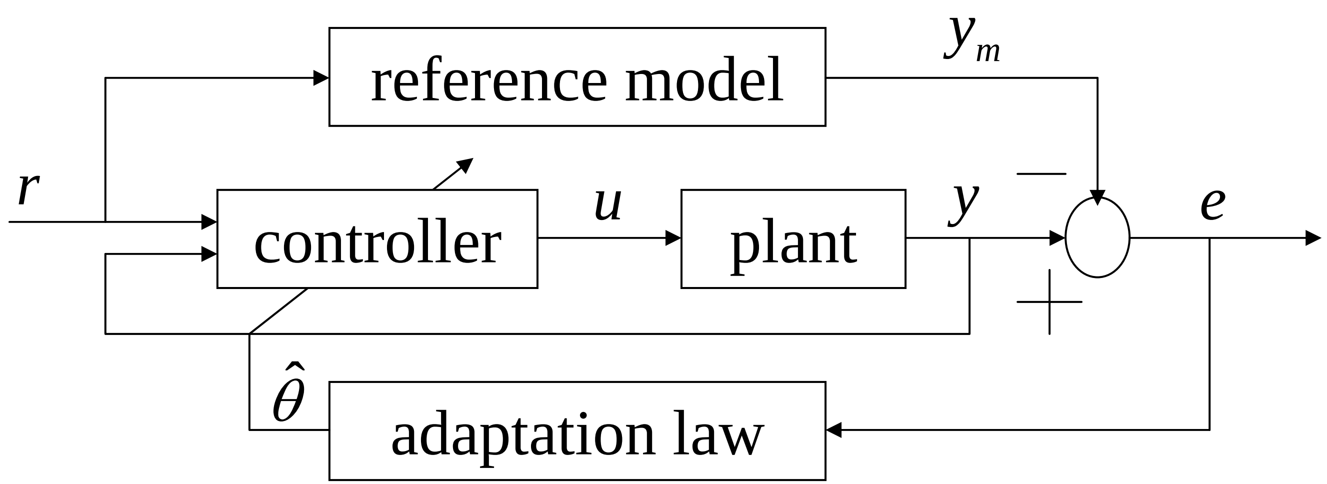
Historical Perspective

- 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

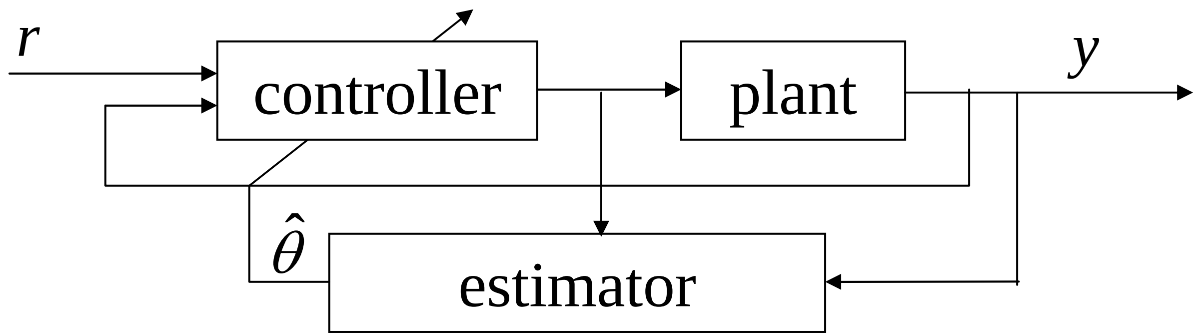
- **Why Adaptive Control?**
  - dealing with complex systems that have unpredictable parameter deviations and uncertainties
- **Basic Objective**
  - maintain consistent performance of a system in the presence of uncertainty and variations in plant parameters
- Adaptive control is superior to robust 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)



- <u>*Plant*</u> has a known structure but the parameters are unknown
- <u>*Reference model*</u> specifies the ideal (desired) response y_m to the external command r
- <u>*Controller*</u> is parameterized and provides tracking
- <u>*Adaptation*</u> 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: $\boxed{\dot{x} = a\,x + b\left(u + f\left(x\right)\right)}$
  - a, b are constant *unknown* parameters
  - *uncertain* nonlinear function: $\boxed{f\left(x\right) = \sum_{i=1}^{N} \theta_i\,\varphi_i\left(x\right) = \theta^T\,\Phi\left(x\right)}$
    - vector of constant *unknown* parameters: $\theta = \left(\theta_1 \quad \ldots \quad \theta_N\right)^T$
    - vector of known basis functions: $\Phi\left(x\right) = \left(\varphi_1\left(x\right) \quad \ldots \quad \varphi_N\left(x\right)\right)^T$
- Stable Reference Model: $\boxed{\dot{x}_m = a_m\,x_m + b_m\,r, \quad \left(a_m < 0\right)}$
- **Control Goal**
  - find u such that: $\boxed{\lim_{t\to\infty}\left(x\left(t\right) - x_m\left(t\right)\right) = 0}$

MRAC Design of 1st Order Systems (continued)

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

$$\boxed{\begin{array}{l} a + b\,\hat{k}_x - a_m = a + b\,\hat{k}_x - a - b\,k_x = b\left(\hat{k}_x - k_x\right) = b\,\Delta k_x \\ b\,\hat{k}_r - b_m = b\,\hat{k}_r - b\,k_r = b\left(\hat{k}_r - k_r\right) = b\,\Delta k_r \end{array}}$$

MRAC Design of 1st Order Systems (continued)

- Tracking Error: $\boxed{e(t) = x(t) - x_m(t)}$
- Error Dynamics (using model matching conditions:

$$\boxed{\begin{aligned} a + b\,\hat{k}_x - a_m &= b\,\Delta k_x \\ b\,\hat{k}_r - b_m &= b\,\Delta k_r \end{aligned}}$$

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

- Lyapunov Function Candidate:

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

  – where: $\gamma_x > 0, \gamma_r > 0, \text{ and } \Gamma = \Gamma^T > 0$ are rates of adaptations

MRAC Design of 1st Order Systems (continued)

- Time-derivative of the Lyapunov function

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

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

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

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

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

MRAC Design of 1st Order Systems (continued)

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

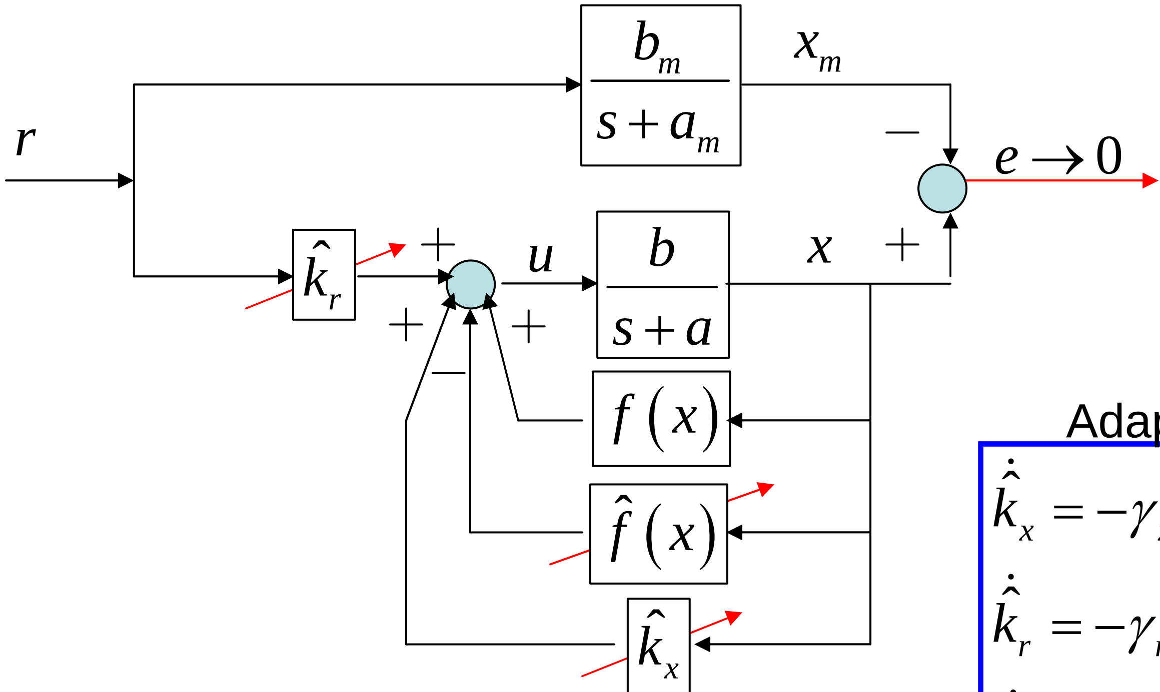
MRAC Design of 1st Order Systems (continued)

- 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: $\boxed{\lim\limits_{t\to\infty} \dot{V}(x,t) = 0}$
- Since $\boxed{\dot{V} = 2\,a_m\,e(t)^2}$ it follows that: $\boxed{\lim\limits_{t\to\infty} e(t) = 0}$
- **Conclusions**
  – achieved asymptotic tracking of any bounded time varying command: $\boxed{x(t) \to x_m(t),\ \text{as } t \to \infty}$
  – all signals in the closed-loop system are uniformly bounded in time

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



Adaptive Laws

$$\dot{\hat{k}}_x = -\gamma_x \, x \, e \, \mathrm{sgn}(b)$$

$$\dot{\hat{k}}_r = -\gamma_r \, r \, e \, \mathrm{sgn}(b)$$

$$\dot{\hat{\theta}} = \Gamma_\theta \, \Phi(x) \, e \, \mathrm{sgn}(b)$$

- Total control: $u = \hat{k}_x \, x + \hat{k}_r \, r - \hat{\theta}^T \, \Phi(x)$
- 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: $\boxed{\dot{x} = a\,x + b\,u + f(x)}$
  - a, b are constant ***unknown*** parameters
  - ***uncertain*** nonlinear function: $\boxed{f(x) = \sum_{i=1}^{N} \theta_i\, \varphi_i(x) = \theta^T\, \Phi(x)}$
    - 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: $\boxed{\dot{x}_m = a_m\, x_m + b_m\, r, \quad (a_m < 0)}$
- **Control Goal**
  - find u such that: $\boxed{\lim_{t\to\infty} \left(x(t) - x_m(t) \right) = 0}$

ADI Design of 1st Order Systems (continued)

- Rewrite system dynamics:

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

- Function estimation error:

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

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

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

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

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

- Adaptive laws

$$
\begin{aligned}
\dot{\hat a}&=\gamma_a\,x\,e\\
\dot{\hat b}&=\gamma_b\,u\,e\\
\dot{\hat\theta}&=\Gamma_\theta\,\Phi\left(x\right)e
\end{aligned}
\quad\Rightarrow\quad
\dot V\left(e,\Delta a,\Delta b,\Delta\theta\right)=2\,a_m\,e^2\le 0
$$

System energy decreases

ADI Design of 1st Order Systems (stability analysis)

- Similar to MRAC
- Using Barbalat's Lemma and Lyapunov-like Lemma: $\boxed{\lim\limits_{t\to\infty} \dot{V}(x,t) = \lim\limits_{t\to\infty}\left[2\,a_m\,e(t)^2\right] = 0}$
- Consequently: $\boxed{\lim\limits_{t\to\infty} e(t) = 0} \Rightarrow \boxed{x(t) \to x_m(t), \text{ as } t \to \infty}$
- **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

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

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

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

Persistency of Excitation (PE) (completed)

- If *r*(*t*) is such that $\boxed{v = \begin{pmatrix} x & r & \Phi^T(x) \end{pmatrix}^T}$ satisfies the so-called "*persistent excitation*" conditions, then the adaptive parameters converge to their true unknown values
  - PE Condition: $\boxed{\exists \alpha > 0 \quad \forall t \quad \exists T > 0 \quad \int_t^{t+T} v(\tau) v^T(\tau)\, d\tau > \alpha\, I_{N+2}}$
- PE Condition enables convergence of the parameter errors 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:

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

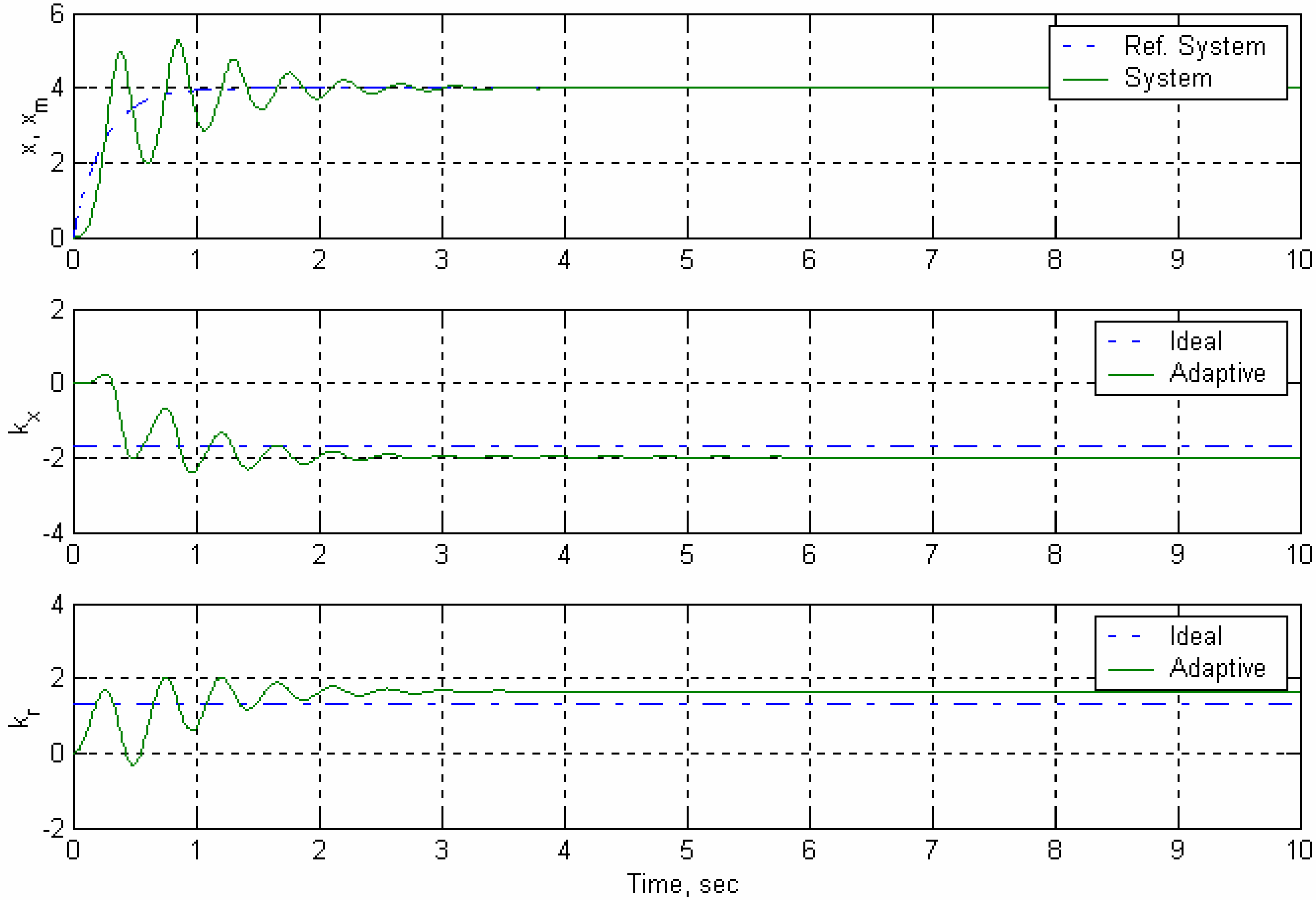
$$\dot{\hat{k}}_r = -2\,r\,e, \quad \hat{k}_r(0) = 0$$

- Two Reference Inputs:

$$r(t) = 4$$

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

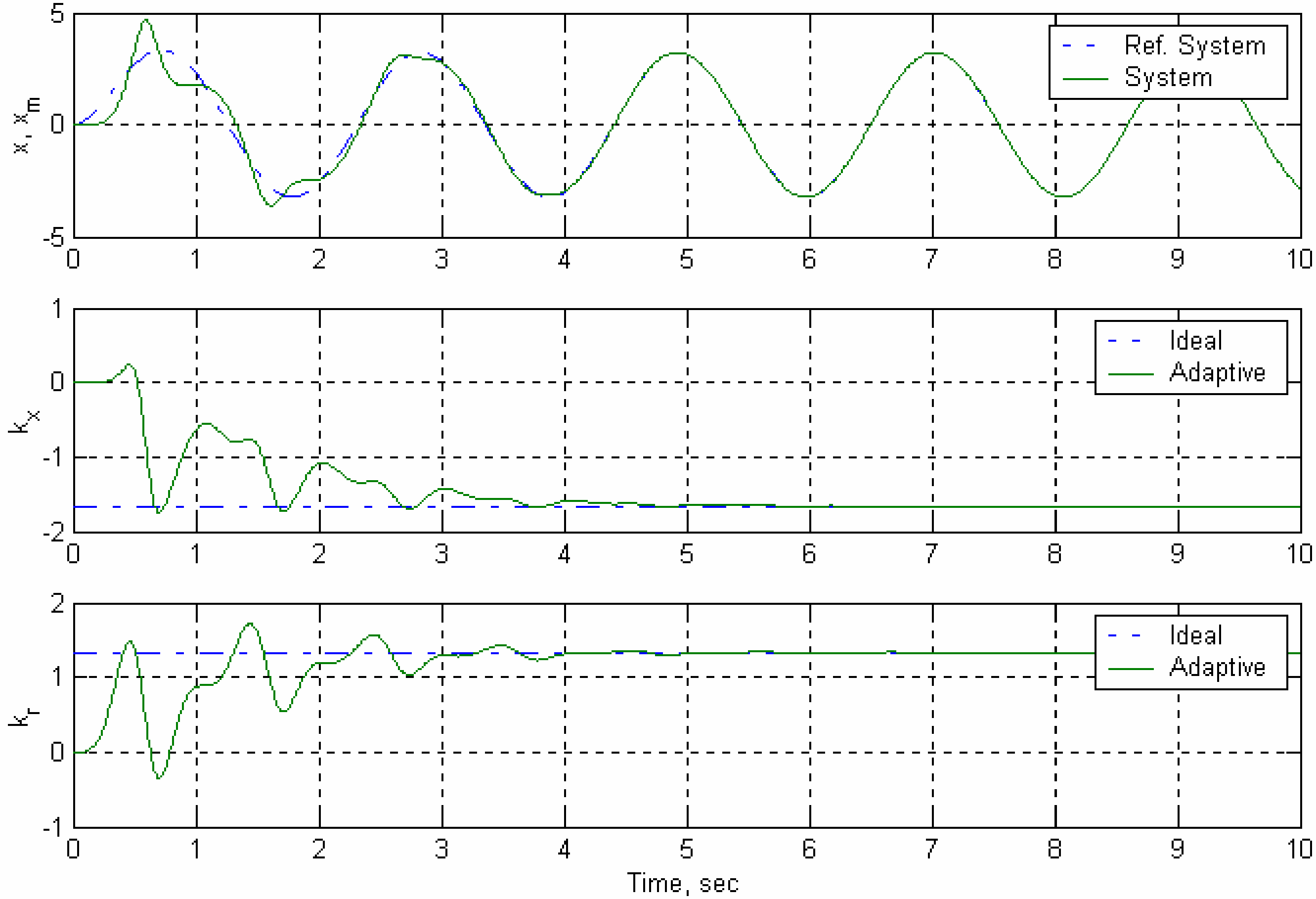
1st-Order *Linear* System MRAC Simulation w/o PE: $r(t) = 4$



Tracking Error Converges to Zero

Parameter Errors don't Converge to Zero

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

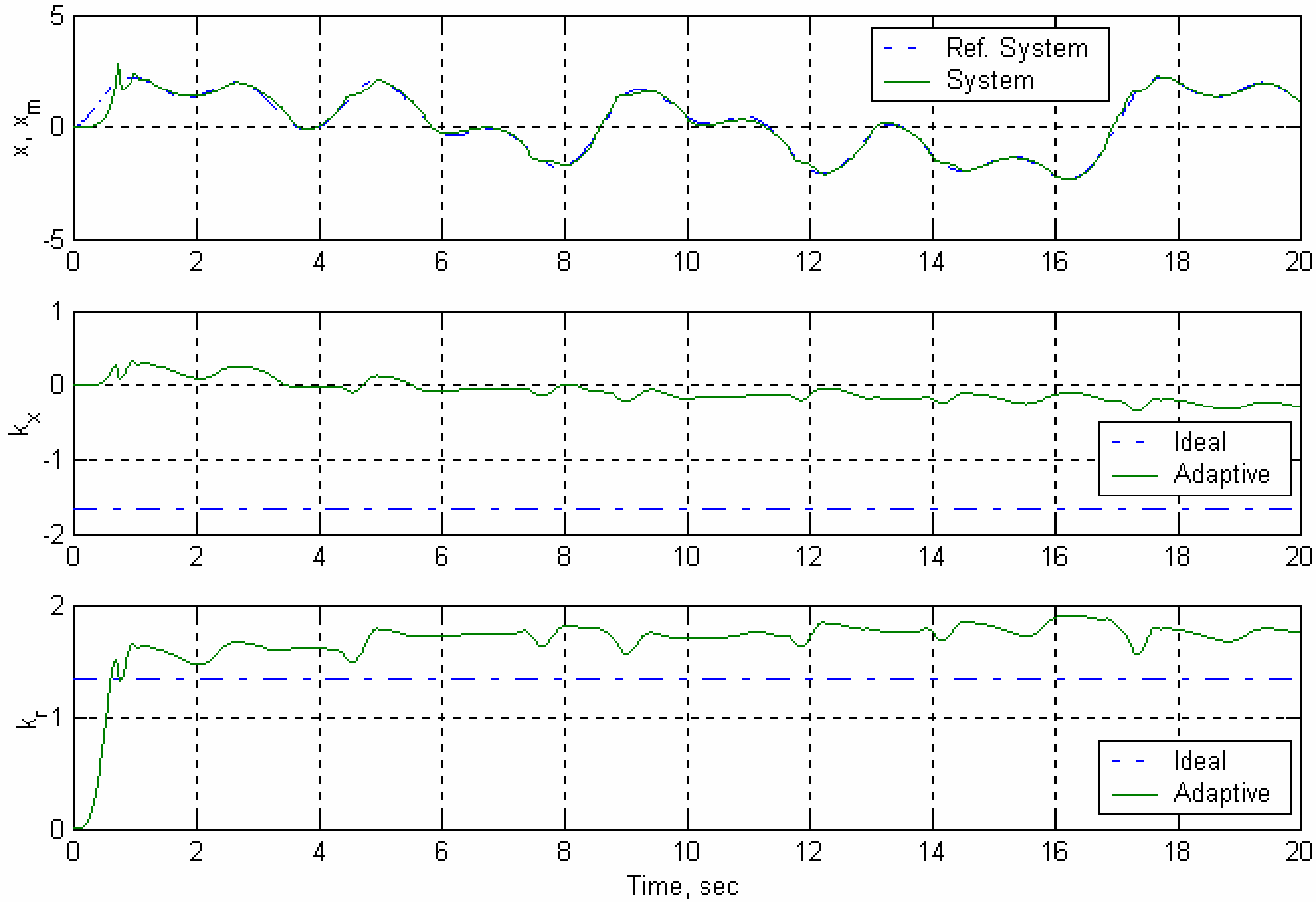


Tracking and Parameter Errors Converge to Zero

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

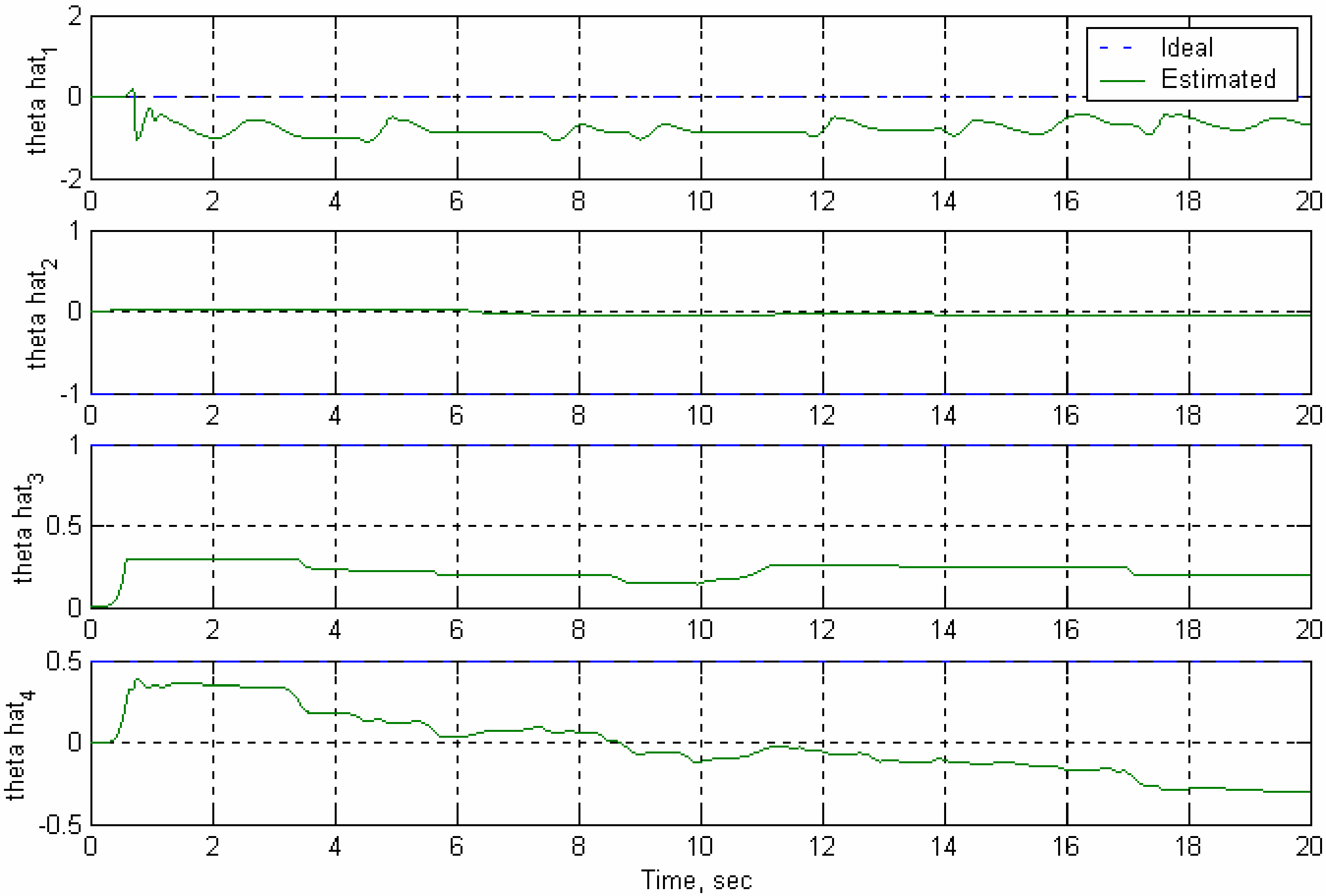
- Unstable Dynamics: $\boxed{\dot{x} = x + 3\left(u + f(x)\right), \quad x(0) = 0}$
  - plant parameters $a = 1, \quad b = 3$ are *unknown*
  - nonlinearity: $f(x) = -\theta^T \Phi(x)$
    - *known* basis functions: $\Phi(x) = \begin{pmatrix} x^3 & e^{-(x+0.5)^2 10} & e^{-(x-0.5)^2 10} & \sin(2x) \end{pmatrix}^T$
    - *unknown* parameters: $\theta = \begin{pmatrix} 0.01 & -1 & 1 & 0.5 \end{pmatrix}^T$
- Reference Model: $\boxed{\dot{x}_m = -4x_m + 4r(t), \quad x_m(0) = 0}$
- Adaptive Control: $\boxed{u = \hat{k}_x x + \hat{k}_r r - \hat{\theta}^T \Phi(x)}$
- Parameter Adaptation: $\rightarrow$ $$\boxed{\begin{aligned} \dot{\hat{k}}_x &= -2xe, \quad \hat{k}_x(0) = 0 \\ \dot{\hat{k}}_r &= -2re, \quad \hat{k}_r(0) = 0 \\ \dot{\hat{\theta}} &= 2\Phi(x)e, \quad \hat{\theta}(0) = 0_4 \end{aligned}}$$
- Reference Input: $\boxed{r(t) = \sin(3t) + \sin\left(\frac{3t}{2}\right) + \sin\left(\frac{3t}{4}\right) + \sin\left(\frac{3t}{8}\right)}$

1st-Order *Nonlinear* System
MRAC Simulation



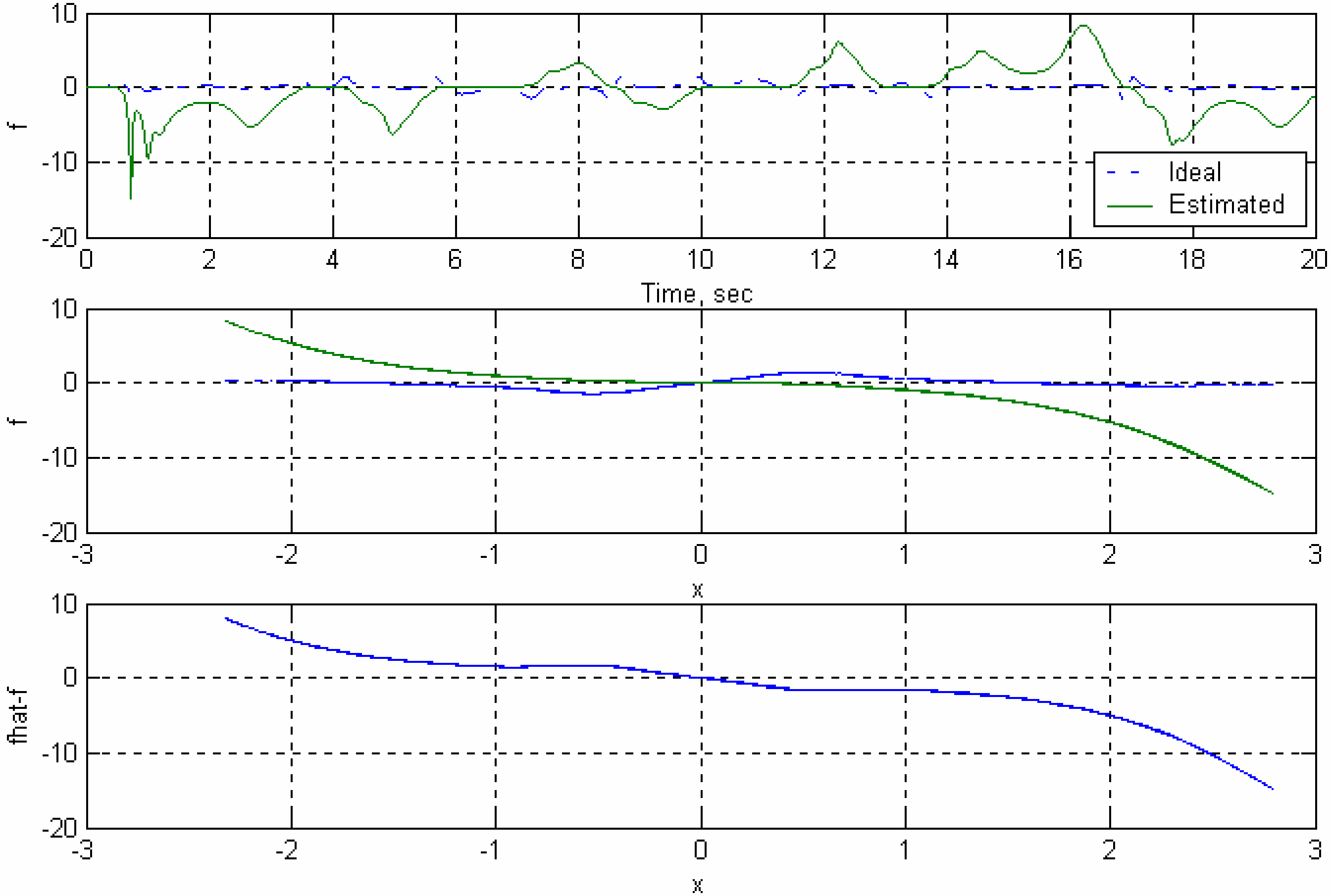
Good Tracking & Poor Parameter Estimation

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



Nonlinearity: Poor Parameter Estimation

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



Nonlinearity: Poor Estimation

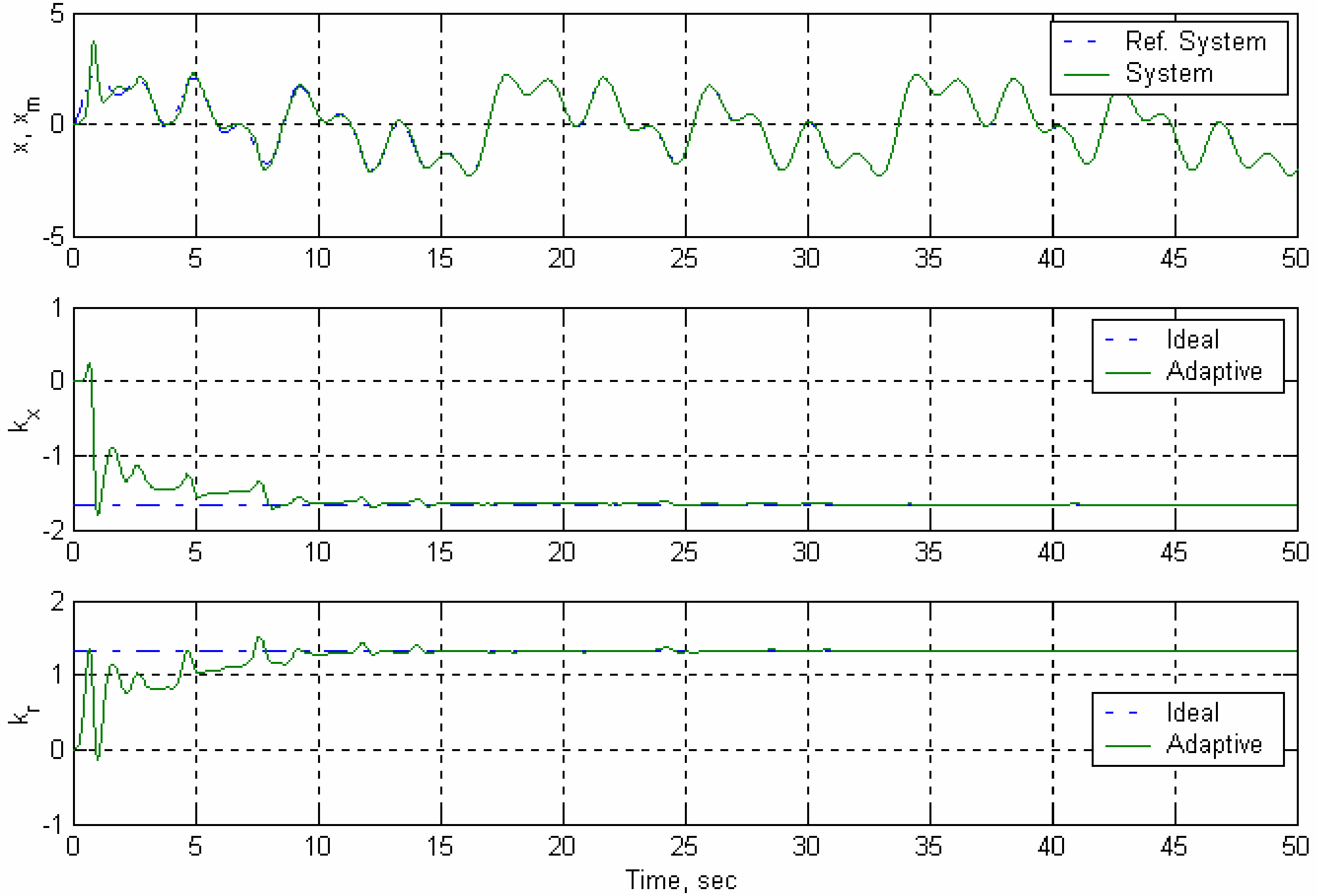
Example: MRAC of a 1st-Order *Nonlinear* System with *Local* Nonlinearity

- Unstable Dynamics: $\boxed{\dot{x} = x + 3\left(u + f(x)\right), \quad x(0) = 0}$
  - 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: $\boxed{\theta = \begin{pmatrix} 0 & -1 & 1 & 0 \end{pmatrix}^T}$
- Reference Model: $\boxed{\dot{x}_m = -4x_m + 4r(t), \quad x_m(0) = 0}$
- Adaptive Control: $\boxed{u = \hat{k}_x x + \hat{k}_r r - \hat{\theta}^T \Phi(x)}$
- Parameter Adaptation:

$$\boxed{\begin{aligned} \dot{\hat{k}}_x &= -2xe, \quad \hat{k}_x(0) = 0 \\ \dot{\hat{k}}_r &= -2re, \quad \hat{k}_r(0) = 0 \\ \dot{\hat{\theta}} &= 2\Phi(x)e, \quad \hat{\theta}(0) = 0_4 \end{aligned}}$$

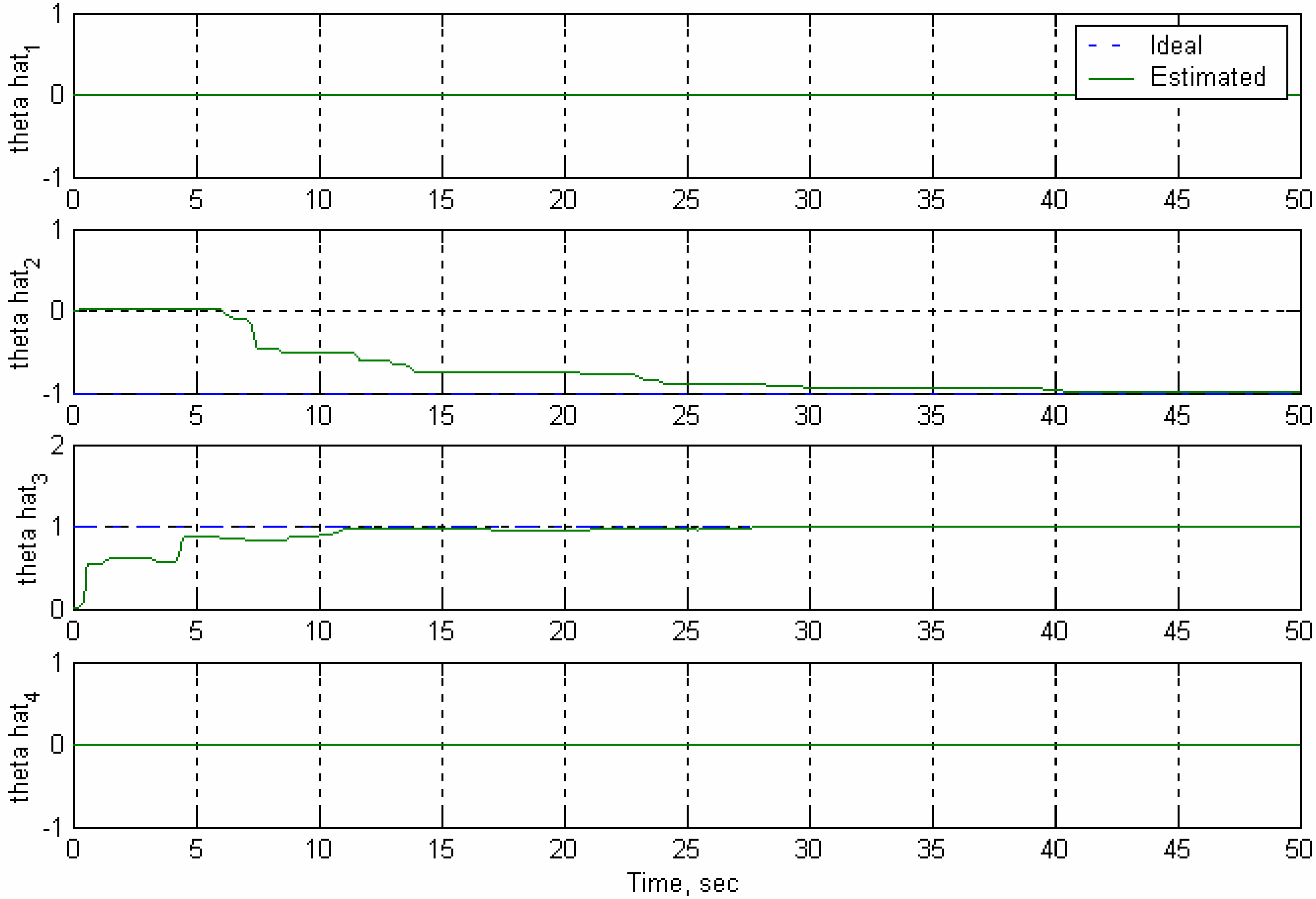
- Reference Input: $\boxed{r(t) = \sin(3t) + \sin\left(\frac{3t}{2}\right) + \sin\left(\frac{3t}{4}\right) + \sin\left(\frac{3t}{8}\right)}$

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



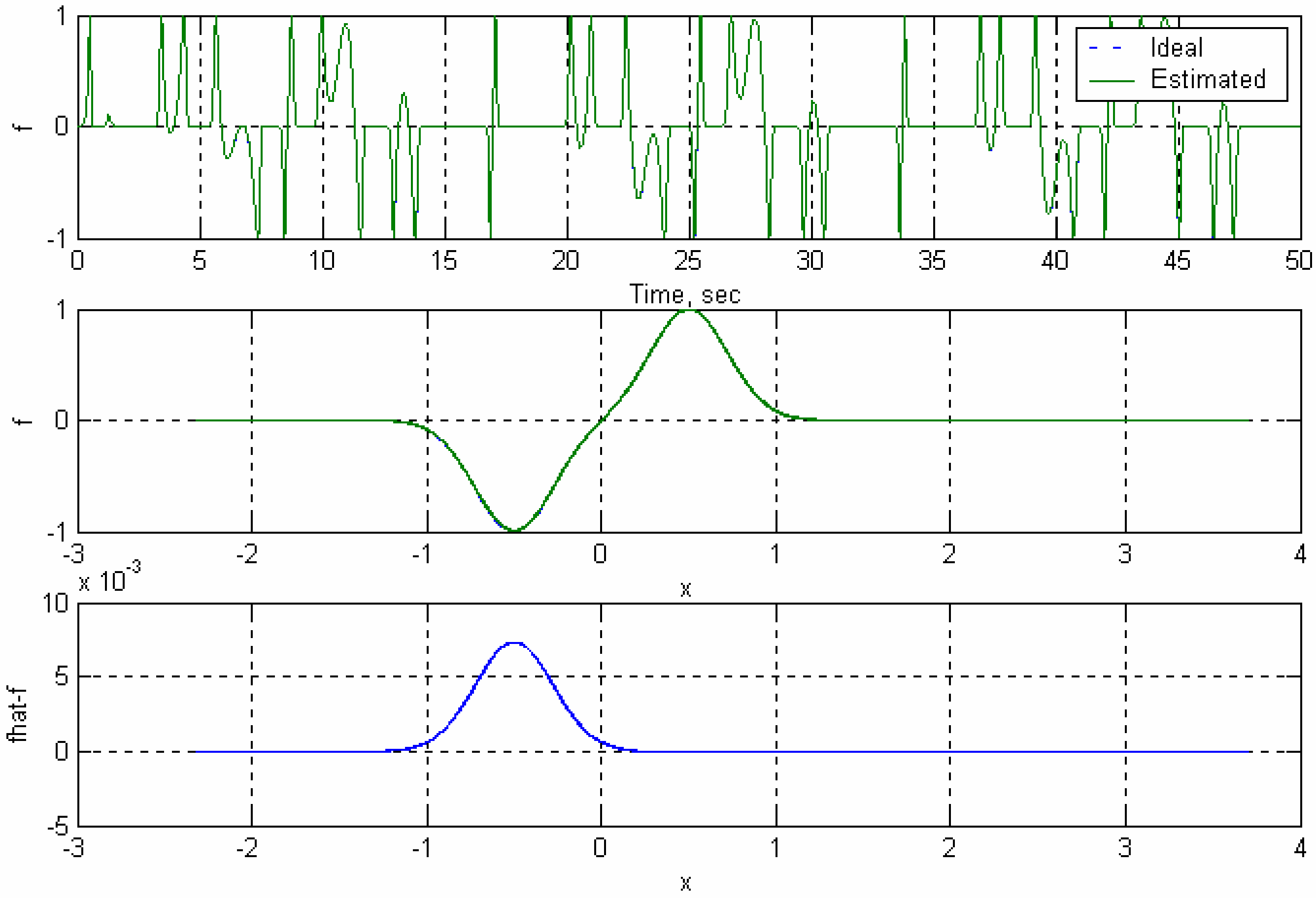
Good Tracking & Parameter Estimation

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



Nonlinearity: Good Parameter Estimation

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

$$\boxed{r \in R^m, \quad A_m \in R^{n\times n}, \quad B_m \in R^{n\times m}}$$

- **Control Goal**
  - find *u* such that: $\boxed{\lim_{t\to\infty}\left\|x\left(t\right) - x_m\left(t\right)\right\| = 0}$

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

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

$$\boxed{\begin{aligned} &A + B\Lambda\hat{K}_x^T - A_m = A + B\Lambda\hat{K}_x^T - A - B\Lambda K_x^T = B\Lambda\left(\hat{K}_x - K_x\right)^T = B\Lambda\Delta K_x^T \\ &B\Lambda\hat{K}_r^T - B_m = B\Lambda\hat{K}_r^T - B\Lambda K_r^T = B\Lambda\left(\hat{K}_r - K_r\right)^T = B\Lambda\Delta\hat{K}_r^T \end{aligned}}$$

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

- Tracking Error

$$\boxed{e(t) = x(t) - x_m(t)}$$

- Model matching conditions

$$\boxed{\begin{aligned} A + B\,\Lambda\,\hat{K}_x^T &= A_m + B\,\Lambda\,\Delta K_x^T \\ B\,\Lambda\,\hat{K}_r^T &= B_m + B\,\Lambda\,\Delta \hat{K}_r^T \end{aligned}}$$

- Error Dynamics:

$$\boxed{\begin{aligned} &\dot{e}(t) = \dot{x}(t) - \dot{x}_m(t) \\ &= \underbrace{\left(A + B\,\Lambda\,\hat{K}_x^T\right)}_{A_m + B\,\Lambda\,\Delta K_x^T} x + \left(\underbrace{B\,\Lambda\,\hat{K}_r^T}_{B_m + B\,\Lambda\,\Delta \hat{K}_r^T}\, r - B\,\Lambda \underbrace{\left(\hat{\Theta} - \Theta\right)}_{\Delta\Theta}^T \Phi(x)\right) - A_m\, x_m - B_m\, r \\ &= A_m\, e + B\,\Lambda\left(\Delta K_x^T\, x + \Delta K_r^T\, r - \Delta\Theta^T\, \Phi(x)\right) \end{aligned}}$$

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

- Lyapunov Function Candidate

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

  - where: $\text{trace}(S) \triangleq \sum s_{ii}$
  - $\left|\Lambda\right| \triangleq \text{diag}\left(\left|\lambda_1\right| \quad \ldots \quad \left|\lambda_m^i\right|\right)$ 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 $\boxed{P\, A_m + A_m^T\, P = -Q}$
    - $Q = Q^T > 0$ 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

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

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

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

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

- Using Barbalat's and Lyapunov-like Lemmas: $\boxed{\lim\limits_{t\to\infty} \dot{V}(x,t) = 0}$
- Since $\boxed{\dot{V} = -e^T(t)\, Q\, e^T(t)}$ it follows that: $\boxed{\lim\limits_{t\to\infty} \|e(t)\| = 0}$
- **Conclusions**
  - achieved asymptotic tracking for any bounded time varying command: $\boxed{x(t) \to x_m(t), \text{ as } t \to \infty}$
  - 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

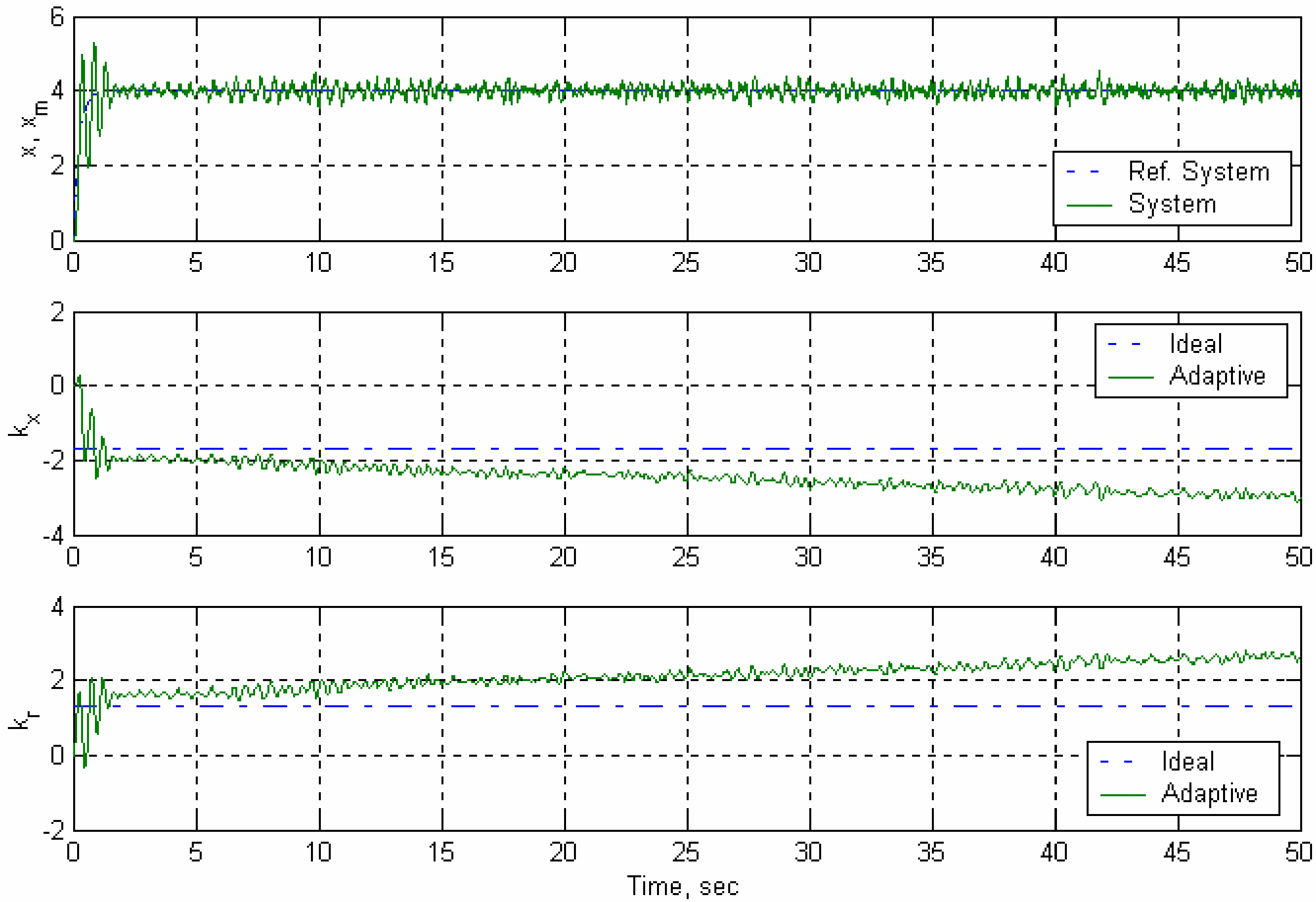
- Method is based on the observation that small tracking errors contain mostly noise and disturbance
- Solution
  - Turn off the adaptation process for “small” tracking errors
  - MRAC using Dead-Zone ⇨
  - ε is the size of the dead-zone
- Outcome
  - *Bounded* Tracking

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

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

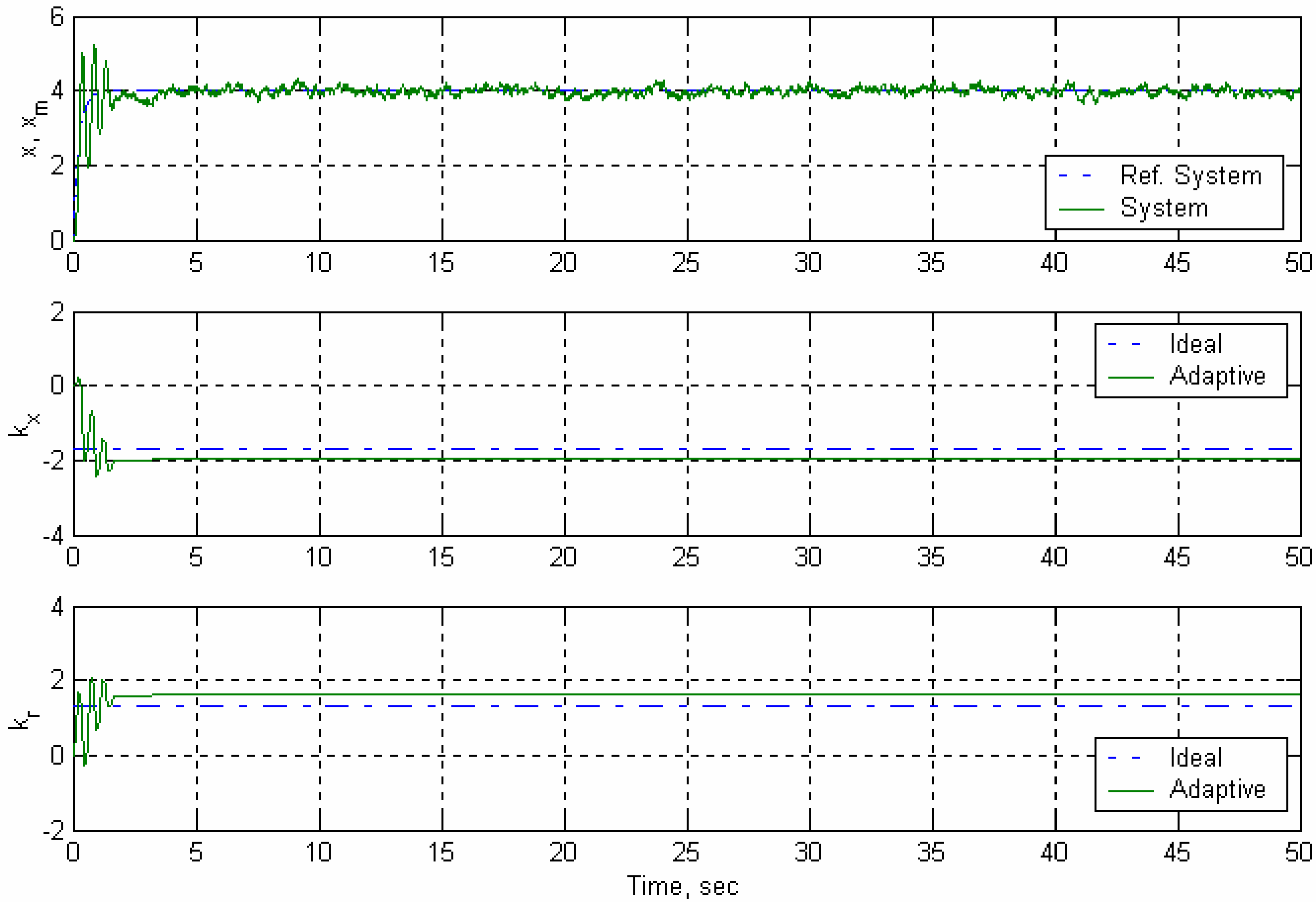
$$\dot{\hat{\Theta}} = \begin{cases} \Gamma_\Theta \, \Phi(x) e^T P \, B \operatorname{sgn}(\Lambda), & \|e\| > \varepsilon \\ 0, & \|e\| \le \varepsilon \end{cases}$$

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: $\boxed{m\,\ddot{x} = u}$
    - 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

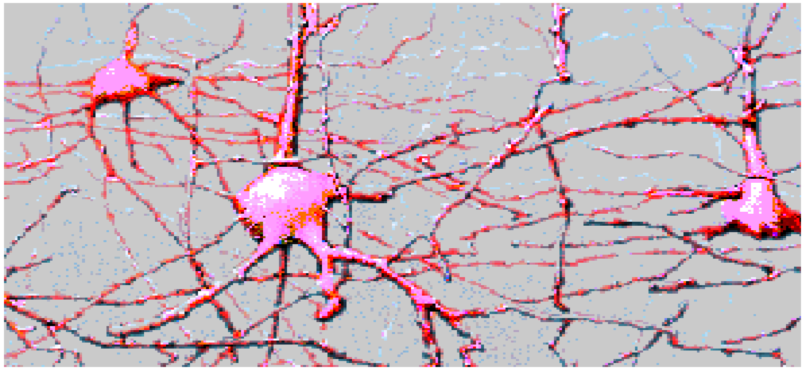
$$\hat{f}(x) = \sum_{i=1}^{N} \theta_i\,\varphi_i(x) + \varepsilon(x)$$

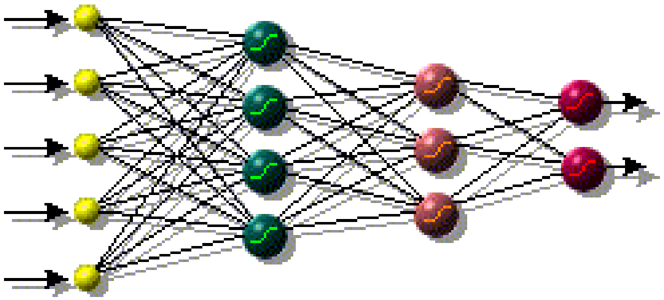
parametric

non-parametric

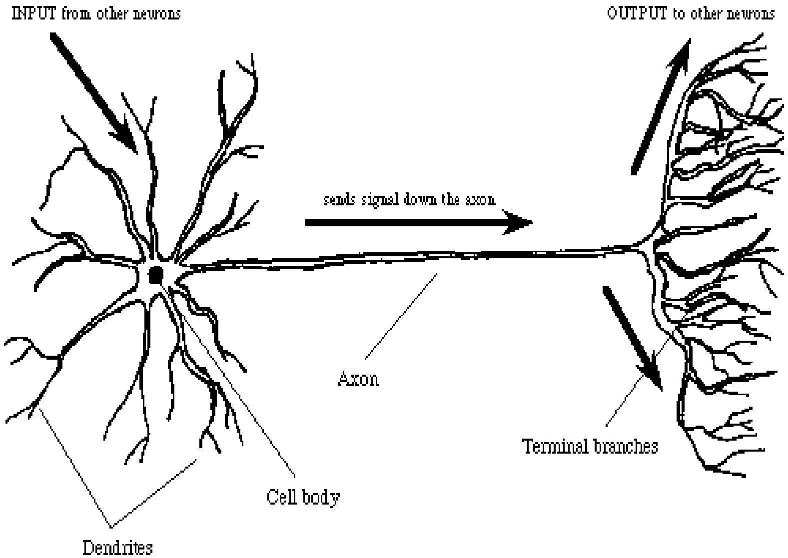
Enforcing Robustness in MRAC Systems

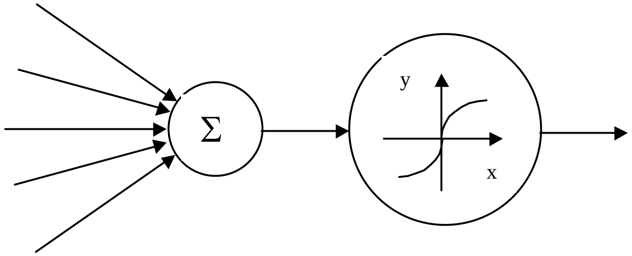
- Non-Parametric Uncertainty
  - Dead-Zone modification
  - Others ?
- Parametric Uncertainty
  - Need a set of basis functions that can approximate a large class of functions within a given tolerance
    - Fourier series
    - Splines
    - Polynomials
    - Artificial Neural Networks
      - sigmoidal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





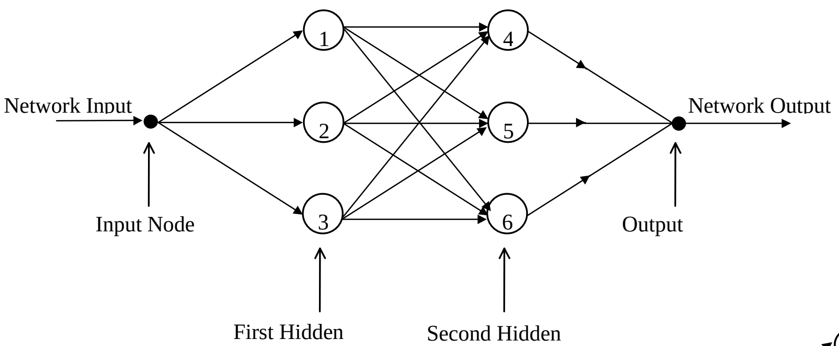
Artificial Neural Networks



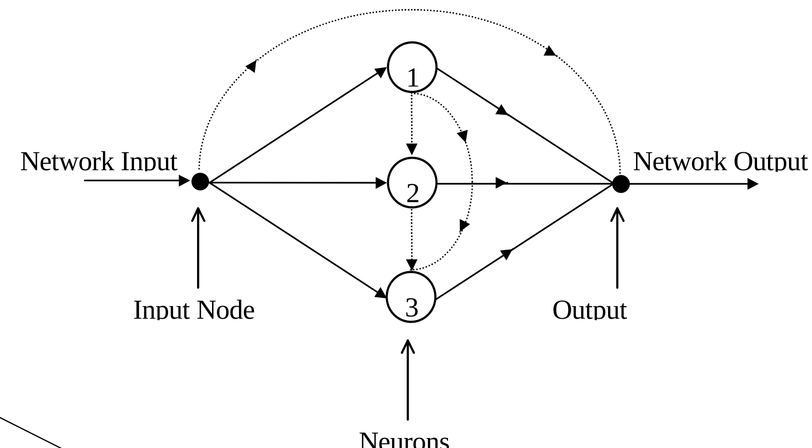


NN Architectures

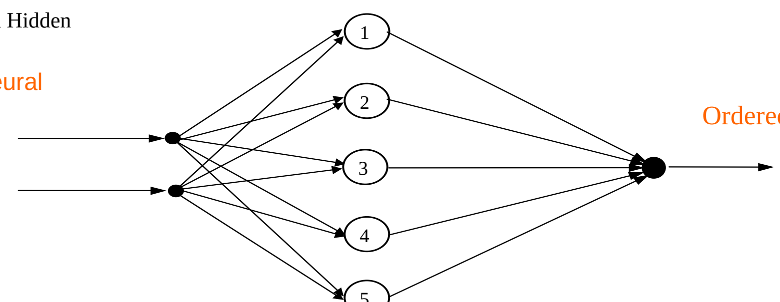
- ***Artificial Neural Networks*** are multi-input-multi-output systems composed of many inter-connected nonlinear processing elements (neurons) operating in parallel



Two-Hidden-Layers Neural Network



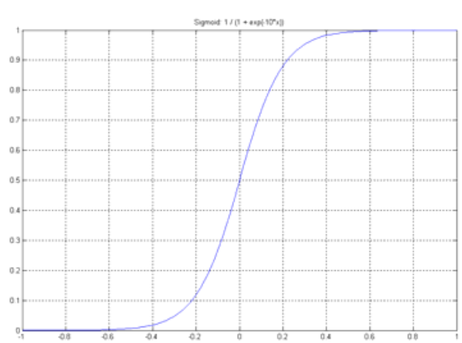
Ordered Neural Network

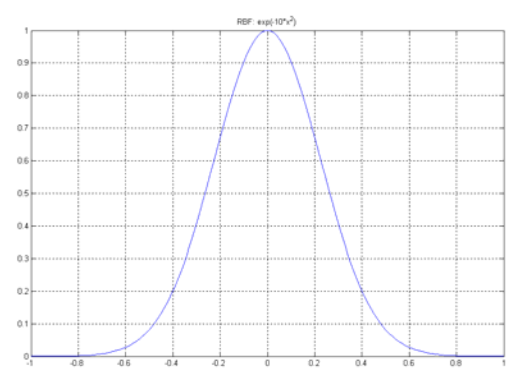


Single-Hidden-Layer Neural Network

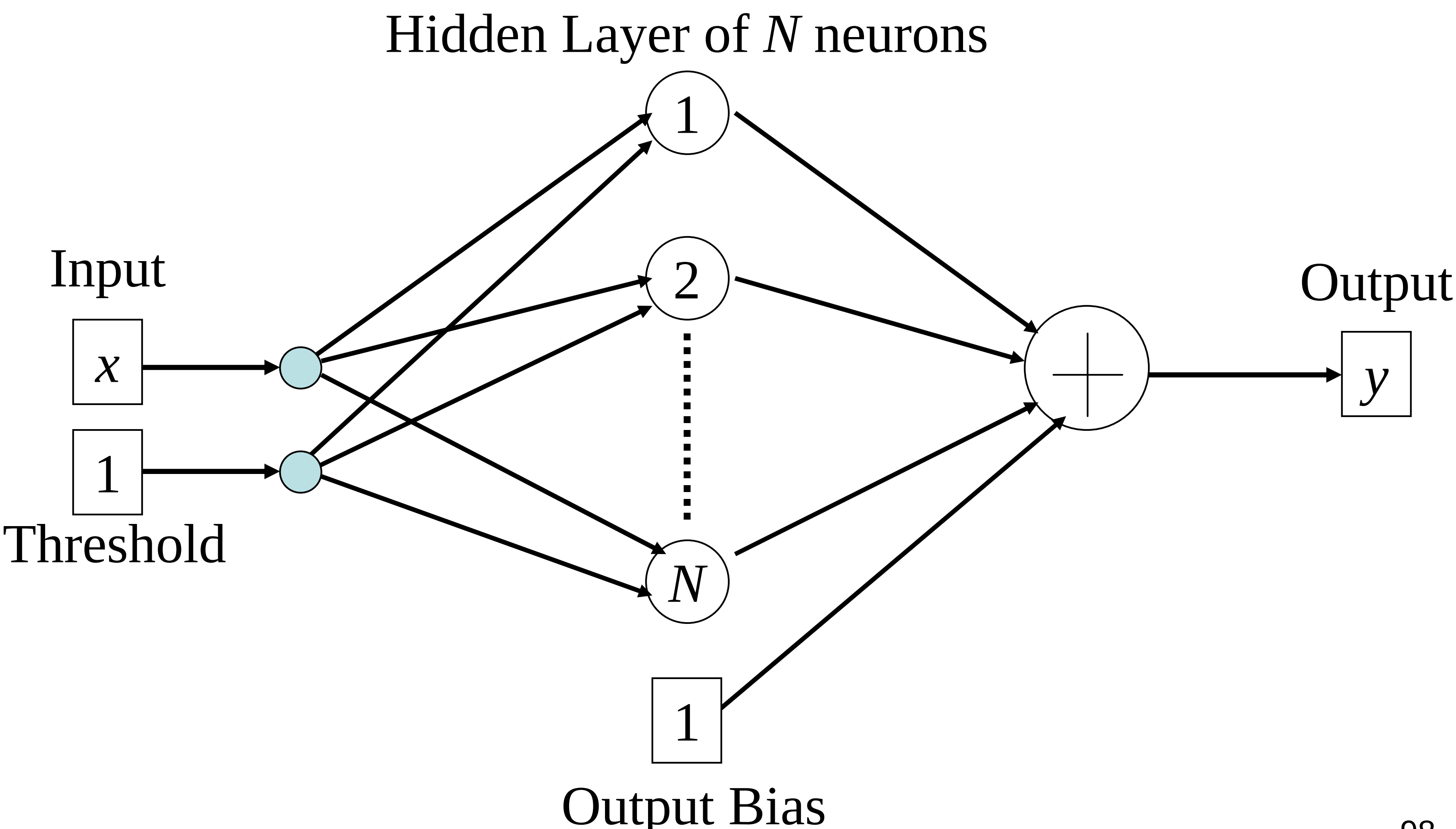
Single Hidden Layer (SHL) Feedforward Neural Networks (FNN)

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

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

Sigmoidal NN

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

- Vector of hidden layer sigmoids:

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

- Matrix of inner-layer weights: $\boxed{V = \left(\vec{v}_1 \quad \ldots \quad \vec{v}_N\right) \in R^{n \times N}}$

- Matrix of output-layer weights: $\boxed{W = \left(\vec{w}_1 \quad \ldots \quad \vec{w}_m\right) \in R^{N \times m}}$

- Vector of output biases $\boxed{c \in R^m}$ and thresholds $\boxed{\theta \in R^N}$

- k^{th} output:

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

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

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

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

- Introduce: $\boxed{W \triangleq \begin{bmatrix} W^T & b \end{bmatrix}^T, \quad V \triangleq \begin{bmatrix} V^T & \theta \end{bmatrix}^T, \quad \vec{\sigma} \triangleq \begin{bmatrix} \vec{\sigma} \\ 1 \end{bmatrix}, \quad \mu \triangleq \begin{bmatrix} x \\ 1 \end{bmatrix}}$
- Then: $\Longrightarrow$ $\boxed{NN(x) = W^T \vec{\sigma}\left(V^T \mu \right)}$

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:

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

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

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

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

RBF NN

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

RBF NN, (continued)

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

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

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

- Introduce: $\boxed{W \triangleq [W \quad b], \quad \Phi(x) \triangleq \begin{bmatrix} \Phi(x) \\ 1 \end{bmatrix}}$

- Then: $\Rightarrow$ $\boxed{NN(x) = W^T \Phi(x)}$

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:

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

- Inclusion of biases into NN output weight matrix simplifies bookkeeping

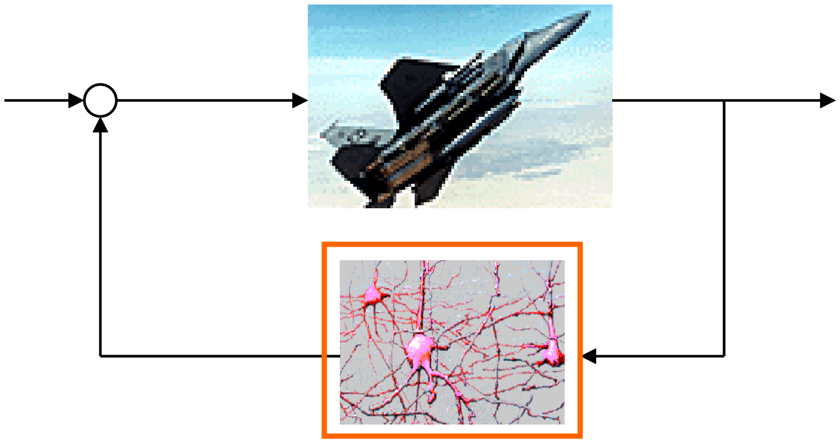
$$\boxed{NN(x) = W^T \Phi(x)}$$

- 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: $$\boxed{\dot{x} = A\,x + B\,\Lambda\big(u + f(x)\big), \quad x \in R^n, \quad u \in R^m}$$
  - $A \in R^{n\times n}, \Lambda = \text{diag}\begin{pmatrix}\lambda_1 & \dots & \lambda_m\end{pmatrix} \in R^{m\times m}$ are constant ***unknown*** matrices
  - $B \in R^{n\times m}$ is ***known*** constant matrix
  - $\forall\, i = 1,\dots,m \quad \text{sgn}(\lambda_i)$ is ***known***
- ***Approximation of uncertainty***: $$\boxed{f(x) = \Theta^T\,\Phi(x) + \varepsilon_f(x)}$$
  - ***matrix*** of constant ***unknown*** parameters: $\Theta \in R^{m\times N}$
  - vector of *N* ***fixed*** RBF-s: $\Phi(x) = \begin{pmatrix}\varphi_1(x) & \dots & \varphi_N(x)\end{pmatrix}^T$
  - 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:

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

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

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

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

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

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

$$\boxed{\begin{aligned} &A + B\,\Lambda\,\hat{K}_x^T - A_m = A + B\,\Lambda\,\hat{K}_x^T - A - B\,\Lambda\,K_x^T = B\Lambda\left(\hat{K}_x - K_x\right)^T = B\,\Lambda\,\Delta K_x^T \\ &B\,\Lambda\,\hat{K}_r^T - B_m = B\,\Lambda\,\hat{K}_r^T - B\,\Lambda\,K_r^T = B\Lambda\left(\hat{K}_r - K_r\right)^T = B\,\Lambda\,\Delta\hat{K}_r^T \end{aligned}}$$

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

- Tracking Error: $\boxed{e(t) = x(t) - x_m(t)}$
- Error Dynamics:

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

- Remarks
  - function estimation error $\varepsilon_f(x)$ is bounded for $x \in X$
  - need to keep x within X

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

- Lyapunov Function Candidate

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

  - where: $\operatorname{trace}(S) \triangleq \sum s_{ii}$
  - $\left|\Lambda\right| \triangleq \operatorname{diag}\left(\left|\lambda_1\right| \quad \ldots \quad \left|\lambda_m^i\right|\right)$ 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 $\boxed{P\, A_m + A_m^T\, P = -Q}$
    - $Q = Q^T > 0$ is any symmetric positive definite matrix

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

- Time-derivative of the Lyapunov function

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

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

- Time-derivative of the Lyapunov function

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

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

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

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

- Time-derivative of the Lyapunov function

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

- 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)

- If we choose adaptive laws as before:

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

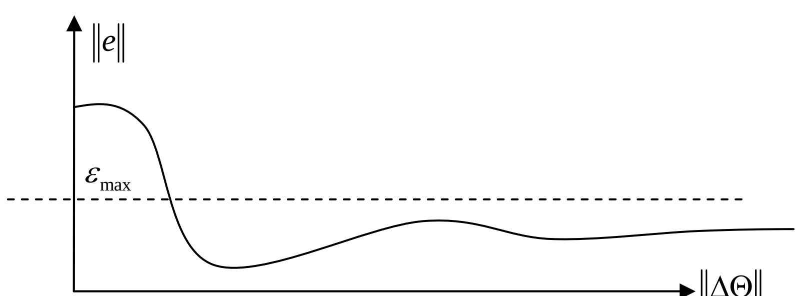
- Then:

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

- Consequently, $\dot{V} < 0$ outside of ⇨

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

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



How to Keep Adaptive Parameters Bounded?

- σ – modification:

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

- ε – modification:

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

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

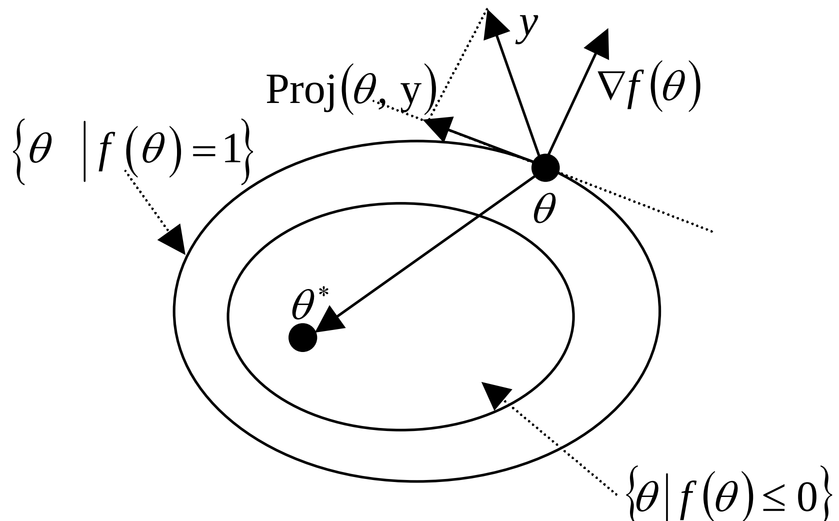
Introducing Projection Operator

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

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



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

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

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

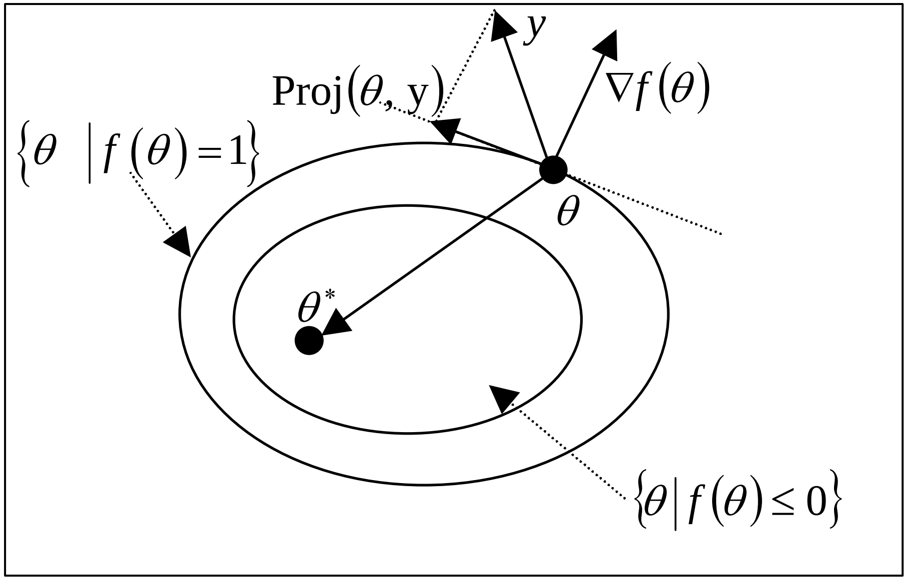
- $\theta_{\max}$ specifies boundary
- ε_θ specifies boundary tolerance

Projection Operator, (continued)

- ***Definition***:

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

- Depends on (θ, y)
- Does not alter *y* if θ is within the pre-specified bounds: $\boxed{\|\theta\| \le \theta_{\max}}$
- Gradient: $\boxed{\nabla f(\theta) = \dfrac{2\theta}{\varepsilon_\theta \theta_{\max}^2}}$
- 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*** $\boxed{(\theta - \theta^*)^T (\mathrm{Proj}(\theta, y) - y) \le 0}$

Lyapunov Function Time-Derivative with Projection Operator

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

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

Adaptation with Projection

- Modified adaptive laws:

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

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

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

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

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

Example: Projection Operator, (scalar case)

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

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

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

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

  - Projection Operator:

$$\boxed{y = -x\,e\,\operatorname{sgn}(b)} \Rightarrow \boxed{\operatorname{Proj}\left(\hat{k}, y\right) = \begin{cases} y\left(1 - f\left(\hat{k}\right)\right), & \text{if } f\left(\hat{k}\right) > 0 \text{ and } y\, f'\left(\hat{k}\right) > 0 \\ y, & \text{if not} \end{cases}}$$

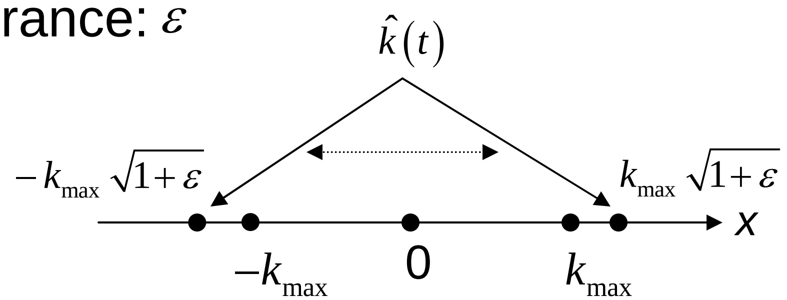
Example: Projection Operator, (scalar case) (continued)

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

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

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

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

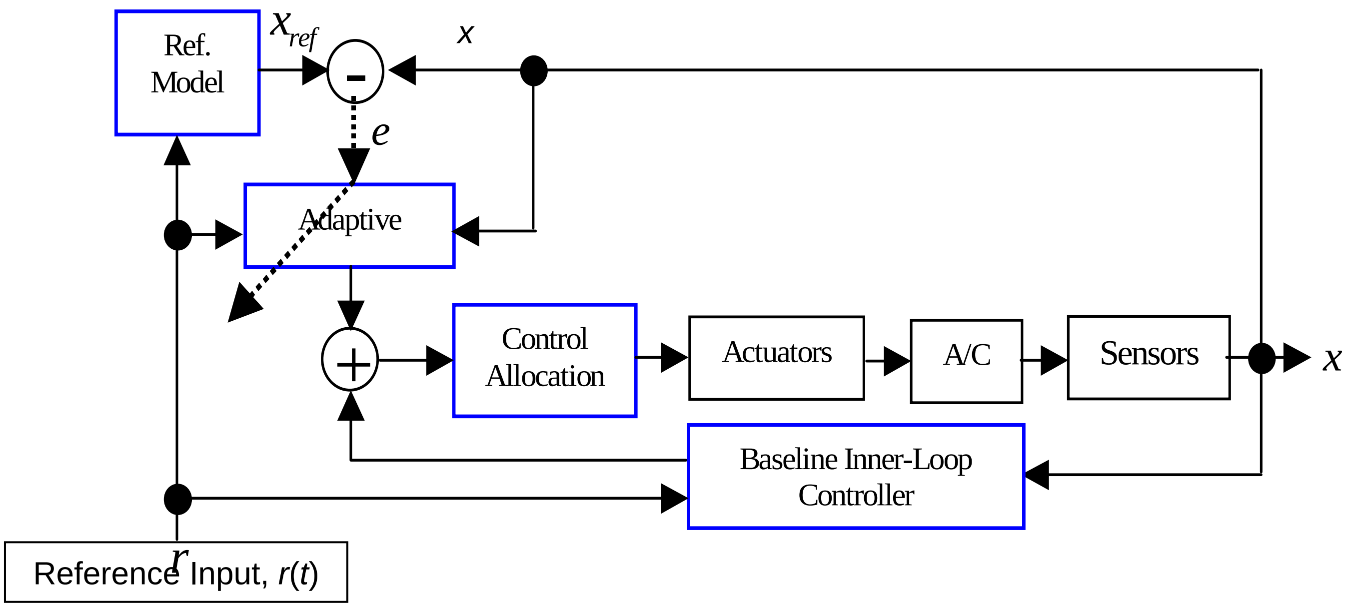
$$\begin{aligned} u &= \hat{K}_x^T\, x + \hat{K}_r^T\, r - \hat{\Theta}^T\, \Phi(x) \pm u_{nom} \\ &= u_{nom} + \underbrace{\left(\hat{K}_x - F_x\right)^T}_{\hat{D}_x} x + \underbrace{\left(\hat{K}_r - F_r\right)^T}_{\hat{D}_r} r - \hat{\Theta}^T\, \Phi(x) \\ &= u_{nom} + \boxed{\hat{D}_x^T\, x + \hat{D}_r^T\,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

adaptive increment

- *Incremental* Adaptation:

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

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 <u>*Sigmoidal*</u> NN

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

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

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

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

Adaptive Control using *Sigmoidal* NN

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

- 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):

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

  – 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: $\boxed{\dot{x}_c = A_c\, x_c + B_{1\,c}\, x_p + B_{2\,c}\, u}$
- 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:

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

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

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

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:

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

- ***<u>Assumption</u>***: 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:

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

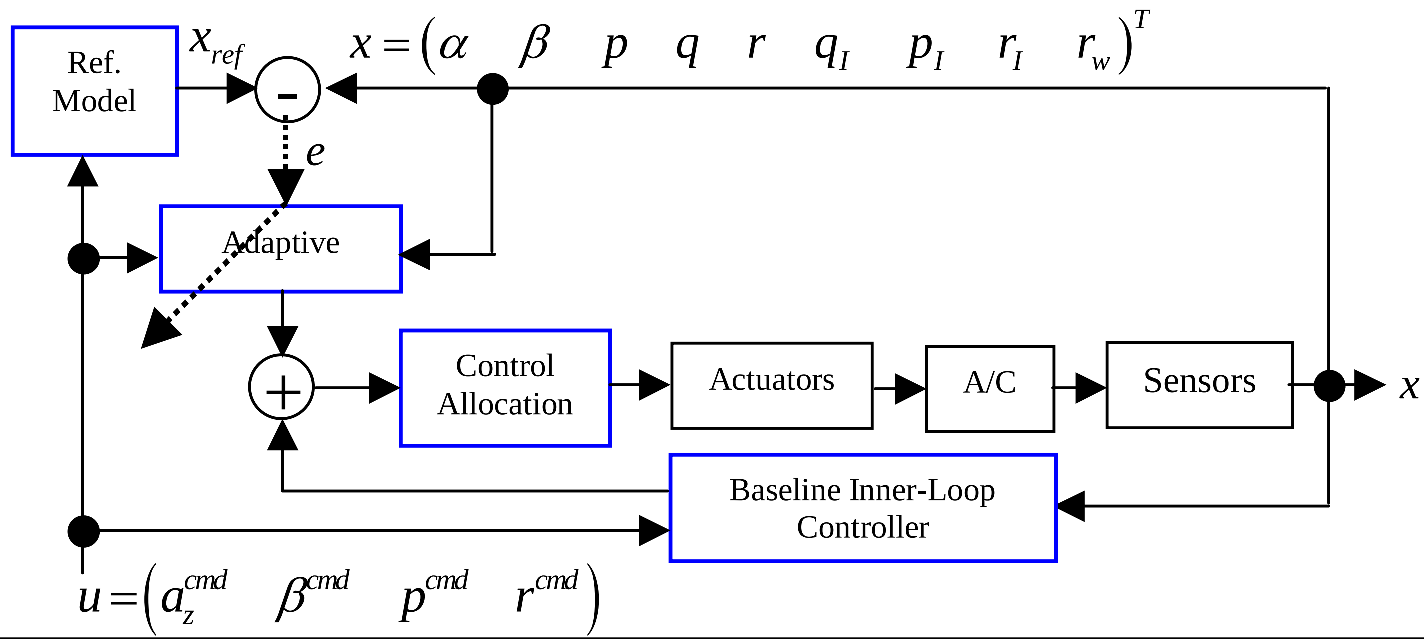
$$= \underbrace{\delta_L\left(x,u\right)}_{\delta_L\left(x,u\right)} + \underbrace{\left(\hat{K}_x - K_x\right)}_{\hat{k}_x}^T x + \underbrace{\left(\hat{K}_u - K_u\right)}_{\hat{k}_u}^T u - \underbrace{\hat{K}_0\left(x_p\right)}_{\hat{\Theta}^T\,\Phi\left(x_p\right)}$$

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

- 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



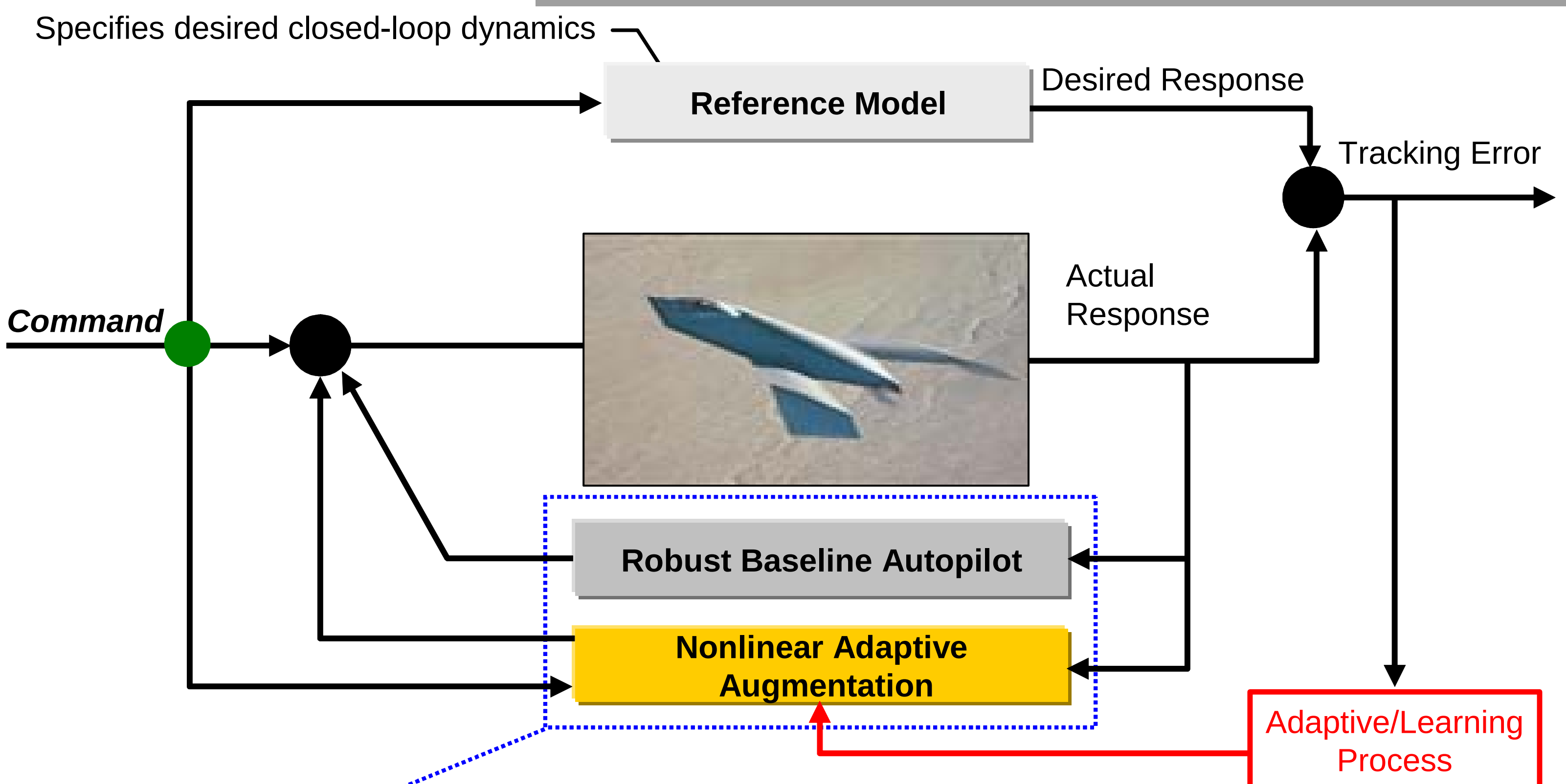
- Reference Model provides desired response
- Nominal Baseline Inner-Loop Controller
- Adaptive Augmentation
  - *<u>Dead-Zone</u>* modification prevents adaptation from changing nominal closed-loop dynamics
  - *<u>Projection Operator</u>* bounds adaptation parameters / gains

Adaptive Augmentation of a Baseline Flight Control System

- Maintains tracking in the presence of unknown events
- Flight proven technology, (X-36, JDAM MK-84, MK-82)

Modifications for Robustness

– ***Dead-Zone*** prevents adaptation unless nominal closed-loop response degrades
– ***Projection Operator*** and **ε – mod** bound adaptive signals
– **μ – mod** protects adaptation from windup during control saturation



***Adaptive Flight Control System* = Robust Baseline Autopilot + Nonlinear Adaptive Augmentation**

Adaptive Backstepping

Why?

- MRAC requires model matching conditions

$$\boxed{\begin{aligned} &A + B\Lambda K_x^T = A_m \\ &B\Lambda K_r^T = B_m \end{aligned}}$$

- 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 2[nd] order *cascaded* system

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

- 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

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

Backstepping Design

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

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

- Dynamic inversion using pseudo control:

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

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

Backstepping Design (continued)

- Dynamic inversion using actual control

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

- 2nd state error dynamics

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

- *Asymptotically stable* error dynamics

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

nonlinear system

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

Adaptive Backstepping Design

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

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

- Dynamic inversion using pseudo control and estimated functions:

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

- 1st state error dynamics:

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

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:

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

- Uncertainty parameterization, function and parameter estimation errors:

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

$$\boxed{\begin{aligned} \Delta\theta_{f_i} &\triangleq \hat{\theta}_{f_i} - \theta_{f_i} \\ \Delta\theta_{g_i} &\triangleq \hat{\theta}_{g_i} - \theta_{g_i} \end{aligned}}$$

Adaptive Backstepping Design (continued)

- Tracking error dynamics:

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

$$\Updownarrow$$

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

- Stable robust adaptive laws:

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

- ***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.

Overview

- Problem: Assure stability of an adaptive control system in the presence of actuator position / rate saturation constraints.
- Solutions
  - Ad-hoc
  - Proof-by-simulation

    not acceptable
  - Lyapunov based

    preferred
    - limited class of systems
    - sufficient conditions are often hard-to-verify
- Need: *Theoretically justified* and *verifiable* conditions for stable adaptation and control design with a possibility of *avoiding* actuator saturation phenomenon.
- Design Solutions include modifications, (adaptive / fixed gain) to:
  - control input
  - tracking error
  - reference model

Known 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 of reference input
- E. Lavretsky, N. Hovakimyan, (2004)
  - positive μ – modification
    - adaptive modification of control and reference inputs
    - rigorous stability proof and verifiable sufficient conditions
    - capability to completely avoid control saturation

Adaptive Control in the Presence of Input Constraints: Problem Formulation

battle damage

control failures

- System dynamics: $$\boxed{\dot{x}(t) = A\,x(t) + b\,\lambda\,u(t),\quad x \in R^n, u \in R}$$
  - *A* is ***unknown*** matrix, (emulates battle damage)
  - *b* is ***known*** control direction
  - $\lambda > 0$ is ***unknown*** positive constant, (control failures)
- Static actuator

amplitude saturation

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

commanded input

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

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

Hurwitz

bounded reference input

Preliminaries

safety zone

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

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

linear feedback / feedforward component

control deficiency feedback

- Need ***explicit*** form of u_c

Positive μ – modification

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

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

continuous in time but *not* continuously differentiable

$$\boxed{\delta = 0 \wedge \left(\mu = 0 \vee \mu = \infty\right)} \Rightarrow \boxed{u = u_{\max}\,\mathrm{sat}\left(\frac{u_{lin}}{u_{\max}}\right)}$$

Closed-Loop Dynamics

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

linear control deficiency

$$\boxed{\begin{gathered}\dot{x} = A\,x + b\,\lambda\,u_{lin} + b\,\lambda\,\overbrace{\left(\mu\,\Delta u_c + \Delta u\right)}^{\Delta u_{lin} = u - u_{lin}} \\ \text{where: } \Delta u_{lin} = u_{\max}\,\text{sat}\left(\frac{u_c}{u_{\max}}\right) - u_{lin}\end{gathered}}$$

$\Updownarrow$

does not depend on μ explicitly

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

Adaptive Reference Model Modification

- Closed-loop system:

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

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

adaptive augmentation

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

reference input

- Matching conditions:

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

Adaptive Laws Derivation

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

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

- Lyapunov function:

$$\boxed{\begin{array}{l} V\left(e, \Delta k_x, \Delta k_r, \Delta k_u\right) = e^T\, P\, e + \lambda\left(\Delta k_x^T\, \Gamma_x^{-1}\, \Delta k_x + \gamma_r^{-1}\, \Delta k_r^2 + \gamma_u^{-1}\, \Delta k_u^2\right) \\ \text{where: } P\, A_m + A_m\, P = -Q < 0 \end{array}}$$

Stable Parameter Adaptation

- Adaptive laws derived to yield stability:

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

- For open-loop stable systems – global result
- For open-loop unstable systems verifiable sufficient conditions established:
    - upper bound on $r_{\max}$
    - lower bound on μ
    - upper bounds on initial conditions *x*(0) and Lyapunov function *V*(0)

$\mu-$ mod Design Steps

- Choose "safety zone" $\boxed{0 < \delta < u_{\max}}$ and sufficiently large $\boxed{\mu > 0}$
- Define *virtual* constraint: $\boxed{u_{\max}^{\delta} = u_{\max} - \delta}$
- *Linear* component of adaptive control signal: $\boxed{u_{lin} = k_x^T\, x + k_r\, r(t)}$
- <u>*Total*</u> adaptive control with $\mu-$ mod:

adaptive laws

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

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

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

modified reference model

Simulation Example

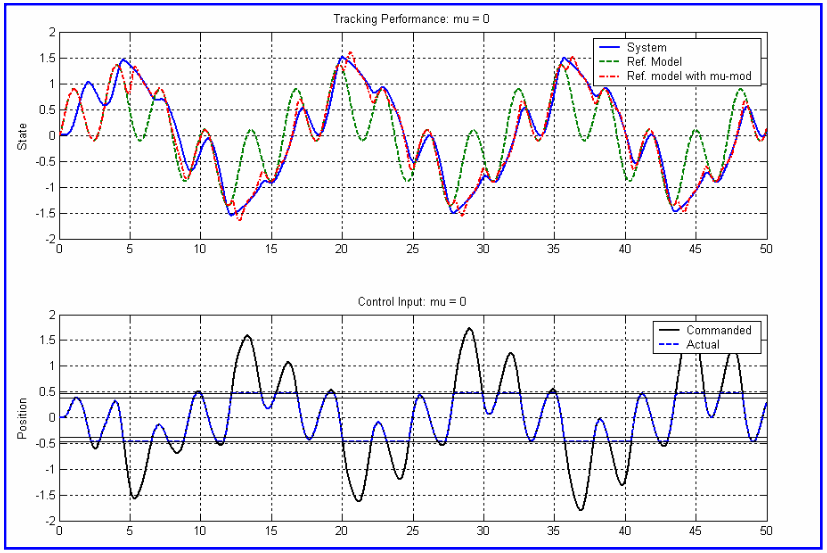
- Unstable open-loop system:

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

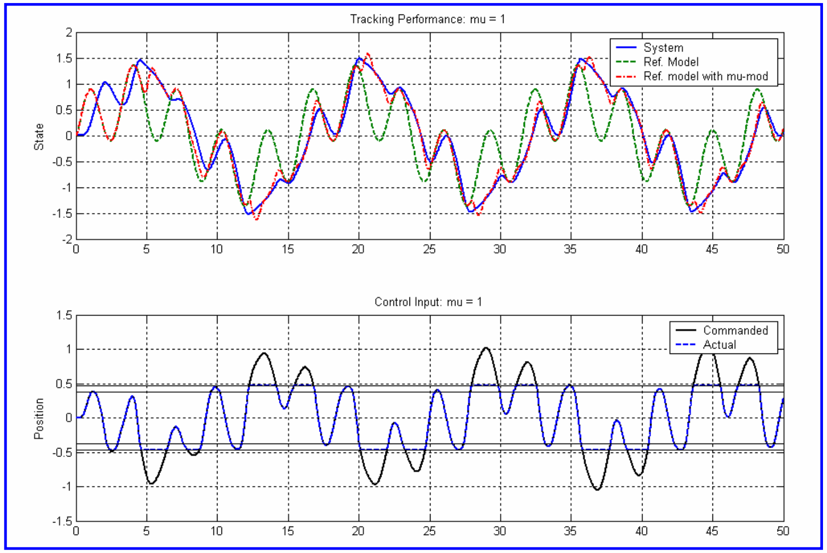
- Choose: $\boxed{\delta = 0.2\,u_{\max}}$ $\longrightarrow$ $\boxed{u_{\max}^{\delta} = u_{\max} - \delta = 0.8\,u_{\max}}$
- Ideal reference model: $\boxed{\dot{x}_m = -6\left(x_m - r(t)\right)}$
- Reference input: $\boxed{r(t) = 0.7\left(\sin(2t) + \sin(0.4t)\right)}$
- Adaptation rates set to unity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

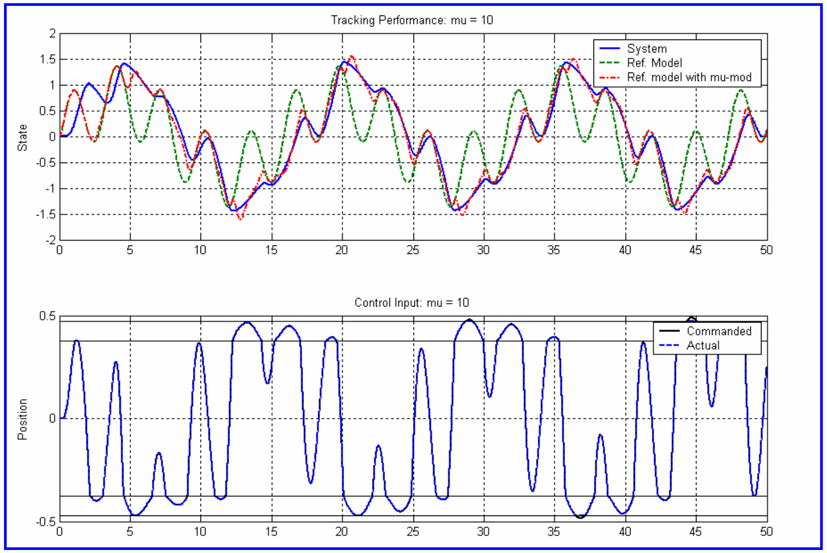
Simulation Data

$\mu = 0$

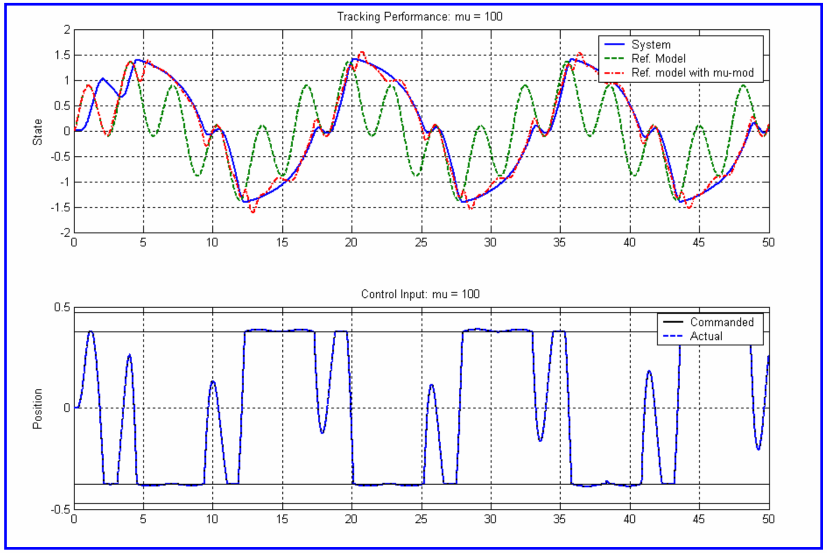


$\mu = 1$





$\mu = 10$



$\mu = 100$

μ – mod Design Summary

- Lyapunov based
- Provides closed-loop stability and bounded tracking
  - convex combination of linear adaptive control and its $u_{\max}^{\delta}$ – limited value
  - adaptive reference model modification
- Verifiable sufficient conditions
- Flight proven technology
- **<u>Future Work</u>**
  - account for control rate constraints using dynamic actuator model

Control of Nonaffine Systems using Time-Scale Separation*

*in collaboration with Prof. Naira Hovakimyan of Virginia Tech

Motivation

- Often in practice, control effectiveness is nonlinear in control, (such as in flight control applications) $I_y \dot{q} = M_0(\alpha, q) + \boxed{M_{\delta_e}(\alpha, q, \delta_e)}\, \delta_e$
- **Need**: Control design methods for nonaffine-in-control systems

Design Overview

- Scalar system: $\boxed{\dot{x} = f(x,u)}$
- Assumptions
  - no loss of control: $\boxed{f_u \triangleq \dfrac{\partial f(x,u)}{\partial u} \neq 0, \quad \forall \left(x \in X \wedge u \in U \right)}$
    - $\boxed{\operatorname{sgn} f_u}$ is known
    - ideal dynamic inversion (DI) solution exists
  - ideal DI solution can not be expressed analytically, (i.e., using elementary functions)
    - example: $\boxed{f(x,u) = x^2 u + \tan u, \forall \left(x \in R \wedge u \in \left(-\dfrac{\pi}{2}, \dfrac{\pi}{2} \right) \right)}$

Approximate DI Solution

- Control Goal
  - tracking of $r(t)$ – bounded reference input whose time derivative is known and bounded
- “Fast” control approximation dynamics

$$\boxed{\varepsilon \dot{u} = -\left[\operatorname{sgn} f_u(x,u)\right]\left(f(x,u) - \dot{r} + k(x-r)\right), \text{ where: } \begin{cases} 0 < \varepsilon \ll 1 \\ k > 0 \end{cases}}$$

$$\Downarrow$$

$$\boxed{f(x,u) = \dot{r} - k(x-r) + \mathrm{o}(1), \quad \text{as } t \to \infty}$$

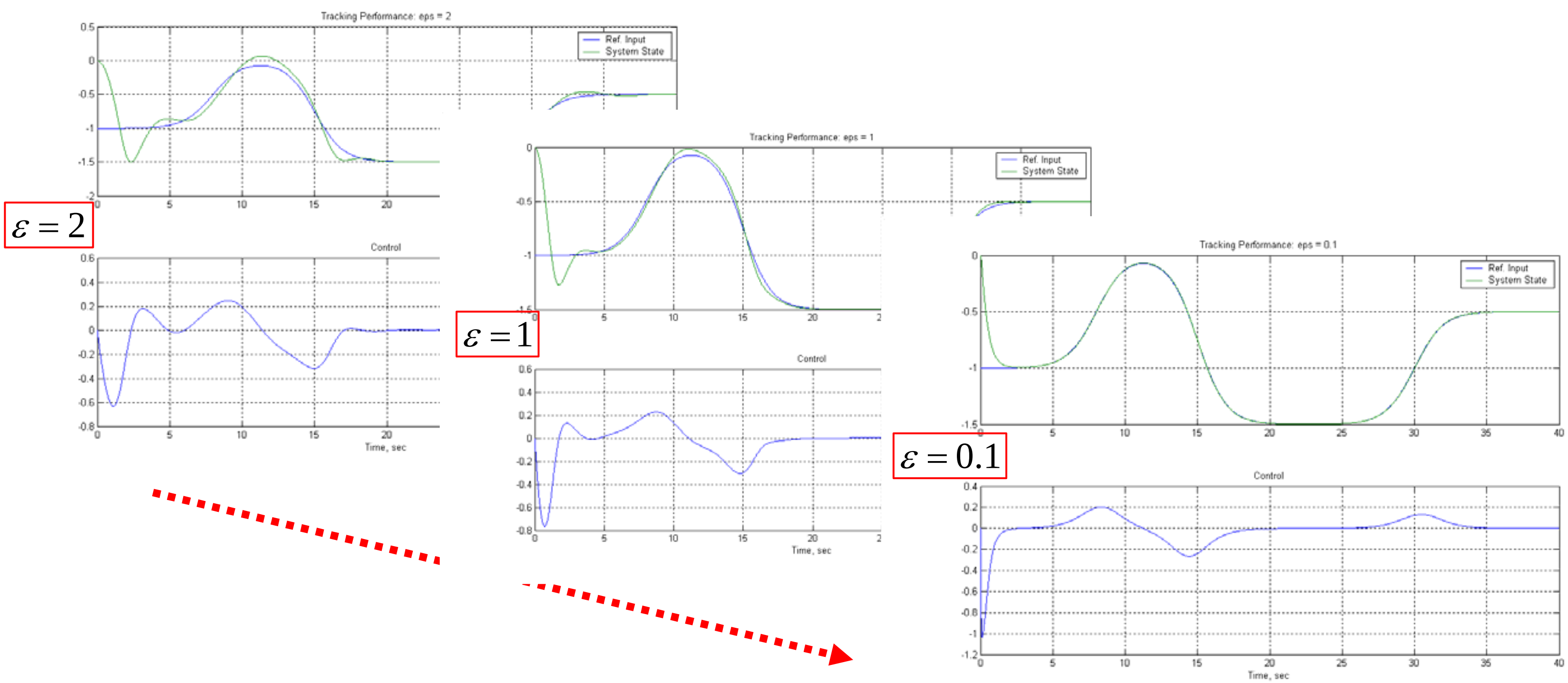
$$\Downarrow$$

$$\boxed{\dot{e} = -k\,e + \mathrm{o}(1), \quad \text{as } t \to \infty, \quad \text{where } e = x - r}$$

- Using time-scale separation method (due to Tikhonov)
  - on-line approximation of the ideal DI solution
  - asymptotically stable tracking error dynamics

Nonaffine-in-Control Design Example 1

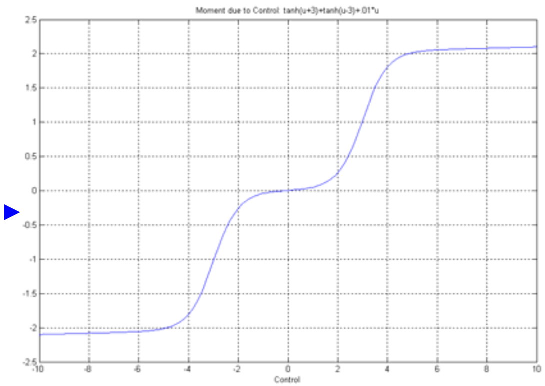
- Scalar system dynamics: $\boxed{\dot{x} = x^2 u + \tan u}$
- Reference input: $\boxed{r(t) = -1 + \sigma(t-8) - 1.5\sigma(t-15) + \sigma(t-30)}$
- Approximate DI: $\boxed{\varepsilon \dot{u} = -\left(x^2 u + \tan u - \dot{r} + 2(x-r)\right)}$



Nonaffine-in-Control Design Example 2
(emulating short period dynamics)

- "Pitch rate" dynamics:

$$\dot{x} = 0.5\,x + \underbrace{\tanh(u-3) + \tanh(u+3) + 0.01u}_{\text{Moment due to Control}}$$

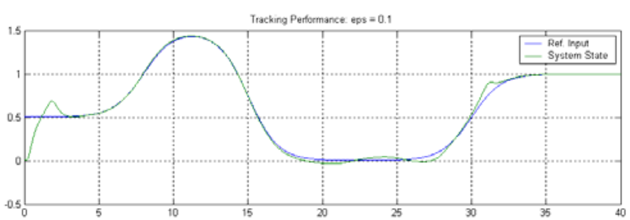


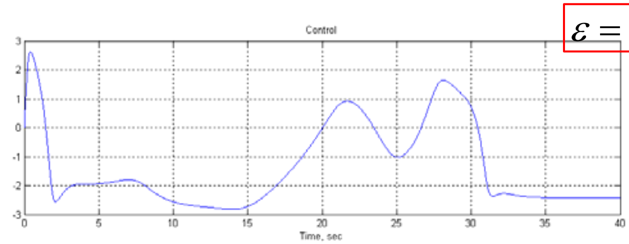
- Reference input:

$$r(t) = \sigma(t-8) - 1.5\sigma(t-15) + \sigma(t-30) + 0.5$$

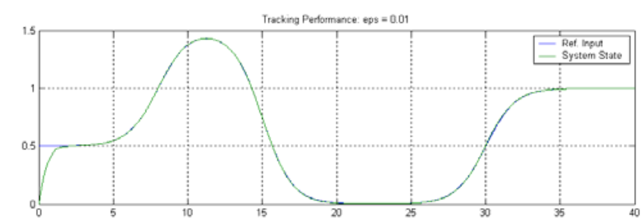
- Approximate DI:

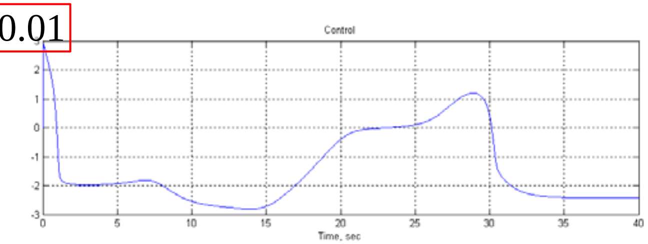
$$\varepsilon\,\dot{u} = -\left(0.5\,x + \tanh(u-3) + \tanh(u+3) + 0.01u - \dot{r} + 2(x-r)\right)$$





$\varepsilon = 0.1$ → $\varepsilon = 0.01$





Adaptive Control using State Predictor

Scalar system dynamics:

$$\dot{x}(t) = \boxed{f(x(t), u(t))}, \quad x(0) = x_0, \quad t \geq 0$$

unknown

Approximation via a set of Gaussians (RBF-s) on a compact set $\Omega_x \times \Omega_u$

$$f(x, u) = W^\top \Phi(x, u) + \varepsilon(x, u), \quad |\varepsilon(x, u)| < \varepsilon^*$$

State predictor using series-parallel model:

$$\dot{\hat{x}}(t) = \hat{W}^\top(t)\Phi(x(t), u(t)) - a(\hat{x}(t) - x(t)), \; \hat{x}(0) = \hat{x}_0$$

Prediction error dynamics:

$$\dot{e}_s(t) = a e_s(t) + \tilde{W}^\top(t)\Phi(x(t), u(t)) - \varepsilon(x(t), u(t))$$

$$e_s(0) = e_{s_0} (= \hat{x}_0 - x_0)$$

Adaptive Dynamic Inversion

Parameter Adaptation Law:

$$\dot{\hat{W}}(t) = \Gamma \mathrm{Proj}(\hat{W}(t), -e_s(t)\Phi(x(t), u(t))),\ \hat{W}(0) = W_0$$

Result: Ultimate boundedness of $e_s(t)$ and $\tilde{W}(t)$

If $\varepsilon^* = 0$ then, following Barbalat's lemma $\lim_{t\to\infty} e_s(t) = 0$

Fast dynamics:

$$\epsilon\dot{u}(t) = -(\hat{W}^\top(t)\Phi(x(t), u(t)) - a e_s(t) + k(\hat{x}(t) - \mathrm{r}(t)) - \dot{\mathrm{r}}(t))$$

Moral: Fast dynamics inverts the series parallel model, while the model tracks the state of the original system with bounded errors.
If $\varepsilon^* = 0$ then this tracking is asymptotic.

Example 3: Adaptive Design

Consider the scalar system (emulating pitch rate dynamics)

$$\dot{x}(t) = 0.5x(t) + \tanh(u(t)+3) + \tanh(u(t)-3) + 0.01u(t)$$

$$x(0) = -0.5$$

Consider the reference input:

$$\mathrm{r}(t) = \frac{1}{1+\exp(t-8)} - \frac{1.5}{1+\exp(t-15)} + \frac{1}{1+\exp(t-30)} + 0.5$$

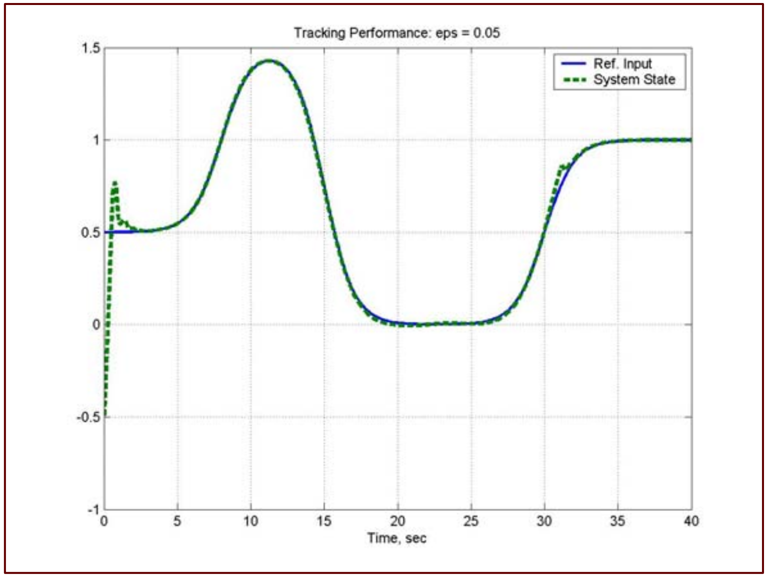
Series parallel model:

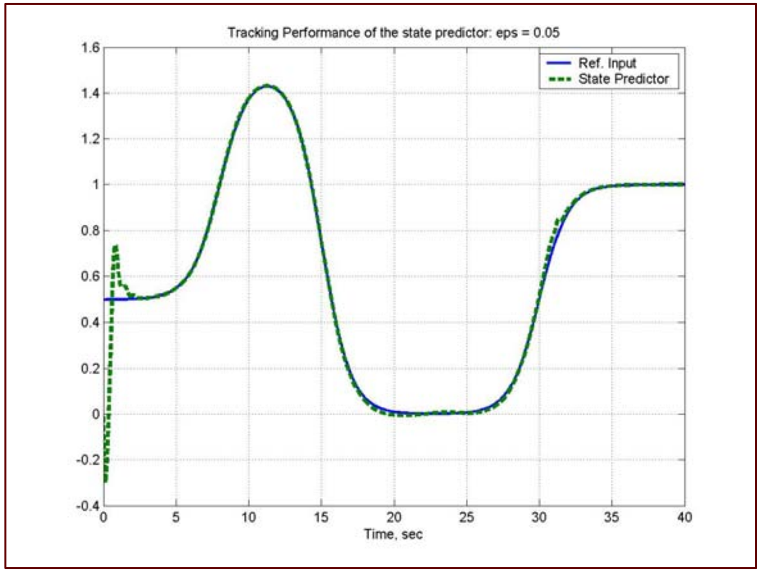
$$\dot{\hat{x}}(t) = \hat{W}^\top(t)\Phi(x(t), u(t)) - 10(\hat{x}(t) - x(t)), \hat{x}(0) = 0$$

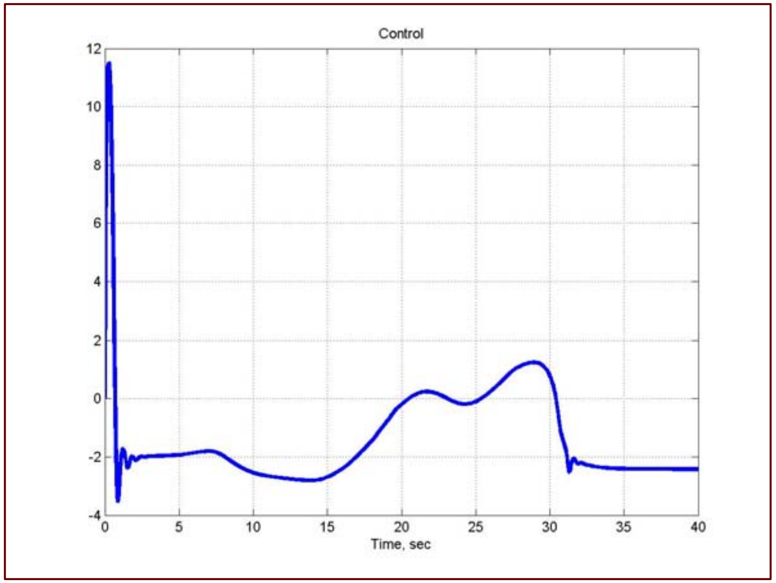
Fast dynamics:

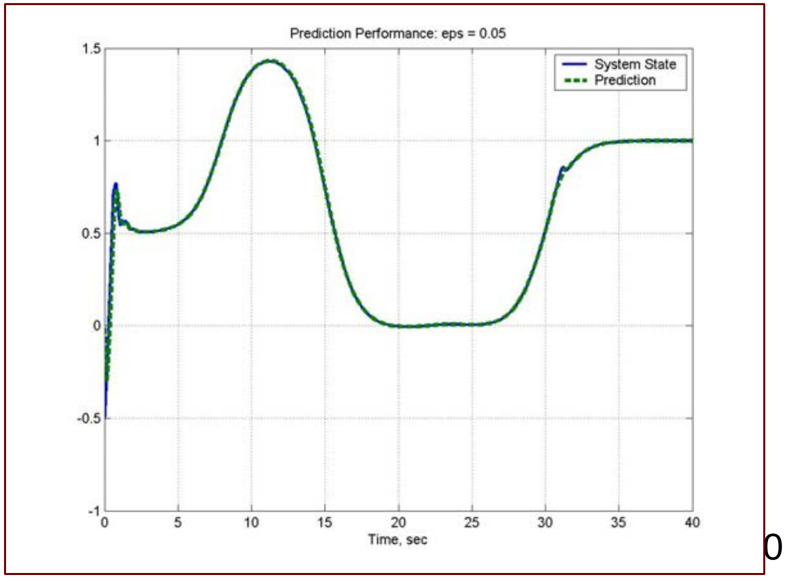
$$0.05\dot{u} = -(\hat{W}^\top(t)\Phi(x(t), u(t)) - 10(\hat{x}(t) - x(t)) + 2(\hat{x}(t) - \mathrm{r}(t)) - \dot{\mathrm{r}}(t))$$

Example 3: Simulation Data









Design Example: F-16 Adaptive Pitch Rate Tracker

Aircraft Data Short-Period Dynamics

- **Trim conditions**
  - CG = 35%, Alt = 0 ft, QBAR = 300 psf, V_T = 502 fps, AOA = 2.1 deg
- **Nominal system**
  - statically unstable
  - open-loop dynamically stable, (2 real negative eigenvalues)
- **Control architecture**
  - baseline / nominal controller
    - LQR pitch tracking design
  - direct adaptive model following augmentation
- **Simulated failures**
  - elevator control effectiveness: 50% reduction
  - battle damage instability
    - static instability: 150% increase
    - pitch damping: 80% reduction
  - pitching moment modeling nonlinear uncertainty

LQR PI Baseline Controller

- Using LQR PI state feedback design
  - nominal values for stability & control derivatives
  - pitch rate step-input command
  - no uncertainties, no control failures
  - system dynamics:
  - "wiggle" system in matrix form

$$\begin{cases} \dot{e}_q^I = q - q^{cmd} \\ \dot{\alpha} = \dfrac{Z_\alpha}{V}\alpha + q + \dfrac{Z_\delta}{V}\delta_e \\ \dot{q} = M_\alpha \alpha + M_q q + M_\delta \delta_e \end{cases}$$

$$\underbrace{\begin{pmatrix} \dot{e}_q \\ \ddot{\alpha} \\ \ddot{q} \end{pmatrix}}_{\dot{\tilde{x}}} = \underbrace{\begin{pmatrix} 0 & 0 & 1 \\ 0 & \dfrac{Z_\alpha}{V} & 1 \\ 0 & M_\alpha & M_q \end{pmatrix}}_{\tilde{A}} \underbrace{\begin{pmatrix} e_q \\ \dot{\alpha} \\ \dot{q} \end{pmatrix}}_{\tilde{x}} + \underbrace{\begin{pmatrix} 0 \\ \dfrac{Z_\delta}{V} \\ M_\delta \end{pmatrix}}_{\tilde{B}} \underbrace{\dot{\delta}_e}_{\tilde{u}} \quad \Leftrightarrow \quad \dot{\tilde{x}} = \tilde{A}\tilde{x} + \tilde{B}\tilde{u}$$

LQR PI Baseline Controller (continued)

- LQR design for the "wiggle" system
  - Optimal feedback solution: $\boxed{\tilde{u} = -\tilde{K}\,\tilde{x}}$
  - Using original states:

$$\dot{\delta}_e^{bl} = -\begin{pmatrix} K_q^I & K_\alpha & K_q \end{pmatrix}\begin{pmatrix} e_q \\ \dot{\alpha} \\ \dot{q} \end{pmatrix} = -K_q^I\, e_q - K_\alpha\,\dot{\alpha} - K_q\,\dot{q}$$

  - Integration yields LQR PI feedback: $\delta_e^{bl} = -\overbrace{K_x}^{\tilde{K}}\, x$

$$\delta_e^{bl} = -K_q^I\, e_q^I - K_\alpha\,\alpha - K_q\, q \;\Rightarrow\; \delta_e^{bl} = -10\, e_q^I - 3.2433\,\alpha - 10.7432\, q$$

$$\tilde{A} = \begin{pmatrix} 0 & 0 & 1 \\ 0 & -1.0189 & 0.9051 \\ 0 & 0.8223 & -1.0774 \end{pmatrix}, \quad \tilde{B} = \begin{pmatrix} 0 \\ -0.0022 \\ -0.1756 \end{pmatrix}, \quad \tilde{Q} = \begin{pmatrix} 100 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 100 \end{pmatrix}$$

closed-loop eigenvalues

Eigenvalue	Damping	Freq. (rad/s)
-7.97e-001 + 3.45e-001i	9.18e-001	8.68e-001
-7.97e-001 - 3.45e-001i	9.18e-001	8.68e-001
-2.39e+000	1.00e+000	2.39e+000

Short-Period Dynamics with Uncertainties

- **System:**

$$\underbrace{\begin{pmatrix} \dot{e}_q^I \\ \dot{\alpha} \\ \dot{q} \end{pmatrix}}_{\dot{x}} = \underbrace{\begin{pmatrix} 0 & 0 & 1 \\ 0 & \dfrac{Z_\alpha}{V} & 1 \\ 0 & M_\alpha & M_q \end{pmatrix}}_{\tilde{A}} \underbrace{\begin{pmatrix} e_q^I \\ \alpha \\ q \end{pmatrix}}_{x} + \underbrace{\begin{pmatrix} 0 \\ \dfrac{Z_\delta}{V} \\ M_\delta \end{pmatrix}}_{B_1 = \tilde{B}} \Lambda\left(\delta_e + K_0(\alpha, q)\right) + \underbrace{\begin{pmatrix} -1 \\ 0 \\ 0 \end{pmatrix}}_{B_2} q^{cmd}$$

- **Reference model:**

$$\dot{x} = A\,x + B_1\,\Lambda\left(\delta_e + K_0\left(\alpha, q\right)\right) + B_2\, q^{cmd}$$

  – no uncertainties

  – (Plant + Baseline LQR PI)

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

- **Control Goal**

  – Model following pitch rate tracking:

tracking error vector

$$\left\| x - x_{ref} \right\| \to 0$$

Adaptive Augmentation Design

- Total elevator deflection:

$$\boxed{\delta_e = \delta_e^{bl} + \delta_e^{ad} = \underbrace{K_q^I\, e_q^I + K_\alpha\, \alpha + K_q\, q}_{\delta_e^{bl}} + \underbrace{\hat{k}_q^I\, e_q^I + \hat{k}_\alpha\, \alpha + \hat{k}_q\, q - \overbrace{\hat{\Theta}^T\, \Phi(\alpha, q)}^{\hat{K}_0(\alpha,q)}}_{\delta_e^{ad}}}$$

$$\Updownarrow$$

$$\boxed{\delta_e = \left(K_x + \hat{k}_x\right)^T x - \hat{\Theta}^T\, \Phi(\alpha, q)}$$

- Adaptive laws:

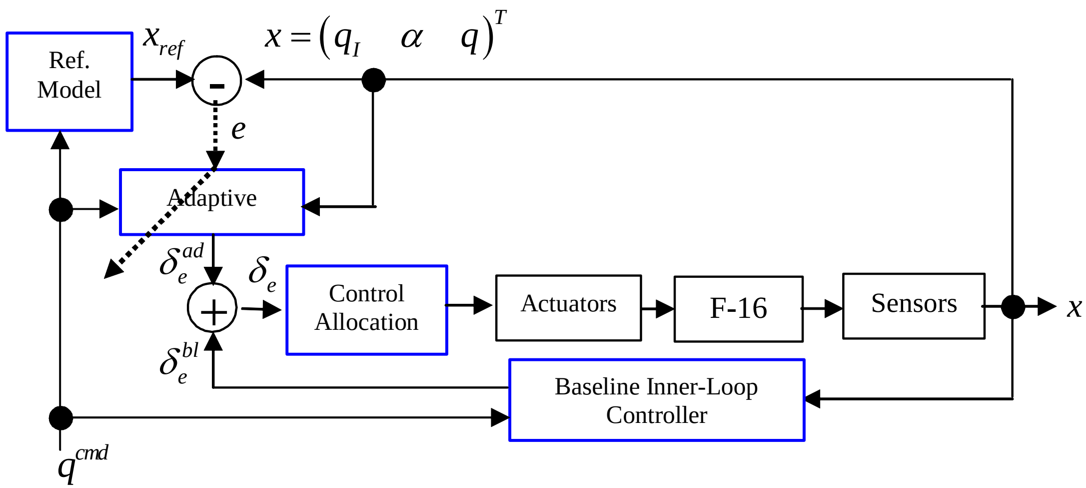
$$\boxed{\begin{cases} \dot{\hat{k}}_x = \Gamma_x\, \mathrm{Proj}\left(\hat{k}_x,\, -x\, e^T\, P\, B_1\right) \\ \dot{\hat{\Theta}} = \Gamma_\Theta\, \mathrm{Proj}\left(\hat{\Theta},\, \Phi\left(x_p\right) e^T\, P\, B_1\right) \end{cases}} \Leftrightarrow \boxed{\begin{cases} \begin{pmatrix} \dot{\hat{k}}_\alpha \\ \dot{\hat{k}}_q \\ \dot{\hat{k}}_q^I \end{pmatrix} = \Gamma_x\, \mathrm{Proj}\left(\begin{pmatrix} \hat{k}_\alpha \\ \hat{k}_q \\ \hat{k}_q^I \end{pmatrix},\, -\begin{pmatrix} \alpha \\ q \\ q_I \end{pmatrix} \begin{pmatrix} q_I - q_I^{ref} & \alpha - \alpha_{ref} & q - q_{ref} \end{pmatrix} P \begin{pmatrix} 0 \\ \frac{Z_\delta}{V} \\ M_\delta \end{pmatrix} \right) \\ \dot{\hat{\Theta}} = \Gamma_\Theta\, \mathrm{Proj}\left(\hat{\Theta},\, \Phi(\alpha, q) \begin{pmatrix} q_I - q_I^{ref} & \alpha - \alpha_{ref} & q - q_{ref} \end{pmatrix} P \begin{pmatrix} 0 \\ \frac{Z_\delta}{V} \\ M_\delta \end{pmatrix} \right) \end{cases}}$$

Adaptive Augmentation Design (continued)

- **Free design parameters**
  – symmetric positive definite matrices: $\boxed{\left(Q,\ \Gamma_x,\ \Gamma_\Theta\right)}$
- **Need to solve algebraic Lyapunov equation**

$$\boxed{P\,A_{ref} + A_{ref}^T\,P = -Q}$$

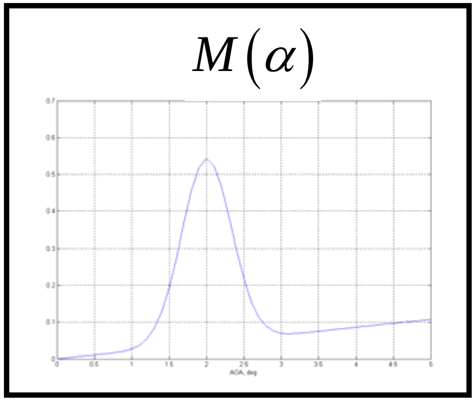
- Using **Dead-Zone** modification and **Projection Operator**



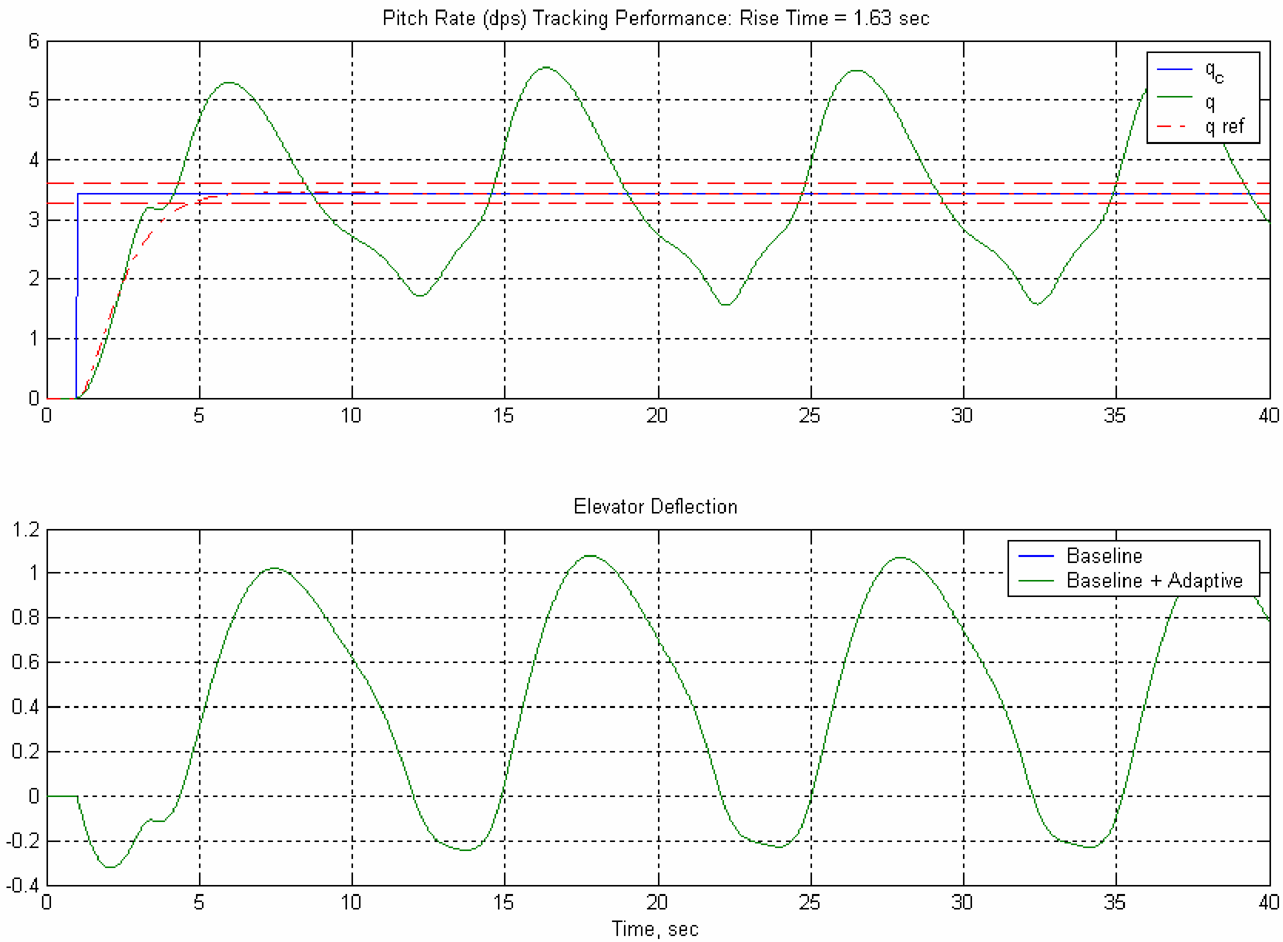
Adaptive Design Data

- **Design parameters**
  - using 11 RBF functions: $\phi_i = e^{-\frac{(\alpha-\alpha_i)^2}{\sigma^2}}, \quad \alpha_i \in [-10:.1:10]$
  - Rates of adaptation: $\Gamma_x = 0, \quad \Gamma_\Theta = 1$
  - Solving Lyapunov equation with: $Q = \mathrm{diag}([0 \quad 1 \quad 800])$
- **Zero initial conditions**
- **Pitch rate command input**
- **System Uncertainties**
  - 50% elevator effectiveness failure, $(0.5 * M_\delta^{bl})$
  - 50% increase in static instability, $(1.5 * M_\alpha^{bl})$
  - 80% decrease in pitch damping, $(0.2 * M_q^{bl})$
  - nonlinear pitching moment

$$M(\alpha) = 1.5 * M_\alpha^{bl} + e^{-\frac{\left(\alpha - \frac{2\pi}{180}\right)^2}{0.0116^2}}$$

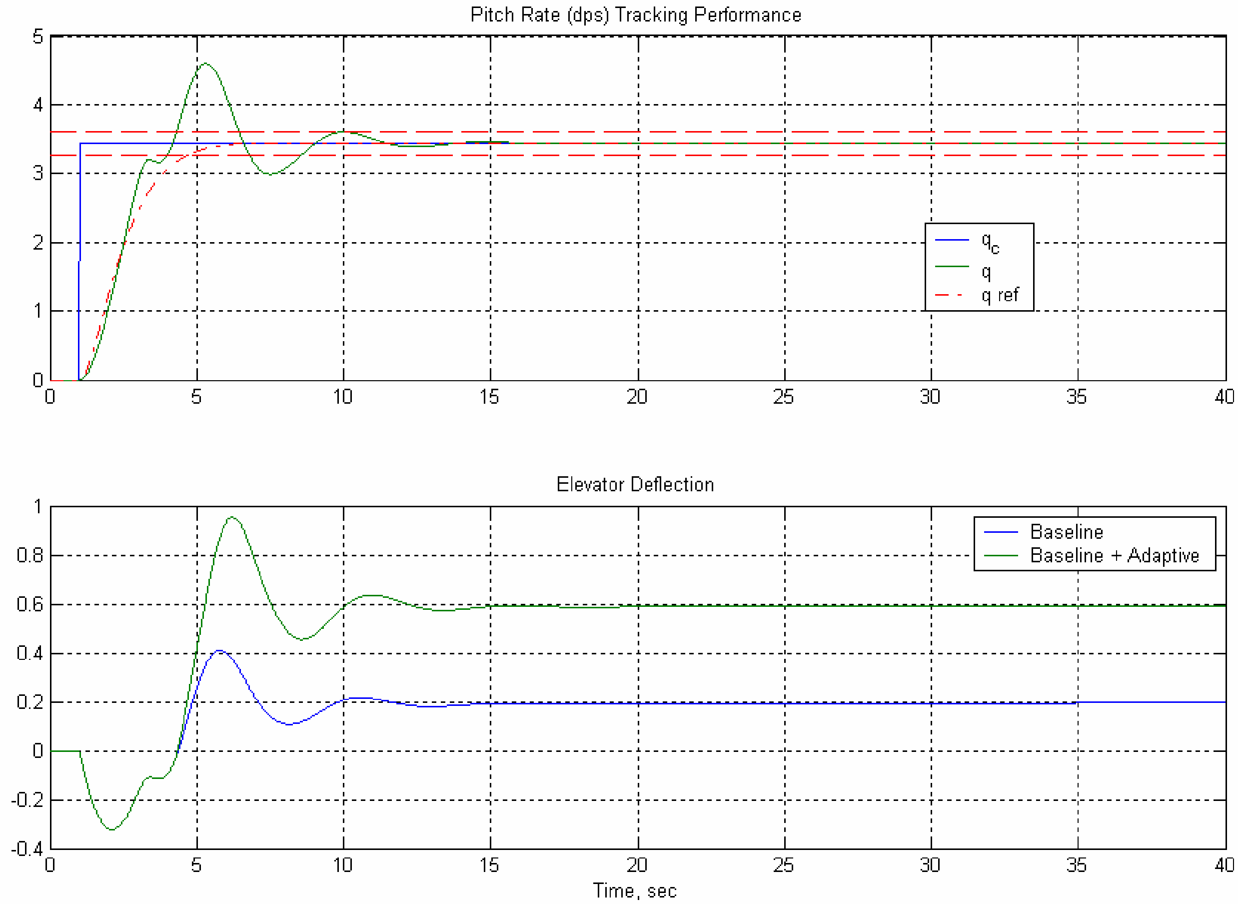


LQR PI: Tracking Step-Input Command



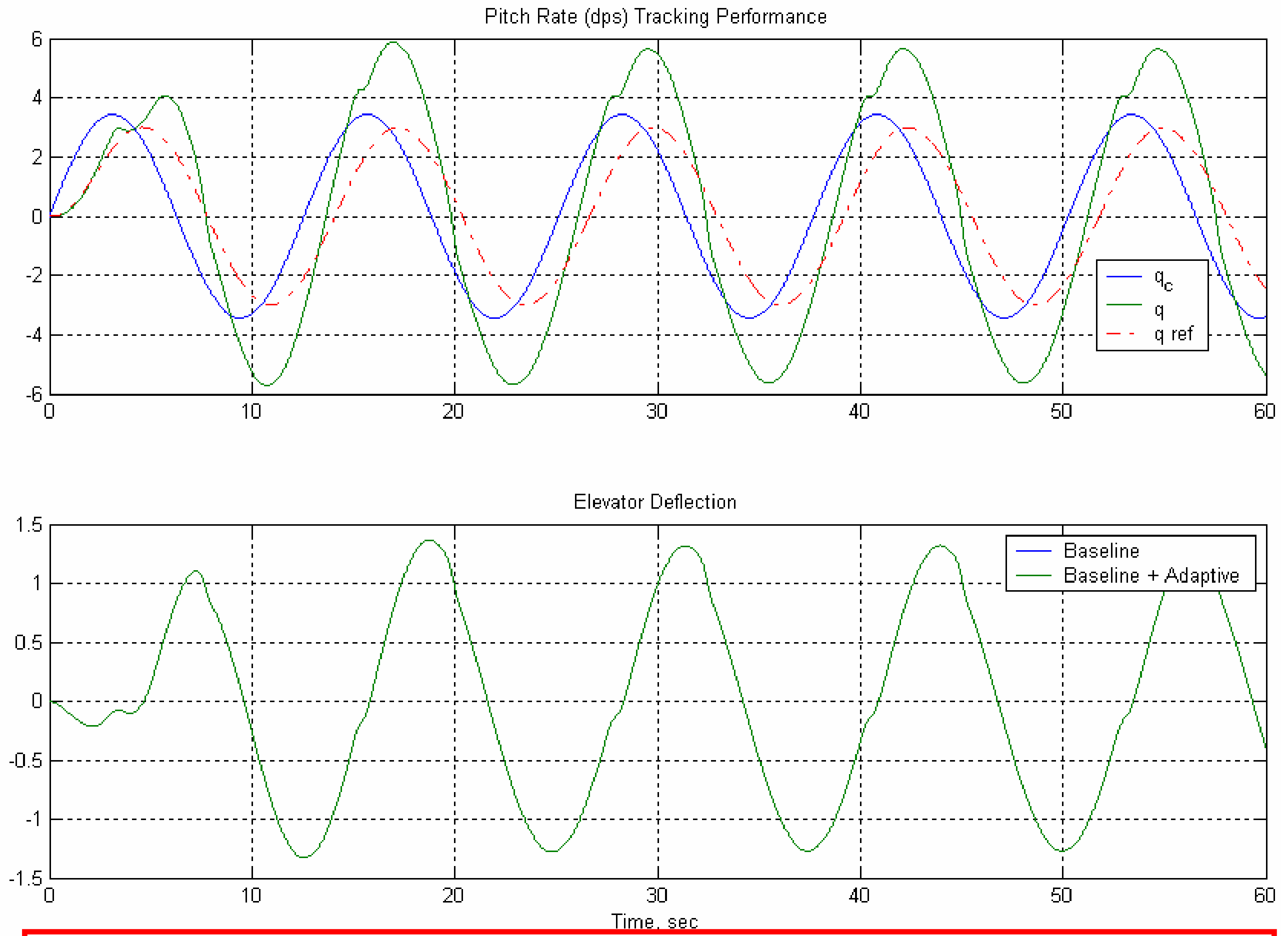
Unstable Dynamics due to Uncertainties

LQR PI + Adaptive: Tracking Step-Input Command



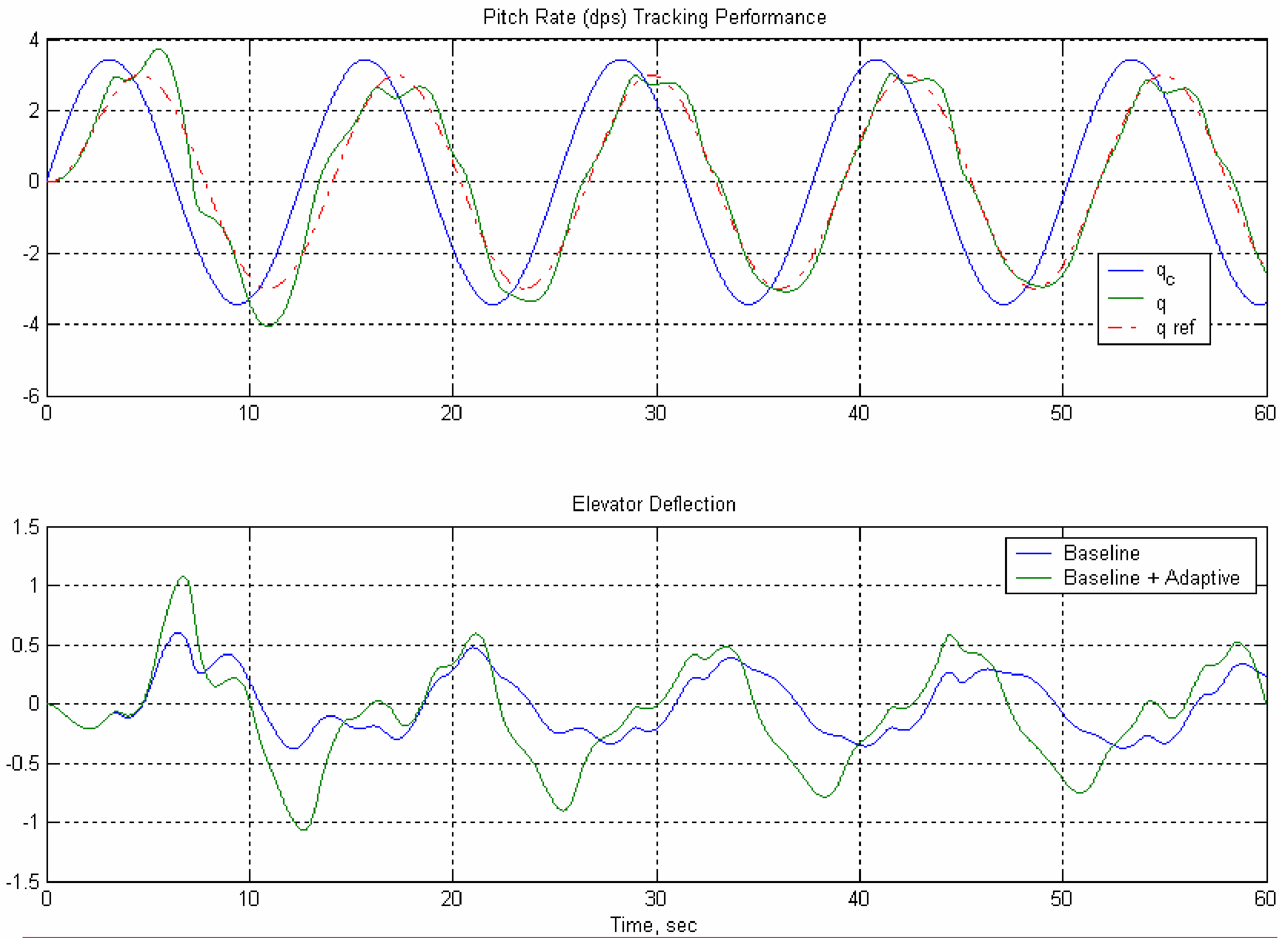
Adaptive Augmentation yields Bounded Stable Tracking in the Presence of Uncertainties

LQR PI: Tracking Sinusoidal Input with Uncertainties



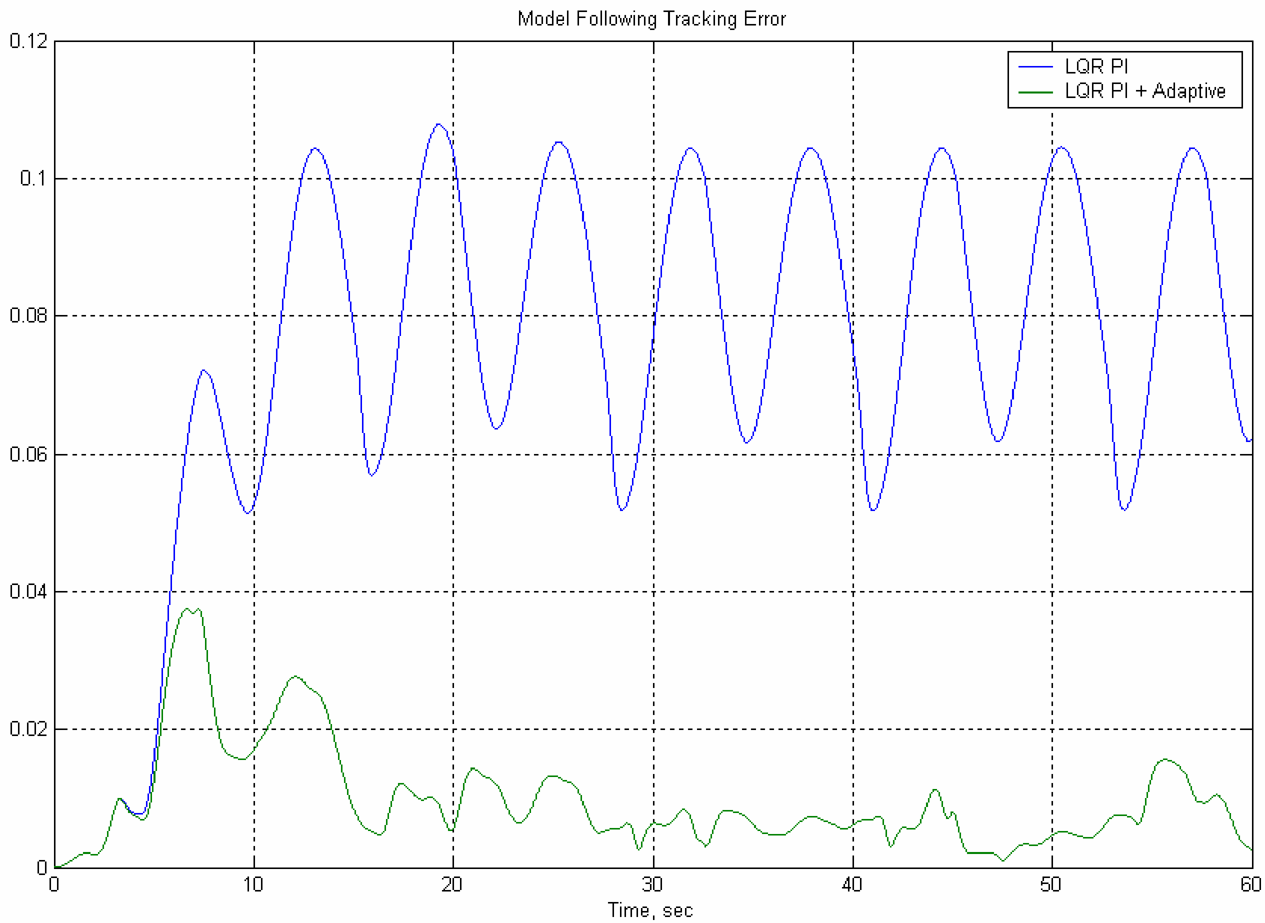
LQR PI Tracking Performance Degradation in the Presence of Uncertainties

LQR PI + Adaptive: Tracking Sinusoidal Input with Uncertainties



Adaptive Augmentation Recovers Target Tracking Dynamics in the Presence of Uncertainties

Model Following Tracking Error Comparison



Adaptive Augmentation yields Significant Reduction in Tracking Error Magnitude

Adaptive Design Comments

- **RBF NN adaptation dynamics**

$$\boxed{\dot{\hat{\Theta}}_i = \left(\Gamma_\Theta\right)_{ii} \Phi_i(\alpha, q)\left(k_{1i}\left(q_I - q_I^{ref}\right) + k_{2i}\left(\alpha - \alpha_{ref}\right) + k_{3i}\left(q - q_{ref}\right)\right)}$$

- **Fixed RBF NN gains**

$$\boxed{\begin{pmatrix} k_{1i} \\ k_{2i} \\ k_{3i} \end{pmatrix} = P \begin{pmatrix} 0 \\ \dfrac{Z_\delta}{V} \\ M_\delta \end{pmatrix} \left(\Gamma_\Theta\right)_{ii}}$$

  – simulation data

$$\boxed{k_{1i} = 0, \quad k_{2i} = -1.1266, \quad k_{3i} = -24.0516}$$

- **Projection Operator**
  – keeps parameters bounded
  – nonlinear extension of anti-windup integrator logic
- **Dead-Zone modification**
  – freezes adaptation process if: $\boxed{\left\| x - x_{ref} \right\| \le \varepsilon}$

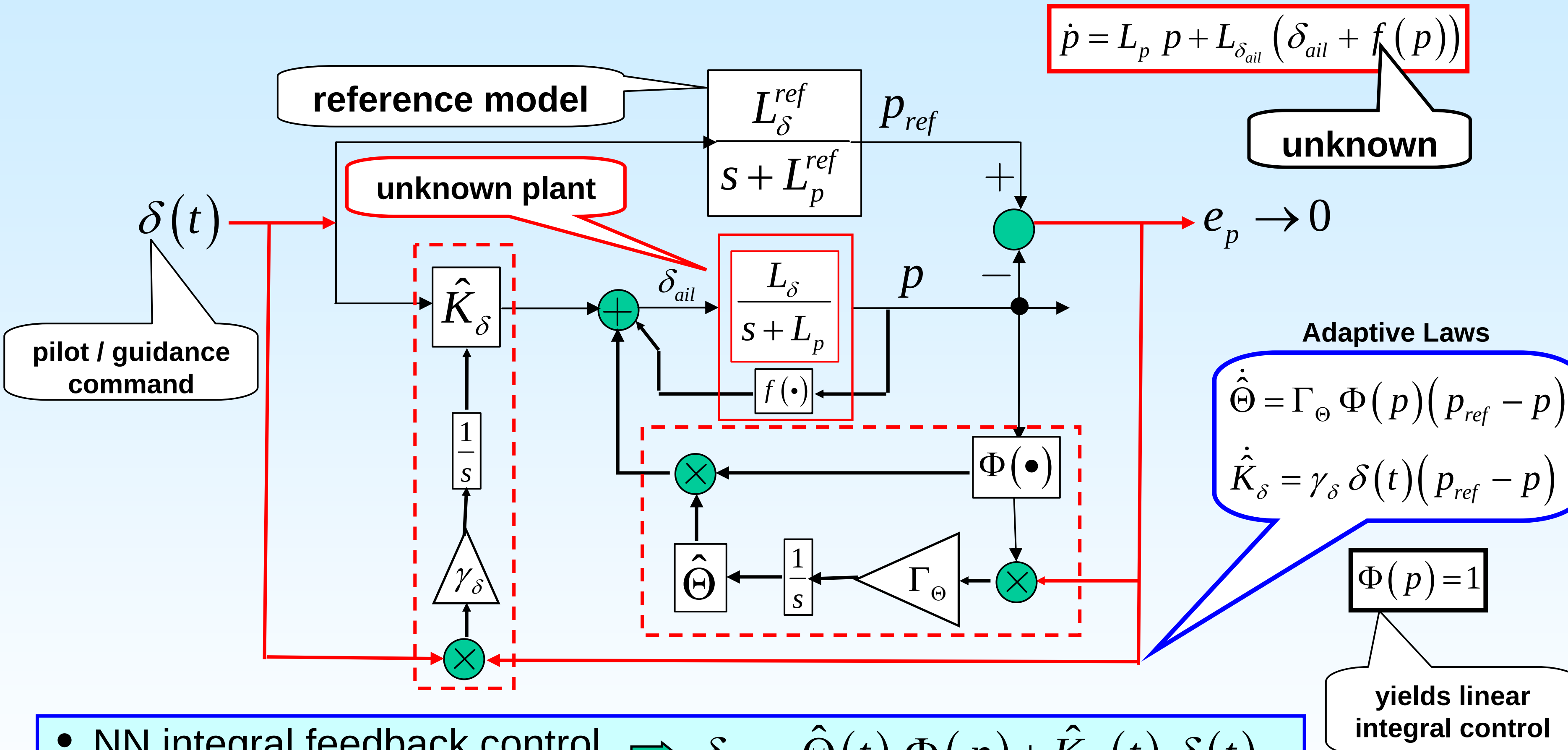
  dead-zone tolerance

  – separates adaptive augmentation from baseline controller

Open Problem in Adaptive Control

Stability / Robustness Margins

Adaptive NeuroControl = Nonlinear Integral Control



- NN integral feedback control $\Rightarrow$ $\delta_{ail} = \hat{\Theta}(t)\, \Phi(p) + \hat{K}_\delta(t)\, \delta(t)$
- Rates of adaptation $\Rightarrow$ $(\Gamma_\Theta,\ \gamma_\delta)$
- Time-delay sensitivity (phase margin) is an open problem

"State of the Art" in Adaptive Control

- **Know how**
  - design robust stable AFCS
  - provide adequate bounded tracking in the presence of unknown-unknowns
    - modeling uncertainties
    - control failures
    - environmental disturbances

- **Do not know how,** (i.e., using simulation trade off studies and Monte Carlo analysis)
  - choose free design parameters to yield oscillations free transient responses
    - adaptive gains must be large enough but not too large
    - Lyapunov solution matrix considerations
  - quantify and compute stability / robustness margins

- **Adaptive Control V&V Technical Challenge**
  - Need theoretically justified and numerically verifiable AFCS V&V technologies
    - provide standard process against which adaptive systems can be certified
    - offer guidelines during early design cycle of such systems

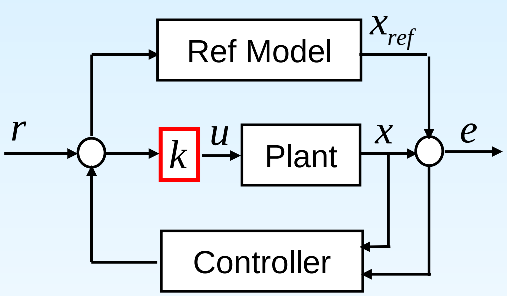
V&V Challenge 1: Gain Margin

- **Ideal Adaptive Control Command**

$$\dot{\hat{\Theta}}(t) = \Gamma_{\Theta}\,\mathrm{Proj}\left(\hat{\Theta}(t),\ \Phi(x(t))\left(x(t) - x_{ref}(t)\right)^T P B\right)$$
$$u(t) = \hat{\Theta}^T(t)\,\Phi(x(t))$$

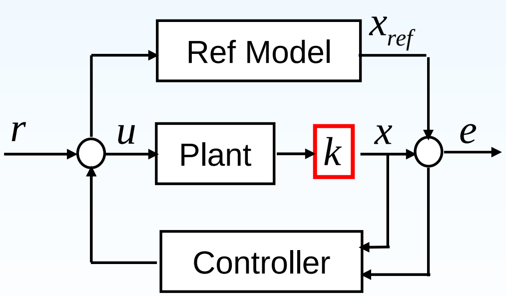
- **Breaking loop at input**

$$\dot{\hat{\Theta}}(t) = \Gamma_{\Theta}\,\mathrm{Proj}\left(\hat{\Theta}(t),\ \Phi(x)\left(x(t) - x_{ref}(t)\right)^T P B\right)$$
$$u(t) = k\,\hat{\Theta}^T(t)\,\Phi(x(t))$$



- **Breaking loop at output**

$$\dot{\hat{\Theta}}(t) = \Gamma_{\Theta}\,\mathrm{Proj}\left(\hat{\Theta}(t),\ \Phi(k\,x(t))\left(k\,x(t) - x_{ref}(t)\right)^T P B\right)$$
$$u(t) = \hat{\Theta}^T(t)\,\Phi(k\,x(t))$$



Problem: Find min and max *k* such that closed-loop stability and tracking are preserved

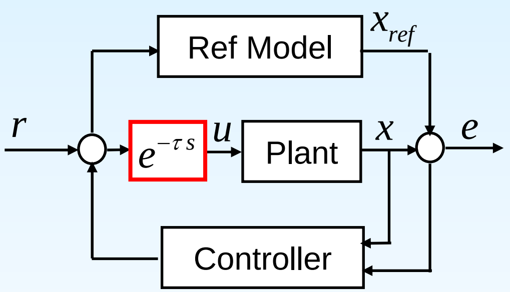
V&V Challenge 2: Time–Delay Margin
(phase margin equivalent)

- **Ideal Adaptive Control Command**

$$\dot{\hat{\Theta}}(t) = \Gamma_\Theta \operatorname{Proj}\left(\hat{\Theta}(t),\ \Phi(x(t))(x(t) - x_{ref}(t))^T P B\right)$$
$$u(t) = \hat{\Theta}^T(t)\Phi(x(t))$$

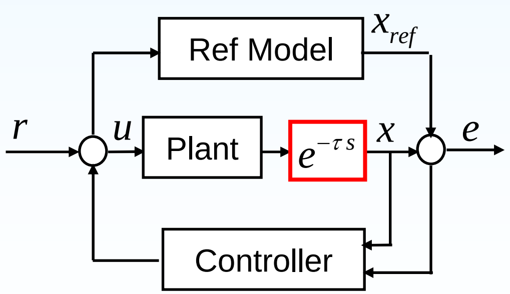
- **Breaking loop at input**

$$\dot{\hat{\Theta}}(t) = \Gamma_\Theta \operatorname{Proj}\left(\hat{\Theta}(t),\ \Phi(x(t))(x(t) - x_{ref}(t))^T P B\right)$$
$$u(t) = \hat{\Theta}^T(t-\tau)\Phi(x(t-\tau))$$



- **Breaking loop at output**

$$\dot{\hat{\Theta}}(t) = \Gamma_\Theta \operatorname{Proj}\left(\hat{\Theta}(t),\ \Phi(x(t-\tau))(x(t-\tau) - x_{ref}(t))^T P B\right)$$
$$u(t) = \hat{\Theta}^T(t)\Phi(x(t-\tau))$$



Problem: Find max τ such that closed-loop stability and tracking are preserved

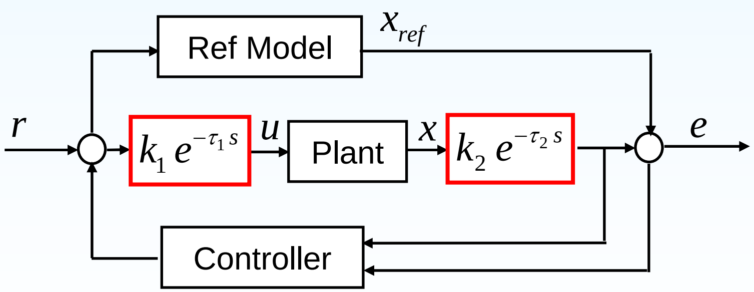
The Main Challenge: Stability Margins

- **Ideal Adaptive Control Command**

$$\dot{\hat{\Theta}}(t) = \Gamma_\Theta \,\mathrm{Proj}\left(\hat{\Theta}(t),\ \Phi(x(t))\left(x(t) - x_{ref}(t)\right)^T P\,B\right)$$
$$u(t) = \hat{\Theta}^T(t)\,\Phi(x(t))$$

- **Breaking loop at input-output**

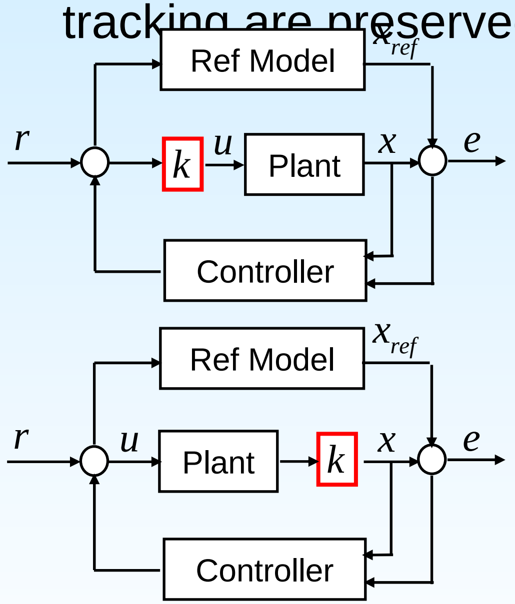
$$\dot{\hat{\Theta}}(t) = \Gamma_\Theta \,\mathrm{Proj}\left(\hat{\Theta}(t),\ \Phi(k_2\,x(t-\tau_2))\left(k_2\,x(t-\tau_2) - x_{ref}(t)\right)^T P\,B\right)$$
$$u(t) = k_1\,\hat{\Theta}^T(t-\tau_1)\,\Phi(k_2\,x(t-\tau_1-\tau_2))$$



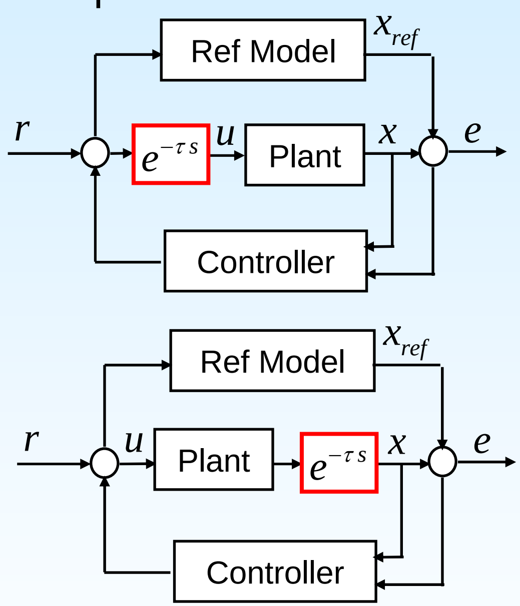
Problem: Find min / max boundaries for k_1, k_2, τ_1 and τ_2 such that closed-loop stability and tracking are preserved

At a Glance: Gain and Time-Delay Margins

- **Gain Margin**
  - Find min and max k such that closed-loop stability and tracking are preserved

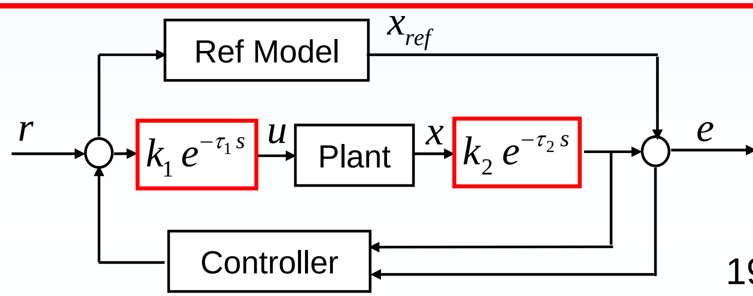


- **Time-Delay Margin**
  - Find max τ such that closed-loop stability and tracking are preserved



Gain and Time-Delay Margins:

Find min / max boundaries for k_1, k_2, τ_1 and τ_2 such that closed-loop stability and tracking are preserved



V&V of Adaptive Control Challenge

- **Goal: Define and compute Gain and Time-Delay margins**
  - extend classical notations of gain and phase margins for nonlinear systems operating under adaptive control and in the presence of uncertainties
  - need sufficient (at least) conditions for maintaining desired closed-loop system performance in the presence of unknown-unknowns both <u>in transient and steady state</u>
    - measure how far is the system dynamics from undesirable oscillations
    - estimate stability / robustness bounds on uncertainties and adaptive gains
  - must be theoretically justified and numerically verifiable

ROBUST ADAPTIVE CONTROL OF UNMANNED COMBAT AIR VEHICLES

Eugene Lavretsky and Kevin Wise

FA9550-04-C-0047

The Boeing Company – Phantom Works
Huntington Beach, CA 92647-2099
Saint Louis, MO 63166-0516

Presented at 2005 AFOSR Joint Program Review
Long Beach, CA

Presentation Overview

- **Theoretical Background**
  - (Baseline + Adaptive) – control design for cascaded systems
    - using Dynamic Inversion controller
    - using LQR PI controller
    - adaptive design modifications to enforce robustness
- **Tech Transitions, Applications and Flight Test Results**
- **Open Problems**
- **Summary**

Adaptive Augmentation of a Dynamic Inversion (DI) Controller

(DI + Adaptive)

Control of Uncertain Cascaded Systems

- **2nd order system:**

$$\boxed{\dim x_1 = \dim x_2 = \dim \dot{x}_2^{cmd} = 1}$$

controlled output

unknown

$$\begin{cases} \dot{x}_1 = F_1^0(x_1, z) + B_1\, x_2 + f_1(x_1, z) \\ \dot{x}_2 = F_2^0(x_1, x_2, z) + \dot{x}_2^{cmd} + f_2(x_1, x_2, z) \end{cases}$$

control input

unknown

- **Control goal**
  - Choose $\dot{x}_2^{cmd}$ such that:

$$\boxed{x_1(t) \to x_1^{com}(t), \quad \text{as } t \to \infty}$$

- **Assumptions**
  - system dimensions:
  - F_i^0 are known
  - f_i are unknown
  - $\det B_1 \neq 0$ does not cross zero
- **Example: Short period dynamics**
  - Control goal: AOA tracking

$$\boxed{\alpha \to \alpha_c, \text{ as } t \to \infty}$$

$$\begin{cases} \dot{\alpha} = \underbrace{-L_\alpha(\alpha, M, h)\overbrace{\alpha}^{x_1}}_{F_1} + \underbrace{1}_{B_1}\overbrace{q}^{x_2} + \underbrace{f_\alpha(\alpha, M, h)}_{\text{unknown}} \\ \dot{q} = \underbrace{M_0\left(\alpha, q, \overbrace{M, h}^{z}\right)}_{F_2} + \underbrace{\dot{q}_{cmd}}_{\dot{x}_2^{cmd}} + \underbrace{f_q(\alpha, q, M, h)}_{\text{unknown}} \end{cases}$$

Total Control Command = Baseline DI + Adaptive

- **Ideal Solution: Augmentation of Dynamic Inversion (DI) based Proportional + Integral + Derivative (PID) Control**

$$\dot{x}_2^{cmd} = B_1^{-1}\left(\ddot{x}_1^m - f\left(x_1, x_2, z, \dot{z}\right) - K_D\left(\dot{x}_1 - \dot{x}_1^m\right) - K_P\left(x_1 - x_1^m\right) - K_I\left(\frac{x_1(t) - x_1^m(t)}{s}\right) - v\right)$$

known function

not known due to uncertainties

adaptive augmentation

- **State Derivative Approximation:** $\dot{\hat{x}}_1 \triangleq F_1^0\left(x_1, z\right) + B_1 x_2$
- **(PID + Adaptive) – Controller:**

$$\dot{x}_2^{cmd} = B_1^{-1}\left(\ddot{x}_1^m - f\left(x_1, x_2, z, \dot{z}\right) - K_D\left(\dot{\hat{x}}_1 - \dot{x}_1^m\right) - K_P\left(x_1 - x_1^m\right) - K_I\left(\frac{x_1(t) - x_1^m(t)}{s}\right) - v\right)$$

- **On-line function approximator / dominator**

$$v \triangleq \hat{D}\left(x_1, x_2, z, \dot{z}\right) = \hat{\theta}_D^T \underbrace{\Phi_D\left(x_1, x_2, z, \dot{z}\right)}_{\text{Regressor Vector}}$$

Adaptive Augmentation

- **Nominal PID feedback gains**

$$\begin{cases} K_D = 2\xi\omega + k_1 \\ K_P = \omega(\omega + 2\xi k_1) \\ K_I = \omega^2 k_1 \end{cases} \Rightarrow \left(\lambda^2 + 2\xi\omega\lambda + \omega^2\right)\left(\lambda + k_1\right) = 0$$

closed-loop characteristic polynomial, (ideal model)

- **Closed-loop tracking error dynamics**

adaptive increment

$$\ddot{e}_1 = -K_D\,\dot{e}_1 - K_P\,e_1 - K_I\int_0^t e_1(\tau)\,d\tau + \underbrace{D(x_1, x_2, z, \dot{z})}_{\text{unknown}} - v$$

$$\Updownarrow$$

approximation error

$$\ddot{e}_1 = -(2\xi\omega + k_1)\dot{e}_1 - \omega(\omega + 2\xi k_1)e_1 - \omega^2 k_1\int_0^t e_1(\tau)\,d\tau - e_D$$

$$\Updownarrow$$

$$\ddot{e}_1 + k_1\,\dot{e}_1 = -2\xi\omega(\dot{e}_1 + k_1 e_1) - \omega^2\left(e_1 + k_1\int_0^t e_1(\tau)\,d\tau\right) - e_D$$

Introducing Filtered Tracking Error

- **Filtered Tracking error signal**

$$\boxed{e_1^f = e_1 + k_1 \int_0^t e_1(\tau)\,d\tau} \Leftrightarrow \boxed{e_1^f = \frac{(s+k_1)}{s} e_1}$$

- **Filtered tracking error dynamics**

$$\boxed{\ddot{e}_1^f = -2\xi\omega\dot{e}_1^f - \omega^2 e_1^f - e_D} \Leftrightarrow \boxed{\underbrace{\begin{pmatrix} \dot{e}_1^f \\ \ddot{e}_1^f \end{pmatrix}}_{\dot{e}_f} = \underbrace{\begin{pmatrix} 0 & 1 \\ -\omega^2 & -2\xi\omega \end{pmatrix}}_{A_{ref}} \underbrace{\begin{pmatrix} e_1^f \\ \dot{e}_1^f \end{pmatrix}}_{e_f} - \underbrace{\begin{pmatrix} 0 \\ 1 \end{pmatrix}}_{b_{ref}} e_D}$$

- **Main asymptotic property**

$$\boxed{\left[e_1^f \to 0 \wedge \dot{e}_1^f \to 0\right] \Rightarrow \left[e_1 \to 0\right]}$$

- **Main idea**
  - control filtered tracking error instead of the original one

Total Control = Baseline DI + Adaptive

- **(Baseline DI + Adaptive) – Controller**

$$\boxed{\begin{aligned}\dot{x}_2^{cmd} &= \underbrace{B_1^{-1}\left(\ddot{x}_1^m - f\left(x_1, x_2, z, \dot{z}\right) - K_D\left(\dot{\hat{x}}_1 - \dot{x}_1^m\right) - K_P\left(x_1 - x_1^m\right) - K_I\left(\frac{x_1 - x_1^m}{s}\right)\right)}_{\text{Dynamic Inversion based PID Nominal Controller}} \\ &\quad \underbrace{- B_1^{-1}\,\hat{\theta}_D^T\,\Phi_D\left(x_1, x_2, z, \dot{z}\right)}_{\text{Adaptive Augmentation}}\end{aligned}}$$

- **Regressor Vector Φ Components**
  - RBF-s
  - linear in $\boxed{\left(x_1, x_2, z, \dot{z}\right)}$
  - includes constant / bias

Parameter Adaptation

- **Adaptive Law**

$$\dot{\hat{\theta}}_D = \Gamma_D \ \mathrm{Proj}\left(\hat{\theta}_D, \ \Phi_D \, e_f^T \, P \, b_{ref}\right)$$

  - Projection Operator
    - provides bounded parameter adaptation
  - Dead-Zone Modification
    - robustness to noise and nominal / adaptive signals separation
- **Simplification**

$$e_f^T \, P \, b_{ref} = \begin{pmatrix} e_1^f & \dot{e}_1^f \end{pmatrix} \begin{pmatrix} p_{11} & p_{12} \\ p_{12} & p_{22} \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} = p_{12} \, e_1^f + p_{22} \, \dot{e}_1^f$$

$$\begin{cases} q_{11} = 0 \\ p_{11} = \dfrac{\omega}{2\xi} q_{22} \\ p_{12} = 0 \\ p_{22} = \dfrac{1}{2\xi\omega} q_{22} \end{cases} \Rightarrow \quad e_f^T \, P \, b_{ref} = p_{22} \, \dot{e}_1^f \quad \Rightarrow \quad \dot{\hat{\theta}}_D = \Gamma_D \ \mathrm{Proj}\left(\hat{\theta}_D, \ \Phi_D \, p_{22} \, \dot{e}_1^f\right)$$

Adaptive Augmentation of an Optimal Linear Controller with Proportional Integral Action

(LQR PI + Adaptive)

LQR PI Baseline Control with Projection

Feedback Gains Designed Using
Optimal Control + Projection Theory

Optimal Robust Servomechanism Linear Quadratic Regulator (RSLQR)

Model: $\dot{z} = \tilde{A}z + \tilde{B}\mu \qquad z \in R^n$

ARE: $\tilde{A}^T P + P\tilde{A} + Q - P\tilde{B}R^{-1}\tilde{B}^T P = 0$

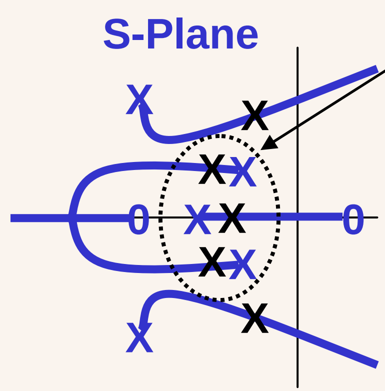
CLAW: $u = -K_{SF}x = -R^{-1}B^T P x$

Augment Dynamics With Integral Control For Perfect Command Tracking

- **AUTOGAIN Tunes LQR Parameters**
- **Convergence Criteria Focus On Stability/Actuator Rates**
- **LQR Design Charts Describe Tuning Process**

- **Preserve Excellent Stability Properties Of State Feedback Using Output Feedback**
- **Eliminates Sensor H/W Required For State Feedback**

Optimal Projection To Output Feedback Architecture



- **Select Dominant Eigenstructure** (Λ_r, X_r), $r<n$
- **Project Gains (Static)**

  $K = K_{SF}X_r(CX_r)^{-1}$

  $u = -Ky$
- **Analyze Output Feedback Design**
- **Iterate LQR To Achieve Desired Bandwidth**

Generalized (Plant + Baseline A/P) <u>Open-loop</u> Dynamics

$$\begin{cases} \dot{x}_p = A_p\, x_p + B_p\, \Lambda\left(u + f\left(x_p\right)\right) \\ \dot{x}_c = A_c\, x_c + B_{1c}\, r(t) + B_{2c}\, F\, y \\ y = C_p\, x_p + D_p\, \Lambda u \\ u = K_{x_p}^T\, x_p + K_{x_c}^T\, x_c + K_r^T\, r(t) \end{cases}$$

system state

controller state

$\Leftrightarrow$

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

$$y = \underbrace{\begin{pmatrix} C_p & 0 \end{pmatrix}}_{C} x + \underbrace{D_p}_{D}\, u = C\, x + D\, u$$

extended system state

control failures

moment uncertainties
$f(x) = \Theta^T\, \Phi(x) + \varepsilon_f(x)$

$$\boxed{\dot{x} = A\, x + B_1\, \Lambda\left(u + f\left(x_p\right)\right) + B_2\, r(t)}$$

inner-loop commands

guidance commands

Reference Model and Adaptive Control

- **Set uncertainties to zero:** $$\boxed{\Lambda = I_{m\times m}, \quad f\left(x_p\right) = 0_{m\times 1}}$$

- **Use baseline A/P:** $$\boxed{u = u_{bl} = K_x^T\, x + K_r^T\, r\left(t\right)}$$

- **Formulate Closed – Loop System Dynamics**

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

  – **defines <u>nominal</u> closed-loop dynamics achievable under baseline A/P**

  – **forms <u>desired</u> dynamics for adaptive augmentation <u>with uncertainties</u>**

- **Control:** $$\boxed{u = u_{bl} + u_{ad} = \underbrace{K_x^T\, x + K_r^T\, r\left(t\right)}_{\text{Baseline A/P}} + \underbrace{\hat{k}_x^T\, x + \hat{k}_r^T\, r\left(t\right) - \hat{\Theta}^T\, \Phi\left(x_p\right)}_{\text{Adaptive Augmentation}}}$$

$$\boxed{u = \left(K_x + \hat{k}_x\right)^T x + \left(K_r + \hat{k}_r\right)^T r\left(t\right) - \hat{\Theta}^T\, \Phi\left(x_p\right)}$$

Parameter Adaptation

- **Theoretical Basis**
  - **2nd Theorem of Lyapunov**
  - **Barbalat Lemma**
  - **Universal Approximation Property of RBF NN**
- **Adaptive laws yield <u>bounded tracking performance</u> with all signals bounded, in the presence of uncertainties, (UUB)**

$$\boxed{\begin{cases} \dot{\hat{k}}_x = \Gamma_x \operatorname{Proj}\left(\hat{k}_x, -x\, e^T P B_1\right) \\ \dot{\hat{k}}_r = \Gamma_r \operatorname{Proj}\left(\hat{k}_z, -r(t)\, e^T P B_1\right) \\ \dot{\hat{\Theta}} = \Gamma_\Theta \operatorname{Proj}\left(\hat{\Theta}, \Phi\left(x_p\right) e^T P B_1\right) \end{cases}} \Rightarrow \boxed{\lim_{t\to\infty} \left\| x - x_{ref} \right\| \le C}$$

can be made small

  - **using <u>Dead-zone</u> modification, (enforces robustness to noise)**
    - **freezes adaptation if:** $\boxed{\left\| x(t) - x_{ref}(t) \right\| \le \varepsilon}$
  - **using <u>Projection Operator</u>, (bounds adaptive parameters)**
  - **using <u>E – modification</u>, (adds damping and bounds adaptive parameters)**
  - **using <u>μ– modification</u>, (protects against control saturation)**

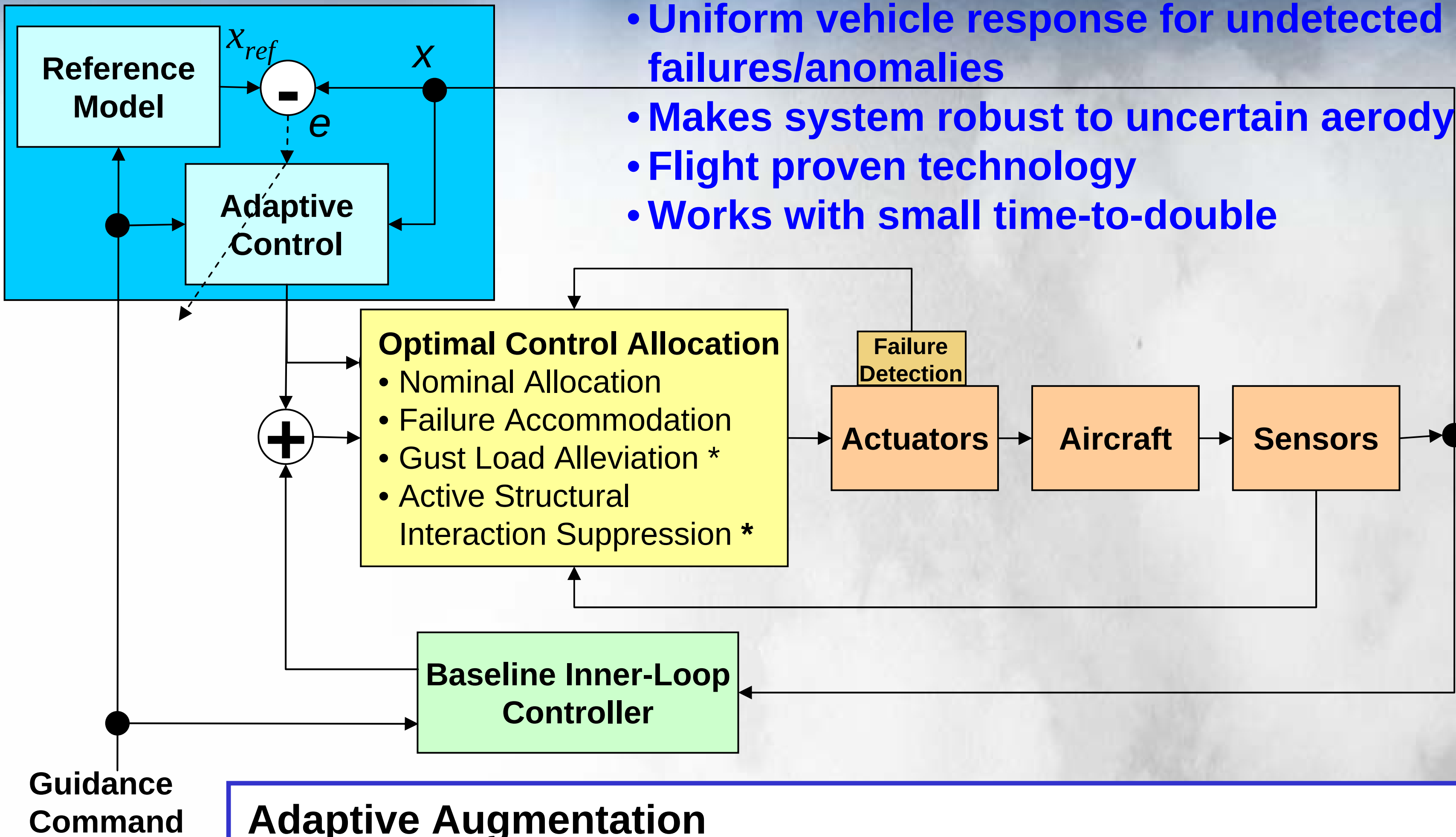
Control Design Summary

- **Total Control Command = Baseline + Adaptive**
  - Baseline autopilot
    - Dynamic Inversion or LQR PI based
    - Provides guidance command tracking
      - assumes nominal aerodynamics
  - Direct adaptive augmentation
    - Adds robustness due to modeling errors and battle damage
    - Explicit model following
    - Provides bounded / local adaptation process
      - Projection Operator, (bounds adaptive parameters)
      - E-modification, (adds damping, bounds adaptive parameters)
      - Dead-Zone modification, (reduces sensitivity to noise)
      - μ – modification, (protects against control saturation)
    - Regressor vector can be chosen to address system specific uncertainties
      - adaptation to off-nominal aerodynamics
      - UUB of all signals

Flight Control Block-Diagram

Flown on X-36, MK-84 JDAM, MK-82 LJDAM

- **Uniform vehicle response for undetected failures/anomalies**
- **Makes system robust to uncertain aerodynamics**
- **Flight proven technology**
- **Works with small time-to-double**



Adaptive Augmentation

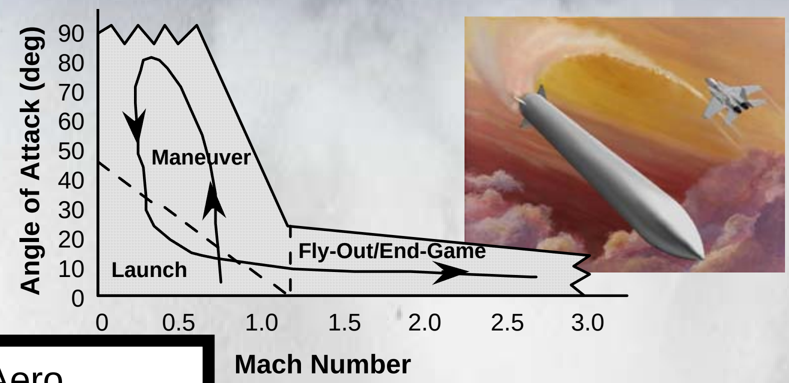
- ***Dead-Zone*** modification prevents adaptation from changing nominal closed-loop dynamics
- ***Projection Operator*** and ***E-mod*** bound adaptive gains
- μ **– mod** protects against control saturation

Flight Control Challenges, Tech Transition, Applications, and Flight Test Results

Adaptive Flight Control Challenges

4th Generation Escape System

Air Superiority Missile Technology



- Nonlinear Aero
- Large Uncertainties
- Nonlinear Control Effectors
- Limited Actuation

X-45A J-UCAS

X-36 Tailless Agility Research Aircraft

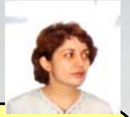
Unstable In Multiple Axes, Non-minimum Phase

Adaptive Control in Development For Advanced Weapon Systems

- Adaptive Control Based upon Earlier Aircraft Application
  - –Extended to Munitions (00-02) with GST
  - –Boeing IRAD Improvements Focus on System ID, Implementation, and Actuator Saturation Issues
  - –Design Retrofits Onto Existing Flight Control Laws
  - –Flight Proven on MK-82 L-JDAM, (04-05)

Adaptive Flight Control

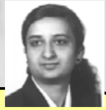
AFOSR Adaptive Control of UCAVs I,II



Boeing Collaborates With Prof. N. Hovakimyan at VaTech on limited actuation

Technology Transition Timeline

93 94 95 96 97 98 99 00 01 02 03 04 05



Boeing funds MIT (Dr. A. Annaswamy) to initiate research in V&V of adaptive systems

Intelligent Flight Control System (NASA/Boeing) F-15 ACTIVE

- Gen I, flown 1999, 2003
- Gen II, 2002 – 2006
  - flight test 4th Q 2005
- Gen III, 2006

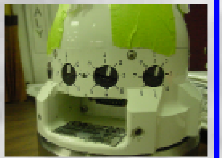
Adaptive Control For Munitions (AFRL-MN/GST//Boeing) MK-84

MK-84 JDAM

Boeing IRAD

Reconfigurable Control For Tailless Fighters (AFRL-VA/Boeing) X-36

MK-82 L-JDAM

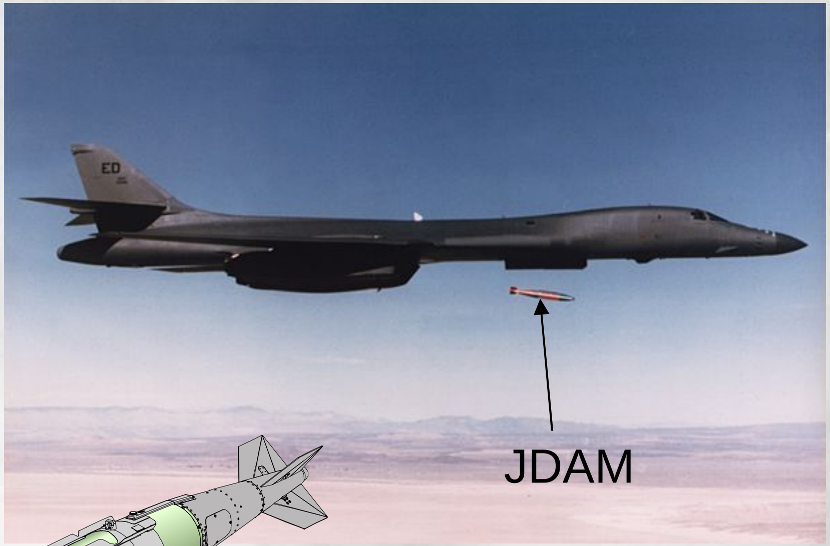


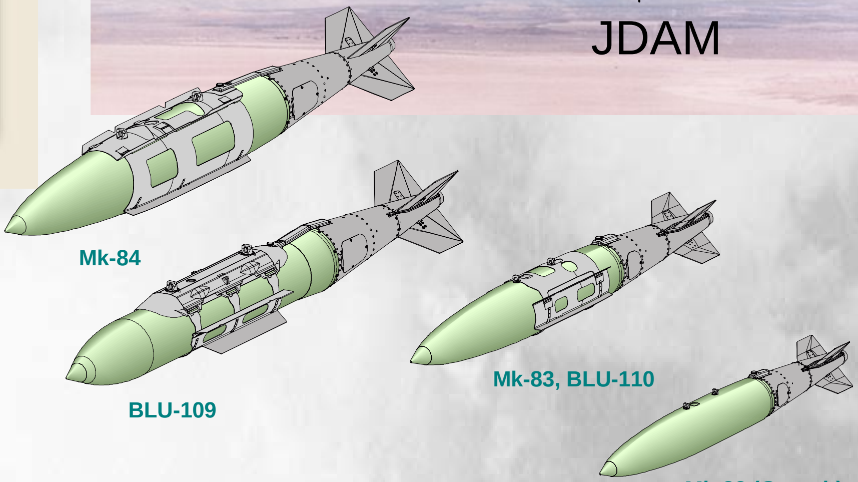
- Ongoing NASA/Boeing IFCS
- Other Transitions

Adaptive Flight Control Applications

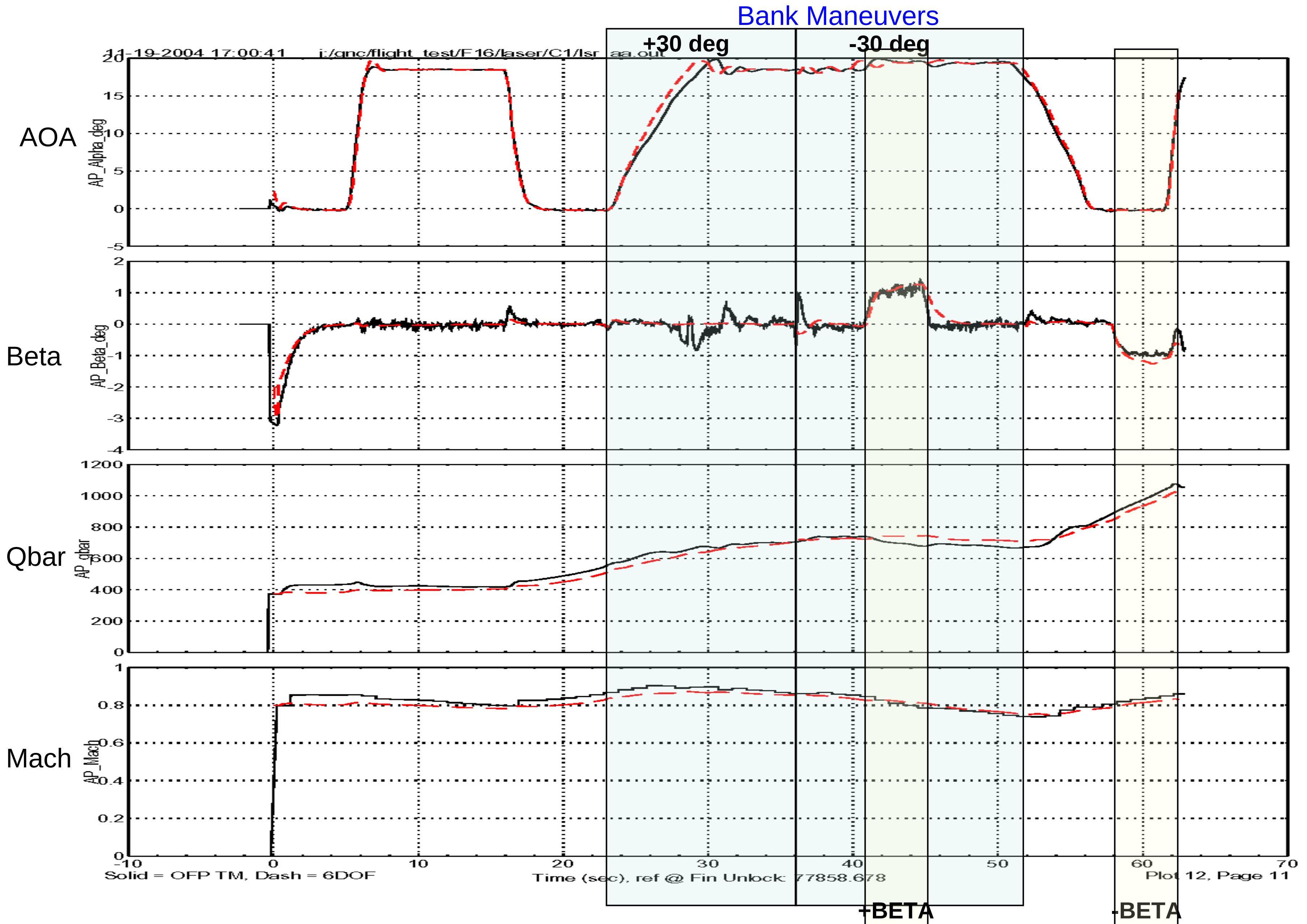
X-36 Tailless Agility Research Aircraft

Joint Direct Attack Munition

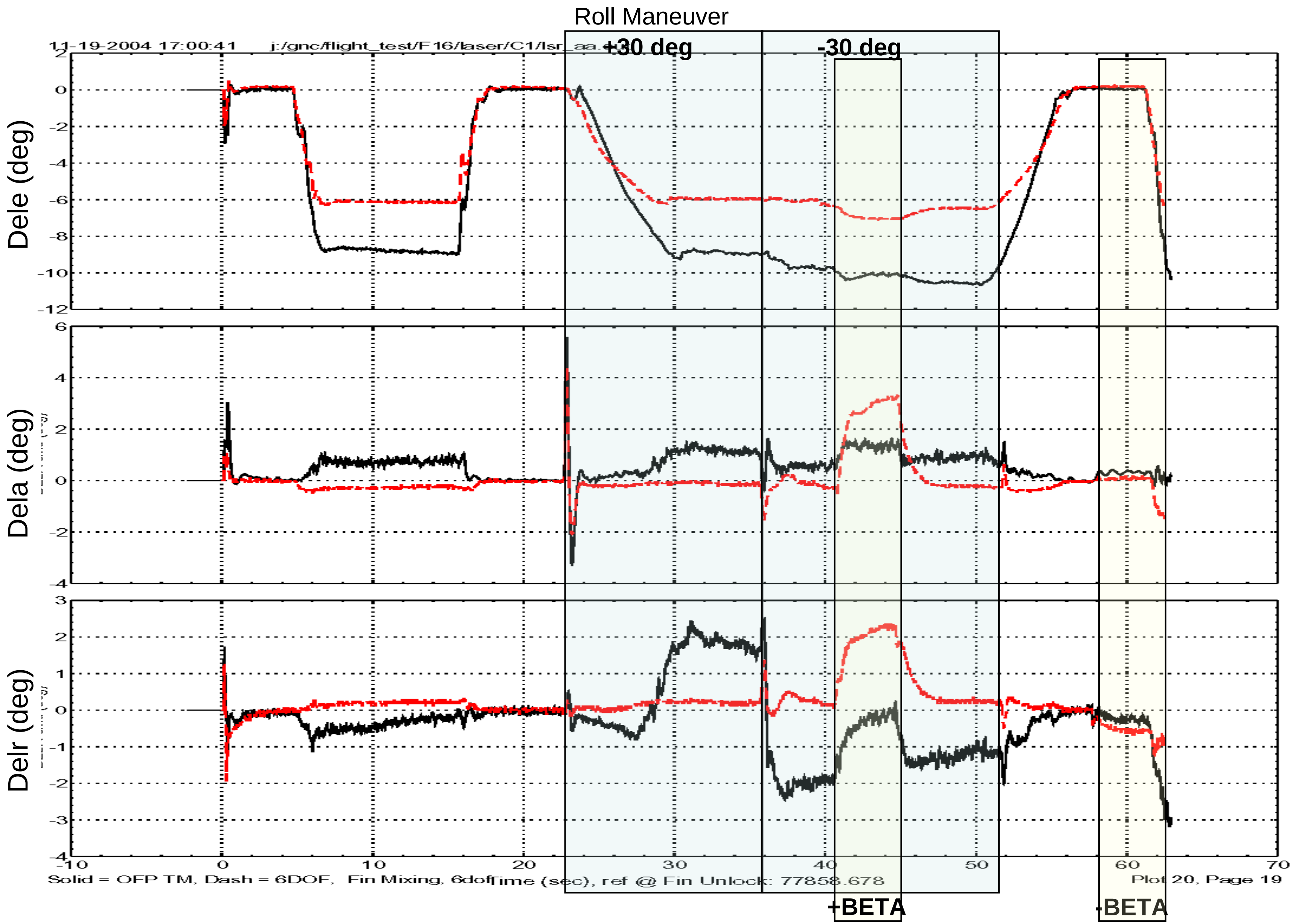




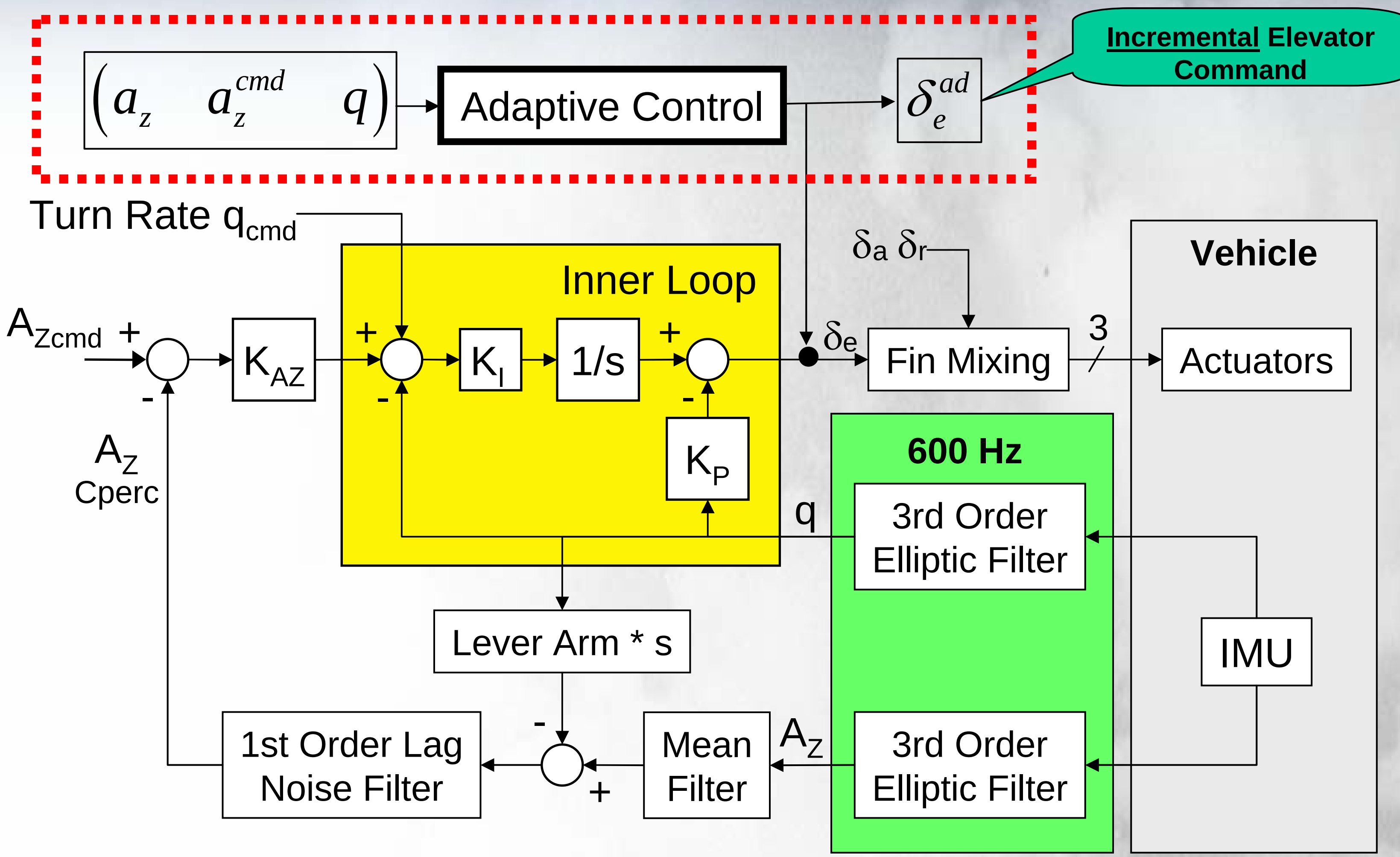
Oct 04 Flight Data (1 of 2)



Oct 04 Flight Data (2 of 2)

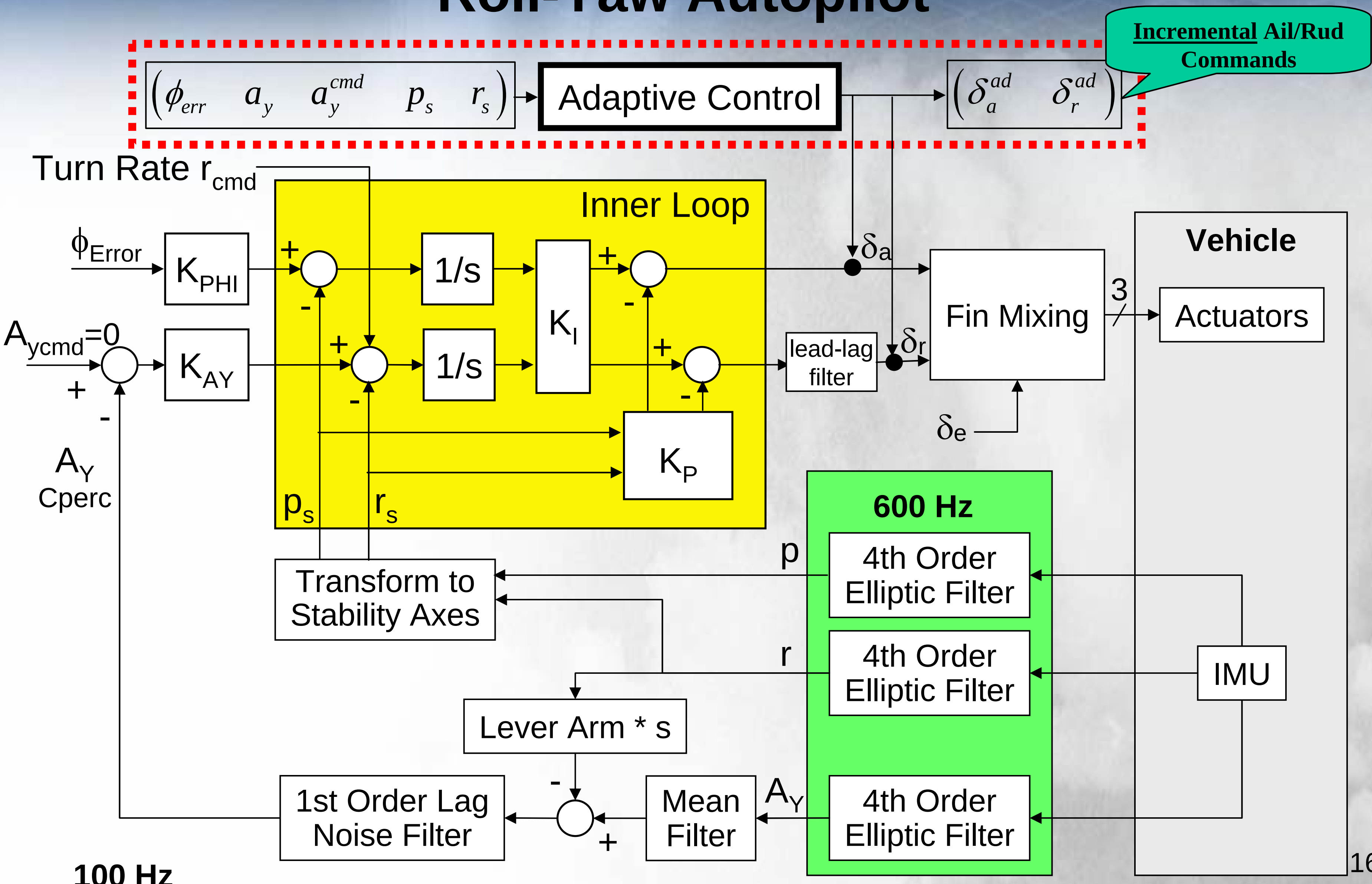


Adaptive Augmentation of Pitch Autopilot



100 Hz

Adaptive Augmentation of Roll-Yaw Autopilot



LJDAM - Jan 05 Fixed Target

LJDAM – May 05 Moving Target

LJDAM – May 05 Moving Target





Lessons Learned

- **X-36 RESTORE Flight Test**

  - **Stabilized Unstable Airframe Under Significant Failures**
  - **Limited Flight Envelope**

- **MK-84 JDAM Dynamic Inversion CLAW**

  - **Eliminated Gain Scheduling Requirements**
  - **Used Existing Truth Model for Analysis/Comparison**

- **MK-82 LJDAM Augmented LQR**

  - **Retrofit Onto Baseline Control**
  - **Significant Parameter Tuning Required For Performance**

Flight Results Have Created List of Open Problems

Open Problems

- *Reference Model Design*
- *Parameter Tuning Guidelines*
- *Adaptive Dead-zone and Learning Rates*
- *Adaptive Structural Mode Suppression*
- *Gain and Phase Margins for Adaptive Systems*
- Retrofit For Legacy Systems

Summary

- ***DOD Requires Robust System Behavior for Autonomous UAV and Weapon System Operation – Need for Adaptive Control***
- ***Flight Quality Computer Hardware Now Capable of Advanced Algorithms***
- ***Industry Actively Maturing Technology***
- ***Open Problems To Be Solved By Joint Academic/Industry Collaboration***
- ***Adaptive/Nonlinear Control Design Challenge in Work***

Frequency Domain Analysis Papers

Wise, K. A., Missile Autopilot Robustness Using The Real Multiloop Stability Margin, *Journal of Guidance, Control, and Dynamics*, Vol. 16, No. 2, 1993, pp. 354-362.

Wise, K. A., A Comparison of Six Robustness Tests Evaluating Missile Autopilot Robustness To Uncertain Aerodynamics, *Journal of Guidance, Control, and Dynamics*, Vol. 15, No. 4, 1992, pp. 861-870.

Wise, K. A., Singular Value Robustness Tests For Missile Autopilot Uncertainties, *Journal of Guidance, Control, and Dynamics*, Vol. 14, No. 3, 1991, pp. 597-606.

Wise, K. A., A Parameter Space Robustness Test for Real and Complex Uncertainties, Proc. of the 32nd IEEE CDC, San Antonio, TX, Dec. 13-17, 1993.

Wise, K. A., B. Mears, C. Tang, and A. Godhwani, A Convex Hull Program Evaluating Control System Robustness To Real Parameter Variations, Proceedings of the AIAA Guidance, Navigation, and Control Conference, Portland, OR, August, 1990, Paper AIAA 90-3339, pp. 223-231

Linear Robust Control Design Papers

Wise, K. A., and E. Hamby, H_∞ Missile Autopilot Design With and Without Imaginary Axis Zeros, Proc. of the ACC, Baltimore MD, June 1994, pp. 1670-1674.

Wise, K. A., and T. Nguyen, Optimal Disturbance Rejection In Missile Autopilot design Using Projective Controls, Proc. of the 30th IEEE CDC, Brighton, UK, Dec. 11-13, 1991.

Wise, K. A., and F. Deylami, Approximating a Linear Quadratic Missile Autopilot Design Using an Output Feedback Projective Control, Proceedings of the AIAA Guidance, Navigation, and Control Conference, New Orleans, LA, August, 1991, Paper AIAA 91-2613, pp. 114-122.

Wise, K. A., B. Mears and K. Poolla, Missile Autopilot Design Using H_∞ Optimal Control With μ-Synthesis, Proceedings of the American Control Conference, San Diego, CA. May, 1990, pp. 2362-2367.

Wise, K. A., A Bank-To-Turn Missile Autopilot Design Approach Using LQG/LTR Control Theory", Proceedings of the AIAA 26th Aerospace Sciences Meeting, Reno, Nevada, January, 1988.

Neural Adaptive Control Application Papers

Brinker, J. S., and K. A. Wise, Flight Testing of Reconfigurable Control Law on the X-36 Tailess Aircraft, *Journal of Guidance, Control, and Dynamics*, Vol. 24, No. 5, 2001, pp. 903-917.

Calise, A., S. Lee, and M. Sharma, Development of a Reconfigruable Flight control Law for the X-36 tailless Fighter Aircraft, , Proc. of the AIAA GNC Conference, Denver, CO, August, 2000, AIAA-Paper 2000-3940.

Brinker, J. S., and K. A. Wise,, Flight Testing of a Reconfigurable Flight Control Law on the X-36 Tailless Fighter Aircraft, Proc. of the AIAA GNC Conference, Denver, CO, August, 2000, AIAA-Paper 2000-3941. – This paper has more material than the above journal version.

Sharma, M., A. Calise, and J. Corban, application of an Adaptive Autopilot Design to a Family of Guided Munitions, Proc. of the AIAA GNC Conference, Denver, CO, August, 2000, AIAA-Paper 2000-3969.

Johnson, E., A. Calise, H. El-Shirbiny, Feedback Linearization with Neural Network Augmentation Applied to the X-33 Attitude Control, Proc. of the AIAA GNC Conference, Denver, CO, August, 2000, AIAA-Paper 2000-4157.

Wise, K. A., J. Brinker, A. Calise, D. Enns, M Elgersma, and P. Voulgaris, Direct Adaptive Reconfigurable Flight Control For A Tailless Advanced Fighter Aircraft, *Int. Journal of Robust Nonlinear Control*, Special Issue on Reconfigurable Flight Control, Vol. 9, pp. 999-1012, 1999.

Wise, K. A., and J. S. Brinker, Nonlinear Simulation Analysis of a Tailless Advanced Fighter Aircraft Reconfigurable Flight Control Law, Proc. of the AIAA GNC Conference, Portland, OR, August, 1999.

Wise, K. A., and J. S. Brinker, Reconfigurable Flight Control for a Tailless Advanced Fighter Aircraft, Proc. of the AIAA GNC Conference, Boston, MA, August, 1998, pp. 75-87.

N. Hovakimyan, E. Lavretsky, B.-J. Yang, A. Calise, Coordinated Decentralized Adaptive Output Feedback For Control of Interconnected Systems, IEEE Transactions on Neural Networks, vol.16, No.1, 2005.

N. Hovakimyan, E. Lavretsky, A. Calise, R. Sattigeri, Decentralized Adaptive Output Feedback Control via Input/Output Inversion, IEEE Transactions on Systems, Man, Cybernetics, 2005.

E. Lavretsky, N. Hovakimyan, A. Calise, Upper Bounds for Approximation of Continuous-Time Dynamics Using Delayed Outputs and Feedforward Neural Networks, IEEE Transactions on Automatic Control, vol. 48 (9), pp.1606-1610, 2003.

E. Lavretsky, On the geometric convergence of neural approximations, IEEE Transactions on Neural Networks, col. 13, No. 2, pp. 274-282, March 2002.

E. Lavretsky, Comments on the unreasonable effectiveness of neural approximation, IEEE Transactions on Automatic Control, vol. 46, pp. 511-512, March 2001.

E. Lavretsky, On the exact solution of the Parity-N problem using ordered neural networks, Neural Networks, vol. 13 / 6, pp. 643-649, July 2000.

K.A. Wise, E. Lavretsky, N. Hovakimyan, "Adaptive Control of Flight: Theory, Applications, and Open Problems," 13th Yale Workshop on Adaptive and Learning Systems, May 30 - June 1, Becton Center, Yale University, New Haven, CT, 2005.

K.A. Wise, E. Lavretsky, J. Zimmerman, J.H. Francis-Jr., D. Dixon, and B. T. Whitehead, "Adaptive Control of a Sensor Guided Munition," In proceedings of 2005 AIAA GNC Conference, San Francisco, CA, Aug 2005.

E. Lavretsky, K.A. Wise, Adaptive Flight Control for Manned / Unmanned Military Aircraft, In Proceedings of American Control Conference, Portland, OR, 2005.

A.M. Annaswamy, M. Schwager, H. Jain, E. Lavretsky, "Towards a User Friendly Adaptive Control Design: Transients, Learning, and Constraint Management," 13th Yale Workshop on Adaptive and Learning Systems, May 30 - June 1, Becton Center, Yale University, New Haven, CT, 2005.

M. Schwager, A. M. Annaswamy, E. Lavretsky, “Adaptation-Based Reconfiguration in the Presence of Actuator Failures and Saturation,” In Proceedings of American Control Conference, Portland, OR, 2005.

N. Hovakimyan, E. Lavretsky, C. Cao, “Dynamic Inversion for Multivariable Nonaffine-in-Control Systems via Time-Scale Separation,” In proceedings of 44th IEEE Conference on Decision and Control and European Control Conference, 2005.

E. Lavretsky, N. Hovakimyan, “Adaptive Compensation of Control Dependent Modeling Uncertainties using Time-Scale Separation, In proceedings of 44th IEEE Conference on Decision and Control and European Control Conference, 2005.

A. Ajami, N. Hovakimyan, E. Lavretsky, “Design Examples: Adaptive Tracking Control in the Presence of Input Constraints for a Generic Missile and a Fighter Aircraft,” In Proceedings of AIAA Guidance, Navigation and Control Conference, San Francisco, CA, 2005.

M. Schwager, A. M. Annaswamy, E. Lavretsky, “Towards Verifiable Adaptive Flight Control for Safety Critical Applications,” In Proceedings of AIAA Guidance, Navigation and Control Conference, San Francisco, CA, 2005.

N. Hovakimyan, E. Lavretsky, A. Sasane, “Stabilization of Nonaffine Nonlinear Systems using Time-Scale Separation,” In Proceedings of 16th IFAC World Congress, Prague, Czech Republic, 2005.

N. Hovakimyan, E. Lavretsky, A. Sasane, “Dynamic Inversion for Nonaffine in Control Systems via Time-Scale Separation: Part I,” In Proceedings of American Control Conference, Portland, OR, 2005.

E. Lavretsky, N. Hovakimyan, “Adaptive Dynamic Inversion for Nonaffine in Control Systems via Time-Scale Separation: Part II,” In Proceedings of American Control Conference, Portland, OR, 2005.

E. Lavretsky, N. Hovakimyan, “Positive μ – modification for Stable Adaptation in Dynamic Inversion Based Adaptive Control with Input Saturation,” In Proceedings of American Control Conference, Portland, OR, 2005.

E. Lavretsky, K.A. Wise, “X-45A J-UCAS Adaptive / Reconfigurable Flight Control System Design and Evaluation”, Boeing Internal Report, 2004.

V. Stepanyan, E. Lavretsky, N. Hovakimyan, “Aerial Refueling Autopilot Design Methodology: Application to F-16 Aircraft,” AIAA-2004-5321, In Proceedings of AIAA Guidance, Navigation and Control Conference and Exhibit, Providence, Rhode Island, Aug. 16-19, 2004.

E. Lavretsky, N. Hovakimyan, "Positive μ – modification for stable adaptation in a class of nonlinear systems with actuator constraints," In Proceedings of American Control Conference, 2004.

N. Hovakimyan, E. Lavretsky, B.-J. Yang, A. Calise, "Decentralized Adaptive Output Feedback for Control of Large-Scale Systems using Neural Networks," In Proceedings of American Control Conference, 2004.

N. Hovakimyan, E. Lavretsky, A. Calise, R. Sattigeri, "Decentralized Adaptive Output Feedback Control via Input/Output Inversion," In Proceedings of 42nd IEEE Conference on Decision and Control, 2003.

N. Kim, E. Lavretsky, N. Hovakimyan, A. J. Calise, "Adaptive Controllers using Multilayer Neural Networks with Asymptotic Tracking," In Proceedings of 42nd IEEE Conference on Decision and Control, 2003.

V. Stepanyan, E. Lavretsky, N. Hovakimyan, "A Differential Game Approach to Aerial Refueling Autopilot Design," In Proceedings of 42nd IEEE Conference on Decision and Control, 2003.

E. Lavretsky, N. Hovakimyan, A. Calise, V. Stepanyan, "Vortex Seeking Formation Flight Neurocontrol," In Proceedings of AIAA Conference on Guidance, Navigation and Control, 2003.

E. Lavretsky, N. Hovakimyan, A. Calise, "Adaptive Extremum Seeking Control Design," In Proceedings of American Control Conference, 2003.

E. Lavretsky, Misovec, K. "Phase I: Formation Flight Control Design", AIAA-2002-3429, 1st Unmanned Aerospace Vehicles, Systems, Technologies, and Operations Conference and Workshop, May 2002, Portsmouth, VA.

E. Lavretsky "F/A-18 Autonomous Formation Flight Control System Design", AIAA-2002-4757, AIAA Guidance, Navigation and Control Conference, August 2002, Monterey, CA.

E. Lavretsky "Health Monitoring of an Electro-Hydraulic System using Ordered Neural Networks", World Congress on Computational Intelligence, May 2002, Honolulu, HI.

E. Lavretsky, "Neurocontrol design using ordered networks," Guidance Navigation and Control Conference, August 2000, Denver, CO.

D.K., Henderson, and E. Lavretsky, "Closed-loop model following and control allocation for transport aircraft using an ordered neural network-based approach," AIAA Paper 99-4080, Guidance Navigation and Control Conference, August 1999, Portland, OR.

E. Lavretsky, and D.K. Henderson, "A neural network-based approach to active structural mode suppression for flexible transport aircraft," ACC99-AIAA0004 / FP08-1, American Control Conference, June 1999, San Diego, CA.

E. Lavretsky, "Ordered Neural Networks: Formulation, Development, Design and Application," AIAA Paper 99-0106, 37th Aerospace Sciences Meeting & Exhibit, January 1999, Reno, NV.