Adaptive Control: Introduction, Overview, and Applications

Eugene Lavretsky, Ph.D.

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

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

- Recent Journal / Conference Publications, (available upon request)
Motivating Example: Roll Dynamics  
(Model Reference Adaptive Control)

- **Uncertain Roll dynamics:**
  - $p$ is roll rate,
  - $\delta_{ail}$ is aileron position
  - $(L_p, L_{\delta_{ail}})$ are *unknown* damping, aileron effectiveness

- **Flying Qualities Model:**
  - $(L_p^m, L_{\delta}^m)$ are *desired* damping, control effectiveness
  - $\delta(t)$ is a reference input, (pilot stick, guidance command)
  - roll rate tracking error:

$$e_p(t) = (p(t) - p_m(t)) \rightarrow 0$$

- **Adaptive Roll Control:**

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

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

Parameter adaptation laws
Motivating Example: Roll Dynamics
(Block-Diagram)

- Adaptive control provides Lyapunov stability
- Design is based on Lyapunov Theorem (2nd method)
- Yields closed-loop asymptotic tracking with all remaining signals bounded in the presence of system uncertainties
Lyapunov Stability Theory
Alexander Michailovich Lyapunov 1857-1918

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

• A nonlinear dynamic system can usually be represented by a set of $n$ differential equations in the form:

$$\dot{x} = f(x, t), \text{ with } x \in R^n, t \in R$$

  – $x$ 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**:

$$\dot{x} = f(x)$$

• A state $x_e$ is an equilibrium if once $x(t) = x_e$, it remains equal to $x_e$ for all future times:

$$0 = f(x)$$
Example: Equilibrium Points of a Pendulum

- **System dynamics:**
  \[
  M R^2 \ddot{\theta} + b \dot{\theta} + M g R \sin(\theta) = 0
  \]

- **State space representation,**
  \[
  (x_1 = \theta, \quad x_2 = \dot{\theta})
  \]
  \[
  \begin{align*}
  \dot{x}_1 &= x_2 \\
  \dot{x}_2 &= -\frac{b}{M R^2} x_2 - \frac{g}{R} \sin(x_1)
  \end{align*}
  \]

- **Equilibrium points:**
  \[
  0 = x_2 \\
  0 = -\frac{b}{M R^2} x_2 - \frac{g}{R} \sin(x_1)
  \]
  \[
  x_2 = 0, \quad \sin(x_1) = 0
  \]
  \[
  x_e = \left( \begin{array}{c}
  \pi k \\
  0 
  \end{array} \right), \quad (k = 0, \pm 1, \pm 2, \ldots)
  \]
Example: Linear Time-Invariant (LTI) Systems

• LTI system dynamics: \( \dot{x} = Ax \)
  - has a single equilibrium point (the origin 0) if \( A \) is nonsingular
  - has an infinity of equilibrium points in the null-space of \( A \): \( Ax_e = 0 \)

• LTI system trajectories: \( x(t) = \exp(A(t-t_0))x(t_0) \)

• 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 $\dot{x} = f(x)$
• New system dynamics: $\dot{y} = f(y + x_e)$
• New equilibrium: $y = 0$, (since $f(x_e) = 0$)
• Conclusion: study the behavior of the new system in the neighborhood of the origin
Nominal Motion

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

- The error dynamics: $\dot{e} = f(x^*(t) + e(t)) - f(x^*(t)) = g(e, t)$
  - non-autonomous!
- **Conclusion**: Instead of studying stability of the nominal motion, study stability of the error dynamics w.r.t. the origin
Lyapunov Stability

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

\[
\forall R > 0, \quad \exists r > 0, \quad \{\|x(0)\| < r\} \Rightarrow \{\forall t \geq 0, \|x(t)\| < R\}
\]

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

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

- Asymptotic stability means that the equilibrium is stable, and that in addition, states started close to 0 actually converge to 0 as time \( t \) goes to infinity.

- Equilibrium point that is stable but not asymptotically stable is called *marginally stable*.
Exponential Stability

- **Definition**: An equilibrium point 0 is **exponentially stable** if:

\[ \exists r, \alpha, \lambda > 0, \quad \forall \{\|x(0)\| < r \wedge t > 0\}: \quad \|x(t)\| \leq \alpha \|x(0)\| e^{-\lambda t}, \]

- 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: \( \dot{x} = f(x) \)

• Assume that \( f(x) \) is continuously differentiable

• Perform linearization:

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

• 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

\[ m \ddot{x} + \begin{cases} b \dot{x} |\dot{x}| + k_0 x + k_1 x^3 \end{cases} = 0 \]

• **Question**: If the mass is pulled away and then released, will the resulting motion be stable?
  – Stability definitions are hard to verify
  – Linearization method fails, (linear system is only marginally stable)
Lyapunov’s Direct Method (Motivating Example, continued)

• Total mechanical energy

\[
V(x) = \frac{1}{2} m \dot{x}^2 + \int_0^x \left( k_0 x + k_1 x^3 \right) dx = \frac{1}{2} m \dot{x}^2 + \frac{1}{2} k_0 x^2 + \frac{1}{4} k_1 x^4
\]

- kinetic
- potential

• Total energy rate of change along the system’s motion:

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

• **Conclusion**: Energy of the system is dissipated until the mass settles down: \( \dot{x} = 0 \)
Lyapunov’s Direct Method (Overview)

• Method
  – based on generalization of energy concepts

• Procedure
  – generate a scalar “energy-like function (Lyapunov function) for the dynamic system, and examine its variation in time, (derivative along the system trajectories)
  – if energy is dissipated (derivative of the Lyapunov function is non-positive) then conclusions about system stability may be drawn
Positive Definite Functions

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

\[
V(0) = 0 \land \{ \forall x \neq 0 \land \|x\| < R \} \Rightarrow V(x) > 0
\]

- If $V(0) = 0 \land \{ \forall x \neq 0 \} \Rightarrow V(x) > 0$ then $V(x)$ is *globally positive definite*

- **Remarks**
  - a positive definite function must have a unique minimum
    \[
    \min_{x \in B_R} V(x) = V(x_{\text{min}}) = V_{\text{min}}
    \]
  - if $V_{\text{min}} \neq 0$ or $x_{\text{min}} \neq 0$ then use
    \[
    W(x) = V(x - x_{\text{min}}) - V_{\text{min}}
    \]
Lyapunov Functions

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

- **Time derivative** of the Lyapunov function

$$\dot{V}(x) = \nabla V(x) \cdot f(x) \leq 0, \quad \nabla V(x) = \left( \frac{\partial V(x)}{\partial x_1} \quad \cdots \quad \frac{\partial V(x)}{\partial x_n} \right) \in \mathbb{R}^n$$
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 $\forall x \in B_R : \ V(x) > 0 \land \dot{V}(x) \leq 0$, then the equilibrium point 0 is **stable**
  - If the time derivative is locally negative definite $\dot{V}(x) < 0$, then the stability is **asymptotic**
  - If $V(x)$ is radially unbounded, i.e., $\lim_{\|x\| \to \infty} V(x) = \infty$, then the origin is **globally asymptotically stable**

- $V(x)$ is called the Lyapunov function of the system
Example: Local Stability

• Pendulum with viscous damping: \( \dot{\theta} + \dot{\theta} + \sin \theta = 0 \)
• State vector: \( x = (\theta \; \dot{\theta})^T \)
• Lyapunov function candidate: \( V(x) = (1 - \cos \theta) + \frac{\dot{\theta}^2}{2} \)
  – represents the total energy of the pendulum
  – locally positive definite
  – time-derivative is negative semi-definite

\[ \dot{V}(x) = \frac{\partial V(x)}{\partial \theta} \dot{\theta} + \frac{\partial V(x)}{\partial \dot{\theta}} \ddot{\theta} = \dot{\theta} \sin \theta + \dot{\theta} \ddot{\theta} \leq 0 \]

• Conclusion: System is locally stable
Example: Asymptotic Stability

- **System Dynamics:**
  \[
  \begin{align*}
  \dot{x}_1 &= x_1 \left( x_1^2 + x_2^2 - 2 \right) - 4x_1 x_2^2 \\
  \dot{x}_2 &= x_2 \left( x_1^2 + x_2^2 - 2 \right) + 4x_1^2 x_2
  \end{align*}
  \]

- **Lyapunov function candidate:**
  \[
  V(x_1, x_2) = x_1^2 + x_2^2
  \]
  - positive definite
  - time-derivative is *negative definite* in the 2-dimensional ball defined by \(x_1^2 + x_2^2 < 2\)
  \[
  \dot{V}(x_1, x_2) = 2(x_1^2 + x_2^2)(x_1^2 + x_2^2 - 2) < 0
  \]

- **Conclusion:** System is *locally* asymptotically stable
Example: Global Asymptotic Stability

• **Nonlinear 1\textsuperscript{st} order system**
  \[ \dot{x} = -c(x), \text{ where: } xc(x) > 0 \]

• **Lyapunov function candidate:**
  – globally positive definite
  – time-derivative is negative definite
  \[ \dot{V}(x) = 2x\dot{x} = -2xc(x) < 0 \]

• **Conclusion:** System is globally asymptotically stable

• **Remark:** Trajectories of a 1\textsuperscript{st} order system are monotonic functions of time, (why?)
La Salle’s Invariant Set Theorems

• It often happens that the time-derivative of the Lyapunov function is only negative semi-definite
• It is still possible to draw conclusions on the asymptotic stability
• Invariant Set Theorems (attributed to La Salle) extend the concept of Lyapunov function
Example: 2\textsuperscript{nd} Order Nonlinear System

- System dynamics: $\ddot{x} + b(\dot{x}) + c(x) = 0$
  - where $b(x)$ and $c(x)$ are continuous functions verifying the sign conditions:
    - $\dot{x}b(\dot{x}) > 0$, for $\dot{x} \neq 0$
    - $xc(x) > 0$, for $x \neq 0$

- Lyapunov function candidate:
  - positive definite
  - time-derivative is negative semi-definite
    $$\dot{V} = \ddot{x} \dot{x} + c(x) \dot{x} = -\ddot{x} b(\dot{x}) \leq 0$$

- system energy is dissipated
  $$\dot{b}(\dot{x}) = 0 \iff \dot{x} = 0 \implies \ddot{x} = -c(x) \implies x_e = 0$$
  - system cannot get “stuck” at a non-zero equilibrium

- **Conclusion**: Origin is globally asymptotically stable
Lyapunov Functions for LTI Systems

• LTI system dynamics: $\dot{x} = Ax$

• Lyapunov function candidate: $V(x) = x^T P x$
  – where $P$ is symmetric positive definite matrix
  – function $V(x)$ is positive definite

• Time-derivative of $V(x(t))$ along the system trajectories:
  $\dot{V}(x) = \dot{x}^T P x + x^T P \dot{x} = x^T (A^T P + PA) x = -x^T Q x < 0$
  – where $Q$ is symmetric positive definite matrix
  – Lyapunov equation: $A^T P + PA = -Q$

• Stability analysis procedure:
  – choose a symmetric positive definite $Q$
  – solve the Lyapunov equation for $P$
  – check whether $P$ is positive definite
Stability of LTI Systems

• **Theorem**
  – An LTI system is stable (globally exponentially) if and only if for any symmetric positive definite matrix $Q$, the unique matrix solution $P$ of the Lyapunov equation is symmetric and positive definite

• **Remark**: In most practical cases $Q$ is chosen to be a diagonal matrix with *positive* diagonal elements
Barbalat’s Lemma: Preliminaries

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

• **Lemma**
  – If a differentiable function \( f(t) \) has a finite limit as \( t \to \infty \) and if \( \dot{f}(t) \) is *uniformly continuous*, then \[
  \lim_{t \to \infty} \dot{f}(t) = 0
  \]

• **Remarks**
  – *Uniform continuity* of a function is difficult to verify directly
  – simple *sufficient condition*:
    • if derivative is bounded then function is uniformly continuous
  – The fact that derivative goes to zero does not imply that the function has a limit, as \( t \) tends to infinity. The converse is also not true, (in general)
  – Uniform continuity condition is very important!
Example: LTI System

• **Statement:** Output of a stable LTI system is uniformly continuous in time
  
  – System dynamics: \( \dot{x} = Ax + Bu \)
  
  – Control input \( u \) is bounded
  
  – System output: \( y = Cx \)

• **Proof:** Since \( u \) is bounded and the system is stable then \( x \) is bounded. Consequently, the output time-derivative \( \dot{y} = C \dot{x} = C(Ax + Bu) \) is bounded. Thus, (using Barbalat’s Lemma), we conclude that the output \( y \) is **uniformly continuous** in time.
Lyapunov-Like Lemma

• If a scalar function $V(x,t)$ satisfies the following conditions
  – function is lower bounded
  – its time-derivative along the system trajectories is negative semi-definite and uniformly continuous in time
• Then: $\lim_{t \to \infty} \dot{V}(x,t) = 0$

• Question: Why is this fact so important?
• Answer: It provides theoretical foundations for stable adaptive control design

E. Lavretsky
Example: Stable Adaptation

• Closed-loop error dynamics of an adaptive system
  \[ \dot{e} = -e + \theta w(t), \quad \dot{\theta} = -e w(t) \]
  – where \( e \) is the tracking error, \( \theta \) is the parameter error,
    and \( w(t) \) is a bounded continuous function

• Stability Analysis
  – Consider Lyapunov function candidate:
    \[ V(e, \theta) = e^2 + \theta^2 \]
    • it is positive definite
    • its time-derivative is negative semi-definite
      \[ \dot{V}(e, \theta) = 2e(-e + \theta w) + 2\theta (-e w) = -2e^2 \leq 0 \]
    • consequently, \( e \) and \( \theta \) are bounded
    • since \( \dot{V}(e, \theta) = -4e(-e + \theta w) \) is bounded,
      \( \dot{V}(e, \theta) \) is uniformly continuous
    • hence:
      \[ \lim_{t \to \infty} (-2e^2) = \lim_{t \to \infty} \dot{V}(e, \theta) = 0 \Rightarrow \lim_{t \to \infty} e(t) \]
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:

\[
\exists C_0 > 0, T = T(C_0, B) > 0: \quad (\|x(t_0)\| \leq C_0) \Rightarrow (\|x(t)\| \leq B, \quad \forall t \geq t_0 + T)
\]

- **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 $|x(t_0)| \leq C_0$ there exists a bound $B$ and a time $T(B, x(t_0))$ such that $|x(t) - x_e| \leq B$ for all $t \geq t_0 + T$.
UUB by Lyapunov Extension

- Milder form of stability than SISL
- More useful for controller design in practical systems with unknown bounded disturbances:
  \[ \dot{x} = f(x) + d(x) \]

- **Theorem**: Suppose that there exists a function \( V(x) \) with continuous partial derivatives such that for \( x \) in a compact set \( S \subset \mathbb{R}^n \)
  - \( V(x) \) 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 \( \|x(t)\| \leq R, \quad \forall t \geq t_0 + T \)
Example: UUB by Lyapunov Extension

- **System:**
  \[
  \begin{align*}
  \dot{x}_1 &= x_1 x_2^2 - x_1 (x_1^2 + x_2^2 - 9) \\
  \dot{x}_2 &= -x_1^2 x_2 - x_2 (x_1^2 + x_2^2 - 9)
  \end{align*}
  \]

- **Lyapunov function candidate:**
  \[
  V(x_1, x_2) = x_1^2 + x_2^2
  \]

- **Time derivative:**
  \[
  \dot{V}(x_1, x_2) = 2(x_1 \dot{x}_1 + x_2 \dot{x}_2) = -2(x_1^2 + x_2^2)(x_1^2 + x_2^2 - 9)
  \]

- **Time derivative negative outside compact set**
  \[
  \dot{V}(x_1, x_2) < 0, \quad \forall \{x : x_1^2 + x_2^2 > 9\}
  \]

- **Conclusion:** All trajectories enter circle of radius \( R = 3 \), in a finite time
Adaptive Control
Introduction

• Basic Ideas in Adaptive Control
  – estimate uncertain plant / controller parameters on-line, while using measured system signals
  – use estimated parameters in control input computation

• Adaptive controller is a dynamic system with on-line parameter estimation
  – inherently nonlinear
  – analysis and design rely on the Lyapunov Stability Theory
Historical Perspective

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

• **Why Adaptive Control?**
  – dealing with complex systems that have unpredictable parameter deviations and uncertainties

• **Basic Objective**
  – maintain consistent performance of a system in the presence of uncertainty and variations in plant parameters

• Adaptive control is superior to robust control in dealing with uncertainties in constant or slow-varying parameters

• Robust control has advantages in dealing with disturbances, quickly varying parameters, and unmodeled dynamics

• **Solution**: Adaptive augmentation of a Robust Baseline controller
Model-Reference Adaptive Control (MRAC)

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

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

- **Indirect**
  - estimate plant parameters
  - compute controller parameters
  - relies on convergence of the estimated parameters to their true unknown values

- **Direct**
  - no plant parameter estimation
  - estimate controller parameters (gains) only

- MRAC and STC can be designed using both Direct and Indirect approaches

- **We consider Direct MRAC design**
MRAC Design of 1st Order Systems

**System Dynamics:** \[ \dot{x} = ax + b(u - f(x)) \]
- \(a, b\) are constant *unknown* parameters
- *uncertain* nonlinear function: \[ f(x) = \sum_{i=1}^{N} \theta_i \varphi_i(x) = \theta^T \Phi(x) \]
  - vector of constant *unknown* parameters: \(\theta = (\theta_1 \ldots \theta_N)^T\)
  - vector of known basis functions: \(\Phi(x) = (\varphi_1(x) \ldots \varphi_N(x))^T\)

**Stable Reference Model:** \[ \dot{x}_m = a_m x_m + b_m r, \quad (a_m < 0) \]

**Control Goal**
- find \(u\) such that: \[ \lim_{t \to \infty} (x(t) - x_m(t)) = 0 \]
MRAC Design of 1st Order Systems (continued)

- Control Feedback: 
  \[ u = \hat{k}_x x + \hat{k}_r r + \hat{\theta}^T \Phi(x) \]
  - \((N + 2)\) parameters to estimate on-line: \(\hat{k}_x, \hat{k}_r, \hat{\theta}\)

- Closed-Loop System:
  \[ \dot{x} = (a + b \hat{k}_x) x + b \left( \hat{k}_r r + \left( \hat{\theta} - \theta \right)^T \Phi(x) \right) \]

- Desired Dynamics:
  \[ \dot{x}_m = a_m x_m + b_m r \]

- Matching Conditions Assumption
  - there exist ideal gains \((k_x, k_r)\) such that:
    \[ \begin{align*}
      a + b k_x &= a_m \\
      b k_r &= b_m
    \end{align*} \]
  - \textbf{Note}: knowledge of the ideal gains is not required, only their existence is needed
  - consequently:
    \[ \begin{align*}
      a + b \hat{k}_x - a_m &= a + b \hat{k}_x - a - b k_x = b \left( \hat{k}_x - k_x \right) = b \Delta k_x \\
      b \hat{k}_r - b_m &= b \hat{k}_r - b k_r = b \left( \hat{k}_r - k_r \right) = b \Delta k_r
    \end{align*} \]
MRAC Design of 1st Order Systems (continued)

- **Tracking Error:** \[ e(t) = x(t) - x_m(t) \]

- **Error Dynamics:**

\[
\dot{e}(t) = \dot{x}(t) - \dot{x}_m(t) = \left(a + b \hat{k}_x\right)x + b\left(k_r + (\hat{\theta} - \theta)^T \Phi(x)\right) - a_m x_m - b_m r \pm a_m x
\]

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

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

- **Lyapunov Function Candidate:**

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

- where: \( \gamma_x > 0, \gamma_r > 0, \) and \( \Gamma = \Gamma^T > 0 \) is symmetric positive definite matrix
MRAC Design of 1st Order Systems (continued)

• Time-derivative of the Lyapunov function

\[
\dot{V}(e, \Delta k_x, \Delta k_r, \Delta \theta) = 2 e \dot{e} + 2 |b| \left( \gamma_x^{-1} \Delta k_x \dot{k}_x + \gamma_r^{-1} \Delta k_r \dot{k}_r + \Delta \theta^T \Gamma_{\theta}^{-1} \dot{\theta} \right) \\
= 2 e \left( a_m e + b (\Delta k_x x + \Delta k_r r) + \Delta \theta^T \Phi(x) \right) \\
+ 2 |b| \left( \gamma_x^{-1} \Delta k_x \dot{k}_x + \gamma_r^{-1} \Delta k_r \dot{k}_r + \Delta \theta^T \Gamma_{\theta}^{-1} \dot{\theta} \right) \\
= 2 a_m e^2 + 2 |b| \left( \Delta k_x \left( x e \text{sgn}(b) + \gamma_x^{-1} \dot{k}_x \right) \right) \\
+ 2 |b| \left( \Delta k_r \left( r e \text{sgn}(b) + \gamma_r^{-1} \dot{k}_r \right) \right) + 2 |b| \Delta \theta^T \left( \Phi(x) e \text{sgn}(b) + \Gamma_{\theta}^{-1} \dot{\theta} \right)
\]
• 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

\[
\begin{align*}
\dot{k}_x &= -\gamma_x x e \text{sgn}(b) \\
\dot{k}_r &= -\gamma_r r e \text{sgn}(b) \\
\dot{\theta} &= -\Gamma_\theta \Phi(x) e \text{sgn}(b)
\end{align*}
\]

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

\[
\dot{V}(e(t), \Delta k_x(t), \Delta k_r(t), \Delta \theta(t)) = 2 a_m e(t)^2 \leq 0
\]
MRAC Design of 1\textsuperscript{st} Order Systems (continued)

• Closed-Loop System Stability Analysis
  – Since $V \geq 0$ and $\dot{V} \leq 0$ then all the parameter estimation errors are bounded
  – Since the true (unknown) parameters are constant then all the estimated parameters are bounded

• Assumption
  – reference input $r(t)$ is bounded

• Consequently, $x_m$ and $\dot{x}_m$ are bounded
MRAC Design of 1st Order Systems (continued)

• Since $x = e + x_m$ then $x$ is bounded
• Consequently, the adaptive control feedback $u$ is bounded
• Thus, $\dot{x}$ is bounded, and $\dot{e} = \dot{x} - \dot{x}_m$ is bounded, as well
• It immediately follows that $\ddot{V} = 4a_m e(t) \dot{e}(t)$ is bounded
• Using Barbalat’s Lemma we conclude that $\dot{V}(t)$ is uniformly continuous function of time
MRAC Design of 1st Order Systems (completed)

- Using Lyapunov-like Lemma: \( \lim_{t \to \infty} \dot{V}(x, t) = 0 \)

- Since \( \dot{V} = 2a_m e(t)^2 \) it follows that: \( \lim_{t \to \infty} e(t) = 0 \)

- Conclusions
  - achieved asymptotic tracking: \( x(t) \to x_m(t) \), as \( t \to \infty \)
  - all signals in the closed-loop system are bounded
MRAC Design of 1st Order Systems (Block-Diagram)

- Adaptive gains: $\hat{k}_x(t), \hat{k}_r(t)$
- On-line function estimation: 
  $$\hat{f}(x) = \hat{\theta}^T(t) \Phi(x) = \sum_{i=1}^{N} \hat{\theta}_i(t) \phi_i(x)$$
Adaptive Dynamic Inversion (ADI) Control
ADI Design of 1st Order Systems

• System Dynamics: \( \dot{x} = ax + bu + f(x) \)
  
  – \( a, b \) are constant \textit{unknown} parameters
  
  – \textit{uncertain} nonlinear function: \( f(x) = \sum_{i=1}^{N} \theta_i \phi_i(x) = \theta^T \Phi(x) \)

• vector of constant \textit{unknown} parameters: \( \theta = (\theta_1 \ldots \theta_N)^T \)

• vector of known basis functions: \( \Phi(x) = (\phi_1(x) \ldots \phi_N(x))^T \)

• Stable Reference Model: \( \dot{x}_m = a_m x_m + b_m r, \quad (a_m < 0) \)

• Control Goal
  
  – find \( u \) such that: \( \lim_{t \to \infty} (x(t) - x_m(t)) = 0 \)
ADI Design of 1\textsuperscript{st} Order Systems (continued)

- Rewrite system dynamics:
  \[
  \dot{x} = \hat{a} x + \hat{b} u + \hat{f}(x) - \left(\hat{a} - a\right)x - \left(\hat{b} - b\right)u - \left(\hat{f}(x) - f(x)\right)
  \]

- Function estimation error:
  \[
  \Delta f(x) \triangleq \hat{f}(x) - f(x) = \left(\hat{\theta} - \theta\right)^T \Phi(x) \quad (\Delta \theta)
  \]

- On-line estimated parameters: \( \hat{a}, \hat{b}, \hat{\theta} \)

- Parameter estimation errors
  \[
  \Delta a \triangleq \hat{a} - a, \quad \Delta b \triangleq \hat{b} - b, \quad \Delta \theta \triangleq \hat{\theta} - \theta
  \]
ADI Design of 1\textsuperscript{st} Order Systems (continued)

- ADI Control Feedback:
  
  \[ u = \frac{1}{\hat{b}} \left( (a_m - \hat{\alpha}) x + b_m r \right) - \hat{\theta}^T \Phi(x) \]

  - \((N + 2)\) parameters to estimate on-line: \(\hat{\alpha}, \hat{b}, \hat{\theta}\)
  - Need to protect \(\hat{b}\) from crossing zero

- Closed-Loop System:
  
  \[ \dot{x} = a_m x + b_m r - \Delta a x - \Delta b u - \Delta \theta \Phi(x) \]

- Desired Dynamics:
  
  \[ \dot{x}_m = a_m x_m + b_m r \]

- Tracking error:
  
  \[ e \triangleq x - x_m \]

- Tracking error dynamics:
  
  \[ \dot{e} = a_m e - \Delta a x - \Delta b u - \Delta \theta \Phi(x) \]

- Lyapunov function candidate
  
  \[ V \left( e(t), \Delta a(t), \Delta b(t), \Delta \theta(t) \right) = e^2 + \gamma_a^{-1} \Delta a^2 + \gamma_b^{-1} \Delta b^2 + \Delta \theta^T \Gamma^{-1}_\theta \Delta \theta \]

E. Lavretsky
ADI Design of 1\textsuperscript{st} Order Systems (continued)

- **Time-derivative of the Lyapunov function**

\[ \dot{V}(e, \Delta a, \Delta b, \Delta \theta) = 2 e \dot{e} + 2 \left( \gamma_a^{-1} \Delta a \dot{a} + \gamma_b^{-1} \Delta b \dot{b} + \Delta \theta^T \Gamma^{-1} \dot{\theta} \right) \]

\[ = 2 e \left( a_m e - \Delta a x - \Delta b u - \Delta \theta \Phi(x) \right) + 2 \left( \gamma_a^{-1} \Delta a \dot{a} + \gamma_b^{-1} \Delta b \dot{b} + \Delta \theta^T \Gamma^{-1} \dot{\theta} \right) \]

\[ = 2 a_m e^2 + \Delta a \left( \gamma_a^{-1} \dot{a} - x e \right) + \Delta b \left( \gamma_b^{-1} \dot{b} - u e \right) + \Delta \theta^T \left( \Gamma^{-1} \dot{\theta} - \Phi(x) e \right) \]

- **Adaptive laws**

\[ \dot{a} = \gamma_a x e \]

\[ \dot{b} = \gamma_b u e \]

\[ \dot{\theta} = \Gamma \Phi(x) e \]

\[ \dot{V}(e, \Delta a, \Delta b, \Delta \theta) = 2 a_m e^2 \leq 0 \]

System energy decreases
ADI Design of 1st Order Systems (stability analysis)

- Similar to MRAC
- Using Barbalat’s Lemma and Lyapunov-like Lemma:
  \[ \lim_{t \to \infty} \dot{V}(x,t) = \lim_{t \to \infty} \left[ 2 a_m e(t)^2 \right] = 0 \]
- Consequently:
  \[ \lim_{t \to \infty} e(t) = 0 \quad \Rightarrow \quad x(t) \to x_m(t), \text{ as } t \to \infty \]
- 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
  \[
  \dot{e}(t) = a_m e + b \left( \Delta k_x x + \Delta k_r r + \Delta \theta^T \Phi(x) \right)
  \]

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

- Using vector form:
  \[
  \begin{pmatrix} x & r & \Phi^T(x) \end{pmatrix} \begin{pmatrix} \Delta k_x \\ \Delta k_r \\ \Delta \theta \end{pmatrix} \approx 0
  \]
Persistency of Excitation (PE) (completed)

- If \( r(t) \) is such that
  \[
  v = \begin{pmatrix} x & r & \Phi^T(x) \end{pmatrix}^T
  \]
satisfies the so-called “persistent excitation” conditions, then the adaptive parameter convergence will take place.

  - PE Condition:
    \[
    \exists \alpha > 0 \quad \forall t \quad \exists T > 0 \quad \int_t^{t+T} v(\tau) v^T(\tau) d\tau > \alpha I_{N+2}
    \]

- PE Condition implies that parameter errors converge to zero.
  - for linear systems: \( m \) - sinusoids ensure convergence of \( (2^m) \) - parameters
  - not known for nonlinear systems
ADI vs. MRAC

• No knowledge about $\text{sgn} \hat{b}$
• 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 \textbf{Linear} System

- Unstable Dynamics: \[ \dot{x} = x + 3u, \quad x(0) = 0 \]
  - plant parameters \( a = 1, \quad b = 3 \) are \textit{unknown} to the adaptive controller

- Reference Model: \[ \dot{x}_m = -4x_m + 4r(t), \quad x_m(0) = 0 \]

- Adaptive Control: \[ u = \hat{k}_x x + \hat{k}_r r \]

- Parameter Adaptation:
  - \[ \dot{\hat{k}}_x = -2x e, \quad \hat{k}_x(0) = 0 \]
  - \[ \dot{\hat{k}}_r = -2r e, \quad \hat{k}_r(0) = 0 \]

- 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

E. Lavretsky
Example: MRAC of a 1st-Order Nonlinear System

- **Unstable Dynamics:**
  \[
  \dot{x} = x + 3(u - f(x)), \quad x(0) = 0
  \]
  - plant parameters \( a = 1, \quad b = 3 \) are *unknown*
  - nonlinearity: \( f(x) = \theta^T \Phi(x) \)
    - *known* basis functions: \( \Phi(x) = \begin{pmatrix} x^3 & e^{-(x+0.5)^2} & e^{-(x-0.5)^2} & \sin(2x) \end{pmatrix}^T \)
    - *unknown* parameters: \( \theta = \begin{pmatrix} 0.01 & -1 & 1 & 0.5 \end{pmatrix}^T \)

- **Reference Model:**
  \[
  \dot{x}_m = -4x_m + 4r(t), \quad x_m(0) = 0
  \]

- **Adaptive Control:**
  \[
  u = \hat{k}_x x + \hat{k}_r r + \hat{\theta}^T \Phi(x)
  \]
  \[
  \dot{\hat{k}}_x = -2x e, \quad \hat{k}_x(0) = 0
  \]
  \[
  \dot{\hat{k}}_r = -2r e, \quad \hat{k}_r(0) = 0
  \]
  \[
  \dot{\hat{\theta}} = -2\Phi(x)e, \quad \hat{\theta}(0) = 0
  \]

- **Parameter Adaptation:**

- **Reference Input:**
  \[
  r(t) = \sin(3t) + \sin\left(\frac{3t}{2}\right) + \sin\left(\frac{3t}{4}\right) + \sin\left(\frac{3t}{8}\right)
  \]
1st-Order **Nonlinear** System
MRAC Simulation

Good Tracking & Poor Parameter Estimation
1st-Order **Nonlinear** System
MRAC Simulation, (continued)

**Nonlinearity: Poor Parameter Estimation**

E. Lavretsky
1st-Order Nonlinear System
MRAC Simulation, (completed)

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

- **Unstable Dynamics:**
  \[
  \dot{x} = x + 3(u - f(x)), \quad x(0) = 0
  \]
  - 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: \( \theta = \begin{pmatrix} 0 & -1 & 1 & 0 \end{pmatrix}^T \)

- **Reference Model:**
  \[
  \dot{x}_m = -4x_m + 4r(t), \quad x_m(0) = 0
  \]

- **Adaptive Control:**
  \[
  u = \hat{k}_x x + \hat{k}_r r + \hat{\theta}^T \Phi(x)
  \]
  \[
  \begin{align*}
  \hat{k}_x &= -2xe, \quad \hat{k}_x(0) = 0 \\
  \hat{k}_r &= -2re, \quad \hat{k}_r(0) = 0 \\
  \hat{\theta} &= -2\Phi(x)e, \quad \hat{\theta}(0) = 0
  \end{align*}
  \]

- **Parameter Adaptation:**

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

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

Good Tracking & Parameter Estimation
1st-Order *Nonlinear* System with *Local* Nonlinearity: MRAC Simulation, (continued)

Nonlinearity: Good Parameter Estimation
1\textsuperscript{st}-Order Nonlinear System with Local Nonlinearity: MRAC Simulation, (completed)

Nonlinearity: Good Function Approximation
MRAC of a 1\textsuperscript{st}-Order \textit{Nonlinear} System
Conclusions & Observations

- \textit{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 \textit{local} nonlinearities
- A control strategy that depends on parameter convergence, (such as \textit{indirect} MRAC), is unreliable, unless PE condition takes place
MRAC Design of \( n^{th} \) Order Systems

- **System Dynamics:**
  \[
  \dot{x} = Ax + B\Lambda(u - f(x)), \quad x \in \mathbb{R}^n, \quad u \in \mathbb{R}^m
  \]
  - \( A \in \mathbb{R}^{n \times n}, \ \Lambda = \text{diag}(\lambda_1, \ldots, \lambda_m) \in \mathbb{R}^{m \times m} \) are constant *unknown* matrices
  - \( B \in \mathbb{R}^{n \times m} \) is *known* constant matrix
  - \( \forall i = 1, \ldots, m \) \( \text{sgn}(\lambda_i) \) is *known*
  - *uncertain matched* nonlinear function:
    \[
    f(x) = \Theta^T \Phi(x) \in \mathbb{R}^m
    \]
    - *matrix* of constant *unknown* parameters: \( \Theta \in \mathbb{R}^{m \times N} \)
    - vector of \( N \) *known* basis functions: \( \Phi(x) = (\varphi_1(x), \ldots, \varphi_N(x))^T \)

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

- **Control Goal**
  - find \( u \) such that:
    \[
    \lim_{t \to \infty} \|x(t) - x_m(t)\| = 0
    \]
MRAC Design of $n^{th}$ Order Systems (continued)

- **Control Feedback:**
  \[ u = \hat{K}_x^T x + \hat{K}_r^T r + \Theta^T \Phi(x) \]
  - $(mn + m^2 + mN)$ - parameters to estimate: $\hat{K}_x, \hat{K}_r, \Theta$

- **Closed-Loop System:**
  \[ \dot{x} = \left( A + B \Lambda \hat{K}_x^T \right) x + B \Lambda \left( \hat{K}_r^T r + \left( \Theta - \Theta \right)^T \Phi(x) \right) \]

- **Desired Dynamics:**
  \[ \dot{x}_m = A_m x_m + B_m r \]

- **Model Matching Conditions**
  - there exist *ideal* gains $(K_x, K_r)$ such that:
    \[ A + B \Lambda K_x^T = A_m \]
    \[ B \Lambda K_r^T = B_m \]
  - *Note:* knowledge of the ideal gains is not required

\[
\begin{align*}
A + B \Lambda \hat{K}_x^T - A_m &= A + B \Lambda \hat{K}_x^T - A - B \Lambda K_x^T = B \Lambda \left( \hat{K}_x - K_x \right)^T = B \Lambda \Delta K_x^T \\
B \Lambda \hat{K}_r^T - B_m &= B \Lambda \hat{K}_r^T - B \Lambda K_r^T = B \Lambda \left( \hat{K}_r - K_r \right)^T = B \Lambda \Delta \hat{K}_r^T
\end{align*}
\]

E. Lavretsky
MRAC Design of $n^{th}$ Order Systems (continued)

- **Tracking Error:**
  \[ e(t) = x(t) - x_m(t) \]

- **Error Dynamics:**

\[
\dot{e}(t) = \dot{x}(t) - \dot{x}_m(t) = \\
\left( A + B \Lambda \hat{K}_x^T \right) x + B \Lambda \left( \hat{K}_r^T r + \left( \Theta - \Theta \right)^T \Phi(x) \right) - A_m x_m - B_m r \pm A_m x \\
= A_m (x - x_m) + \left( A + B \Lambda \hat{K}_x^T - A_m \right) x + B \Lambda \left( \hat{K}_r - K_r \right)^T r + B \Lambda \Delta \Theta^T \Phi(x) \\
= A_m e + B \Lambda \left( \Delta K_x^T x + \Delta K_r^T r + \Delta \Theta^T \Phi(x) \right)
\]
MRAC Design of $n^{th}$ Order Systems (continued)

- **Lyapunov Function Candidate**

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

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

\[
P A_m + A_m^T P = -Q
\]

- \( Q = Q^T > 0 \) is any symmetric positive definite matrix
MRAC Design of $n^{th}$ Order Systems (continued)

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

\[
\begin{align*}
\dot{K}_x &= -\Gamma_x x^T P B \text{sgn}(\Lambda) \\
\dot{K}_r &= -\Gamma_r r^T P B \text{sgn}(\Lambda) \\
\dot{\Theta} &= -\Gamma_\Theta \Phi(x) e^T P B \text{sgn}(\Lambda)
\end{align*}
\]

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

\[
\dot{V}(e(t), \Delta K_x(t), \Delta K_r(t), \Delta \Theta(t)) = -e^T(t) Q e(t) \leq 0
\]
MRAC Design of $n^{th}$ Order Systems (completed)

- Using Barbalat’s and Lyapunov-like Lemmas: $\lim_{t \to \infty} \dot{V}(x, t) = 0$
- Since $\dot{V} = -e^T(t)Qe^T(t)$ it follows that: $\lim_{t \to \infty} \|e(t)\| = 0$
- Conclusions
  - achieved asymptotic tracking: $x(t) \to x_m(t)$, as $t \to \infty$
  - all signals in the closed-loop system are bounded
- **Remark**
  - 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
  - $\varepsilon$ is the size of the dead-zone.
- Outcome
  - Bounded Tracking

\[
\begin{align*}
\dot{\hat{K}}_x &= \begin{cases} 
-\Gamma_x x e^T P B \text{sgn}(\Lambda), & ||e|| > \varepsilon \\
0, & ||e|| \leq \varepsilon
\end{cases} \\
\dot{\hat{K}}_r &= \begin{cases} 
-\Gamma_r r e^T P B \text{sgn}(\Lambda), & ||e|| > \varepsilon \\
0, & ||e|| \leq \varepsilon
\end{cases} \\
\dot{\Theta} &= \begin{cases} 
-\Gamma \Phi(x) e^T P B \text{sgn}(\Lambda), & ||e|| > \varepsilon \\
0, & ||e|| \leq \varepsilon
\end{cases}
\end{align*}
\]
1st-Order Linear System with Noise
MRAC w/o Dead-Zone: $r(t) = 4$

- Satisfactory Tracking
- Parameter Drift due to measurement noise
1\textsuperscript{st}-Order \textit{Linear} System with Noise
MRAC \textit{with} Dead-Zone: $r(t) = 4$

- Satisfactory Tracking
- No Parameter Drift
Parametric and Non-Parametric Uncertainties

- Parametric Uncertainties are often easy to characterize
  - Example: \( m \ddot{x} = u \)
    - uncertainty in mass \( m \) is parametric
    - neglected motor dynamics, measurement noise, sensor dynamics are non-parametric uncertainties

- Both Parametric and Non-Parametric Uncertainties occur during Function Approximation
  \[
  \hat{f}(x) = \sum_{i=1}^{N} \theta_i \varphi_i(x) + \varepsilon(x)
  \]
  - parametric
  - non-parametric
Enforcing Robustness in MRAC Systems

• Non-Parametric Uncertainty
  – Dead-Zone modification
  – Others?

• Parametric Uncertainty
  – Need a set of basis functions that can approximate a large class of functions within a given tolerance
    • Fourier series
    • Splines
    • Polynomials
    • Artificial Neural Networks
      – sigmoidal
      – RBF
Artificial Neural Networks
• **Artificial Neural Networks** are multi-input-multi-output systems composed of many interconnected nonlinear processing elements (neurons) operating in parallel.
Single Hidden Layer (SHL) Feedforward Neural Networks (FNN)

- Three distinct characteristics
  - model of each neuron includes a nonlinear activation function
    - sigmoid
    - radial basis function
      - a single layer of $N$ hidden neurons
      - feedforward connectivity

\[
\sigma(s) = \frac{1}{1 + e^{-s}}
\]

\[
\phi(x) = e^{-\frac{\|x-r\|^2}{2\sigma^2}}
\]
SHL FNN Architecture

Hidden Layer of $N$ neurons

Input $x$
Threshold
Output Bias
Output $y$
SHL FNN Function

• Maps $n$-dimensional input into $m$-dimensional output: $x \rightarrow NN(x)$, $x \in R^n$, $NN(x) \in R^m$

• Functional Dependence
  
  – sigmoidal: $NN(x) = W^T \tilde{\sigma}(V^T x + \theta) + b$
  
  – RBF:

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

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

- Vector of hidden layer sigmoids:
  \[ \tilde{\sigma}\left(V^T x + \theta\right) = \left(\sigma\left(v_1^T x + \theta_1\right) \quad \ldots \quad \sigma\left(v_N^T x + \theta_N\right)\right)^T \]

- Matrix of inner-layer weights:
  \[ V = \begin{pmatrix} \tilde{v}_1 & \ldots & \tilde{v}_N \end{pmatrix} \in R^{n \times N} \]

- Matrix of output-layer weights:
  \[ W = \begin{pmatrix} \tilde{w}_1 & \ldots & \tilde{w}_m \end{pmatrix} \in R^{N \times m} \]

- Vector of output biases and thresholds
  \[ c \in R^m \quad \theta \in R^N \]

- \(k^{th}\) output:
  \[ \tilde{NN}_k(x) = \tilde{w}_k^T \sigma\left(\tilde{v}_k^T x + \theta_k\right) + c_k = \sum_{j=1}^{N} w_{jk} \sigma\left(\sum_{i=1}^{n} v_{ik} x_i + \theta_k\right) + c_k \]
Sigmoidal NN, (continued)

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

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

\[ \left\| f(x) - W^T \tilde{\sigma} \left( V^T \begin{pmatrix} x \\ 1 \end{pmatrix} \right) - b \right\| \leq \varepsilon = O \left( \frac{1}{\sqrt{N}} \right) \]

- Introduce:

\[ W \triangleq \begin{bmatrix} W^T & b^T \end{bmatrix}, \quad V \triangleq \begin{bmatrix} V^T & \theta \end{bmatrix}, \quad \tilde{\sigma} \triangleq \begin{bmatrix} \tilde{\sigma} \\ 1 \end{bmatrix}, \quad \mu \triangleq \begin{bmatrix} x \\ 1 \end{bmatrix} \]

- Then:

\[ NN(x) = W^T \tilde{\sigma} \left( V^T \mu \right) \]
Sigmoidal SHL NN: Summary

• A very large class of functions can be approximated using linear combinations of shifted and scaled sigmoids

• NN approximation error decreases as the number of hidden-layer neurons $N$ increases:

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

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

$$
NN(x) = W^T \tilde{\sigma}(V^T \mu)
$$

• Function approximation using sigmoidal NN means finding connection weights $W$ and $V$
RBF NN

- Matrix form: \( NN(x) = W^T \Phi(x) + b \)

- Vector of RBF-s:

\[
\Phi(x) = \begin{pmatrix}
\frac{-\|x-C_1\|^2}{2\sigma_1^2} \\
\ddots \\
\frac{-\|x-C_N\|^2}{2\sigma_N^2}
\end{pmatrix}^T
\]

- Matrix of RBF centers:

\[
C \triangleq \begin{bmatrix}
\tilde{C}_1 \\
\vdots \\
\tilde{C}_N
\end{bmatrix} \in R^{n \times N}
\]

- Vector of RBF widths:

\[
\bar{\sigma} \triangleq \begin{pmatrix}
\sigma_1 \\
\vdots \\
\sigma_N
\end{pmatrix}^T \in R^N
\]

- Matrix of output weights:

\[
W = \begin{pmatrix}
\tilde{w}_1 \\
\vdots \\
\tilde{w}_m
\end{pmatrix} \in R^{N \times m}
\]

- Vector of output biases:

\[
b \in R^m
\]

- \( k \)th output:

\[
NN_k(x) = \tilde{w}_k^T \Phi(x) + b_k = \sum_{j=1}^{N} w_{jk} e^{\frac{-\|x-C_j\|^2}{2\sigma_j^2}} + b_k
\]
RBF NN, (continued)

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

\[ \forall f(x) : \mathbb{R}^n \rightarrow \mathbb{R}^m \quad \forall \varepsilon > 0 \quad \exists N, W, \tilde{C}, \tilde{\sigma} \quad \forall x \in X \subset \mathbb{R}^n \]

\[ \| f(x) - W^T \Phi(x) - b \| \leq \varepsilon = O \left( N^{-\frac{1}{n}} \right) \]

- **Introduce:**

\[ W \triangleq [W \ b], \quad \Phi(x) \triangleq \begin{bmatrix} \Phi(x) \\ 1 \end{bmatrix} \]

- **Then:**

\[ NN(x) = W^T \Phi(x) \]
RBF NN: Summary

- A very large class of functions can be approximated using *linear combinations of shifted and scaled gaussians*

- NN approximation error decreases as the number of hidden-layer neurons $N$ increases:

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

- Inclusion of biases into NN output weight matrix simplifies bookkeeping

\[
NN(x) = W^T \Phi(x)
\]

- Function approximation using RBF NN means finding output weights $W$, centers $C$, and widths $\sigma$
What is Next?

• Use SHL FNN-s in the context of MRAC systems
  – off-line / on-line approximation of uncertain nonlinearities in system dynamics
    • modeling errors, (aerodynamics)
    • battle damage
    • control failures

• Start with fixed widths RBF NN architectures, (linear in unknown parameters)

• Generalize to using sigmoidal NN-s
Adaptive NeuroControl
$n^{th}$ Order Systems with Matched Uncertainties

- **System Dynamics:** \[ \dot{x} = Ax + B \Lambda (u - f(x)), \quad x \in \mathbb{R}^n, \quad u \in \mathbb{R}^m \]
  - $A \in \mathbb{R}^{n \times n}$, $\Lambda = \text{diag}(\lambda_1, \ldots, \lambda_m) \in \mathbb{R}^{m \times m}$ are constant *unknown* matrices
  - $B \in \mathbb{R}^{n \times m}$ is *known* constant matrix
  - $\forall i = 1, \ldots, m \quad \text{sgn}(\lambda_i)$ is *known*

- **Approximation of uncertainty:** \[ f(x) = \Theta^T \Phi(x) + \varepsilon_f(x) \]
  - *matrix* of constant *unknown* parameters: $\Theta \in \mathbb{R}^{m \times N}$
  - vector of $N$ *fixed* RBF-s: $\Phi(x) = (\phi_1(x), \ldots, \phi_N(x))^T$
  - function approximation tolerance: $\varepsilon_f(x) \in \mathbb{R}^m$
\( n^{th} \) Order Systems with Matched Uncertainties, (continued)

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

\[
\| \varepsilon_f(x) \| = \| f(x) - \Theta^T \Phi(x) \| \leq \varepsilon, \quad \forall x \in X \subset \mathbb{R}^n
\]

- RBF NN estimator:

\[
\hat{f}(x) = \hat{\Theta}^T \Phi(x)
\]

- Estimation error:

\[
NN(x) - f(x) = \left( \hat{\Theta} - \Theta \right)^T \Phi(x) - \varepsilon_f(x) = \Delta \Theta^T \Phi(x) - \varepsilon_f(x)
\]
$n^{th}$ Order Systems with Matched Uncertainties, (continued)

- **Stable Reference Model:**
  \[
  \dot{x}_m = A_m x_m + B_m r, \quad (A_m \text{ is Hurwitz})
  \]
  \[
  r \in \mathbb{R}^m, \quad A_m \in \mathbb{R}^{n \times n}, \quad B_m \in \mathbb{R}^{m \times m}
  \]

- **Control Goal**
  
  – bounded tracking: \[
  \lim_{t \to \infty} \|x(t) - x_m(t)\| \leq \varepsilon_x
  \]

- **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
$n^{\text{th}}$ Order Systems with Matched Uncertainties, (continued)

- **Control Feedback:**
  
  $\begin{aligned}
  u &= \hat{K}_x^T x + \hat{K}_r^T r + \hat{\Theta}^T \Phi(x) \\
  \hat{K}_x, \hat{K}_r, \hat{\Theta}
  \end{aligned}$

- **Closed-Loop:**
  
  \[
  \dot{x} = (A + B \Lambda \hat{K}_x^T) x + B \Lambda (\hat{K}_r^T r + \Delta \hat{\Theta}^T \Phi(x) - \varepsilon_f(x))
  \]

- **Desired Dynamics:**
  
  \[
  \dot{x}_m = A_m x_m + B_m r
  \]

- **Model Matching Conditions**
  
  - there exist *ideal* gains $(K_x, K_r)$ such that:

  \[
  \begin{cases}
  A + B \Lambda K_x^T = A_m \\
  B \Lambda K_r^T = B_m
  \end{cases}
  \]

  - **Note:** knowledge of the ideal gains is not required

  \[
  \begin{aligned}
  A + B \Lambda \hat{K}_x^T - A_m &= A + B \Lambda \hat{K}_x^T - A - B \Lambda K_x^T = B \Lambda (\hat{K}_x - K_x)^T = B \Lambda \Delta K_x^T \\
  B \Lambda \hat{K}_r^T - B_m &= B \Lambda \hat{K}_r^T - B \Lambda K_r^T = B \Lambda (\hat{K}_r - K_r)^T = B \Lambda \Delta \hat{K}_r^T
  \end{aligned}
  \]
\(n^{th}\) Order Systems with Matched Uncertainties, (continued)

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

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

- Remarks
  - estimation error \(\varepsilon_f(x)\) is bounded, as long as \(x \in X\)
  - need to keep \(x\) within \(X\)
$n^{th}$ Order Systems with Matched Uncertainties, (continued)

- **Lyapunov Function Candidate**

$$V(e, \Delta K_x, \Delta K_r, \Delta \Theta) = e^T P e$$

$$+ \text{trace} \left( \Delta K_x^T \Gamma_x^{-1} \Delta K_x |\Lambda| \right) + \text{trace} \left( \Delta K_r^T \Gamma_r^{-1} \Delta K_r |\Lambda| \right) + \text{trace} \left( \Delta \Theta^T \Gamma_{\Theta}^{-1} \Delta \Theta |\Lambda| \right)$$

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

- Time-derivative of the Lyapunov function

\[
\dot{V} = e^T P e + e^T P \dot{e} \\
+ 2 \text{trace}(\Delta K_x^T \Gamma_x^{-1} \dot{K}_x | \Lambda|) + 2 \text{trace}(\Delta K_r^T \Gamma_r^{-1} \dot{K}_r | \Lambda|) + 2 \text{trace}(\Delta \Theta^T \Gamma^{-1}_\Theta \dot{\Theta} | \Lambda|)
\]

\[
= \left( A_m e + B \Lambda \left( \Delta K_x^T x + \Delta K_r^T r + \Delta \Theta^T \Phi(x) - \epsilon_f(x) \right) \right)^T P e
\]

\[
+ e^T P \left( A_m e + B \Lambda \left( \Delta K_x^T x + \Delta K_r^T r + \Delta \Theta^T \Phi(x) - \epsilon_f(x) \right) \right)
\]

\[
+ 2 \text{trace}(\Delta K_x^T \Gamma_x^{-1} \dot{K}_x | \Lambda|) + 2 \text{trace}(\Delta K_r^T \Gamma_r^{-1} \dot{K}_r | \Lambda|) + 2 \text{trace}(\Delta \Theta^T \Gamma^{-1}_\Theta \dot{\Theta} | \Lambda|)
\]

\[
= e^T \left( A_m P + P A_m \right) e
\]

\[
+ 2 e^T P B \Lambda \left( \Delta K_x^T x + \Delta K_r^T r + \Delta \Theta^T \Phi(x) - \epsilon_f(x) \right)
\]

\[
+ 2 \text{trace}(\Delta K_x^T \Gamma_x^{-1} \dot{K}_x | \Lambda|) + 2 \text{trace}(\Delta K_r^T \Gamma_r^{-1} \dot{K}_r | \Lambda|) + 2 \text{trace}(\Delta \Theta^T \Gamma^{-1}_\Theta \dot{\Theta} | \Lambda|)
\]
$n^{th}$ Order Systems with Matched Uncertainties, (continued)

- Time-derivative of the Lyapunov function

\[
\dot{V} = -e^T Q e - 2 e^T P B \Lambda \varepsilon_f(x)
\]

\[
+ 2 e^T P B \Lambda \Delta K_x^T x + 2 \text{trace} \left( \Delta K_x^T \Gamma_x^{-1} \hat{K}_x | \Lambda \right)
\]

\[
+ 2 e^T P B \Lambda \Delta K_r^T r + 2 \text{trace} \left( \Delta K_r^T \Gamma_r^{-1} \hat{K}_r | \Lambda \right)
\]

\[
+ 2 e^T P B \Lambda \Delta \theta^T \Phi(x) + 2 \text{trace} \left( \Delta \theta^T \Gamma_{\theta}^{-1} \hat{\theta} | \Lambda \right)
\]

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

- Example:

\[
\left( e^T P B \Lambda \Delta K_x^T x \right) = \text{trace} \left( \Delta K_x^T x e^T P B \Lambda \right)
\]
\( n^{\text{th}} \) Order Systems with Matched Uncertainties, (continued)

- Time-derivative of the Lyapunov function

\[
\dot{V} = -e^T Q e - 2 e^T P B \Lambda \varepsilon_f(x)
\]

\[
+2 \text{trace} \left( \Delta K_x^T \left\{ \Gamma_x^{-1} \dot{K}_x + x e^T P B \text{sgn}(\Lambda) \right\} |\Lambda| \right)
\]

\[
+2 \text{trace} \left( \Delta K_r^T \left\{ \Gamma_r^{-1} \dot{K}_r + r e^T P B \text{sgn}(\Lambda) \right\} |\Lambda| \right)
\]

\[
+2 \text{trace} \left( \Delta \Theta^T \left\{ \Gamma_\Theta^{-1} \dot{\Theta} + \Phi(x) e^T P B \text{sgn}(\Lambda) \right\} |\Lambda| \right)
\]

- Problem
  - choose adaptive parameters \( \hat{K}_x, \hat{K}_r, \hat{\Theta} \) such that time-derivative \( \dot{V} \) becomes negative definite outside of a compact set in the error state space, and all parameters remain bounded for all future times
$n^{th}$ Order Systems with Matched Uncertainties, (continued)

- Suppose that we choose adaptive laws:
  \[
  \dot{\hat{K}}_x = -\Gamma_x x e^T P B \text{sgn}(\Lambda)
  \]
  \[
  \dot{\hat{K}}_r = -\Gamma_r r e^T P B \text{sgn}(\Lambda)
  \]
  \[
  \dot{\hat{\Theta}} = -\Gamma_{\Theta} \Phi(x) e^T P B \text{sgn}(\Lambda)
  \]

- Then we get:
  \[
  \dot{V} = -e^T Q e - 2 e^T P B \Lambda \epsilon_f(x) \leq -\lambda_{\min}(Q)\|e\|^2 + 2\|e\|\|P B\|\lambda_{\max}(\Lambda)\epsilon
  \]

- Consequently, $\dot{V} < 0$ outside of the compact set
  \[
  E \triangleq \left\{ e : \|e\| \leq \frac{2\|P B\|\lambda_{\max}(\Lambda)\epsilon}{\lambda_{\min}(Q)} \right\}
  \]

- Unfortunately, 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:
  \[
  \dot{\hat{K}}_x = -\Gamma_x \left( x e^T P B + \sigma_x \hat{K}_x \right) \text{sgn}(\Lambda) \\
  \dot{\hat{K}}_r = -\Gamma_r \left( r e^T P B + \sigma_r \hat{K}_r \right) \text{sgn}(\Lambda) \\
  \dot{\hat{\Theta}} = -\Gamma_\Theta \left( \Phi(x) e^T P B + \sigma_\Theta \hat{\Theta} \right) \text{sgn}(\Lambda)
  \]

• **e** - modification:
  \[
  \dot{\hat{K}}_x = -\Gamma_x \left( x e^T P B + \sigma_x \| e^T P B \| \hat{K}_x \right) \text{sgn}(\Lambda) \\
  \dot{\hat{K}}_r = -\Gamma_r \left( r e^T P B + \sigma_r \| e^T P B \| \hat{K}_r \right) \text{sgn}(\Lambda) \\
  \dot{\hat{\Theta}} = -\Gamma_\Theta \left( \Phi(x) e^T P B + \sigma_\Theta \| e^T P B \| \hat{\Theta} \right) \text{sgn}(\Lambda)
  \]

• Modifications add **damping** to adaptive laws
  – damping controlled by choosing \( \sigma_x, \sigma_r, \sigma_\Theta > 0 \)
  – there is a **trade off** between adaptation rate and damping
Introducing Projection Operator

- Requires no damping terms
- Designed to keep NN weights within \textit{prespecified} bounds
- Maintains negative values of the Lyapunov function time-derivative outside of compact subset:

$$E \triangleq \left\{ e : \|e\| \leq \frac{2\|PB\|\max(\Lambda)\varepsilon}{\min(Q)} \right\}$$

- the size of $E$ defines tracking tolerance
- the size of $E$ can be controlled!
Projection Operator

- Function \( f(\theta) \) defines prespecified parameter domain boundary
- Example:

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

\[
\begin{align*}
\{ f(\theta) \leq 0 \} & \Rightarrow \{ \|\theta\| \leq \theta_{max} \} \Rightarrow \theta \text{ is within bounds} \\
\{ 0 < f(\theta) \leq 1 \} & \Rightarrow \{ \|\theta\| \leq \sqrt{1 + \varepsilon_\theta \theta_{max}} \} \Rightarrow \theta \text{ is within } \left(\sqrt{1 + \varepsilon_\theta}\right) \% \text{ of bounds} \\
\{ f(\theta) > 1 \} & \Rightarrow \{ \|\theta\| > \sqrt{1 + \varepsilon_\theta \theta_{max}} \} \Rightarrow \theta \text{ is outside of bounds}
\end{align*}
\]

- \( \theta_{max} \) specifies boundary
- \( \varepsilon_\theta \) specifies boundary tolerance
Projection Operator, (continued)

**Definition:**

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

- Depends on \((\theta, y)\)
- Does not alter \(y\) if \(\theta\) is within the pre-specified bounds: \(\|\theta\| \leq \theta_{\text{max}}\)
- Gradient: \(\nabla f(\theta) = \frac{2\theta}{\varepsilon_\theta \theta_{\text{max}}^2}\)
- In \(\{0 \leq f(\theta) \leq 1\}\) 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**

\[(\theta - \theta^*)^T (\text{Proj}(\theta, y) - y) \leq 0\]
Lyapunov Function Time-Derivative with Projection Operator

- Make trace terms semi-negative \textbf{AND} keep parameters bounded:

$$\dot{V} = -e^T Q e - 2e^T P B \Lambda \varepsilon_f(x)$$

$$+ 2 \text{ trace } \begin{cases} \Delta K^T_x \\ \Gamma_x^{-1} \dot{K}_x + x e^T P B \text{ sgn} \left( \Lambda \right) \end{cases} \begin{cases} \text{Proj}(\hat{K}_x, y) \end{cases}$$

$$+ 2 \text{ trace } \begin{cases} \Delta K^T_r \\ \Gamma_r^{-1} \dot{K}_r + r e^T P B \text{ sgn} \left( \Lambda \right) \end{cases} \begin{cases} \text{Proj}(\hat{K}_r, y) \end{cases}$$

$$+ 2 \text{ trace } \begin{cases} \Delta \Theta^T \\ \Gamma_\Theta^{-1} \dot{\Theta} + \Phi(x) e^T P B \text{ sgn} \left( \Lambda \right) \end{cases} \begin{cases} \text{Proj}(\hat{\Theta}, y) \end{cases}$$
Adaptation with Projection

- Modified adaptive laws:
  \[ \dot{K}_x = \Gamma_x \text{Proj}\left( \hat{K}_x, -x e^T P B \text{sgn}(\Lambda) \right) \]
  \[ \dot{K}_r = \Gamma_r \text{Proj}\left( \hat{K}_r, -r e^T P B \text{sgn}(\Lambda) \right) \]
  \[ \dot{\Theta} = \Gamma_\Theta \text{Proj}\left( \hat{\Theta}, -\Phi(x) e^T P B \text{sgn}(\Lambda) \right) \]

- Projection Operator, its bounds and tolerances are defined \textit{column-wise}

- Lyapunov function time-derivative:
  \[ \dot{V} \leq -e^T Q e - 2 e^T P B \Lambda \varepsilon_f(x) \leq -\lambda_{\min}(Q)\|e\|^2 + 2\|e\|\|P B\|\lambda_{\max}(\Lambda)\varepsilon \]

- Adaptive parameters stay within the pre-specified bounds, while \( \dot{V} < 0 \) outside of the compact set:
  \[ E \triangleq \left\{ e : \|e\| \leq \frac{2\|P B\|\lambda_{\max}(\Lambda)\varepsilon}{\lambda_{\min}(Q)} \right\} \]
Example: Projection Operator, (scalar case)

- Scalar adaptive gain: \[ \hat{k} = \gamma \operatorname{Proj}(\hat{k}, -x e \text{sgn}(b)) \]

- Pre-specified parameter domain boundary:
  - using function: \[ f(\hat{k}) = \frac{\hat{k}^2 - k_{\text{max}}^2}{\varepsilon k_{\text{max}}^2} \]
  - Gradient: \[ \nabla f(\hat{k}) = f'(\hat{k}) = \frac{2\hat{k}}{\varepsilon k_{\text{max}}^2} \]

\[
\begin{align*}
\{ f(\hat{k}) \leq 0 \} & \Rightarrow \{ |\hat{k}| \leq k_{\text{max}} \} \Rightarrow \hat{k} \text{ is within bounds} \\
\{ 0 < f(\hat{k}) \leq 1 \} & \Rightarrow \{ |\hat{k}| \leq \sqrt{1 + \varepsilon} k_{\text{max}} \} \Rightarrow \hat{k} \text{ is within } (\sqrt{1 + \varepsilon})\% \text{ of bounds} \\
\{ f(\hat{k}) > 1 \} & \Rightarrow \{ |\hat{k}| > \sqrt{1 + \varepsilon} k_{\text{max}} \} \Rightarrow \hat{k} \text{ is outside of bounds}
\end{align*}
\]

- Projection Operator:
  \[ y = -x e \text{sgn}(b) \]
  \[ \operatorname{Proj}(\hat{k}, y) = \begin{cases} 
  y(1 - f(\hat{k})), & \text{if } f(\hat{k}) > 0 \text{ and } y f'(\hat{k}) > 0 \\
  y, & \text{if not}
\end{cases} \]
**Example: Projection Operator, (scalar case) (continued)**

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

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

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

- Geometric Interpretation
  - adaptive parameter \(\hat{k}(t)\) changes within the pre-specified interval
  - interval bound: \(k_{\text{max}}\)
  - Bound tolerance: \(\varepsilon\)
Adaptive Augmentation Design

- **Nominal Control:**
  \[ u_{\text{nom}} = F^T_x x + F^T_r r \]

- **Adaptive Control:**
  \[ u = \hat{K}^T_x x + \hat{K}^T_r r + \Theta^T \Phi(x) \]

- **Augmentation:**
  \[ u = \hat{K}^T_x x + \hat{K}^T_r r + \Theta^T \Phi(x) \pm u_{\text{nom}} \]
  \[ = u_{\text{nom}} + \underbrace{\left( \hat{K}^T_x - F^T_x \right) x} + \underbrace{\left( \hat{K}^T_r - F^T_r \right) r} + \Theta^T \Phi(x) \]
  \[ = u_{\text{nom}} + \hat{D}^T_x x + \hat{D}^T_r r + \Theta^T \Phi(x) \]

- **Incremental Adaptation:**
  \[ \dot{\hat{D}}_x = \Gamma_x \text{Proj}\left( \hat{D}_x, -x e^T P B \text{sgn}(\Lambda) \right), \quad \hat{D}_x = 0_{n \times m} \]
  \[ \dot{\hat{D}}_r = \Gamma_r \text{Proj}\left( \hat{D}_r, -r e^T P B \text{sgn}(\Lambda) \right), \quad \hat{D}_r = 0_{m \times m} \]
  \[ \dot{\Theta} = \Gamma_\Theta \text{Proj}\left( \Theta, -\Phi(x) e^T P B \text{sgn}(\Lambda) \right), \quad \Theta = 0_{N \times m} \]
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 \textbf{Sigmoidal} NN

- **System Dynamics:**
  \[
  \dot{x} = Ax + B \Lambda \left( u - f(x) \right), \quad x \in \mathbb{R}^n, \quad u \in \mathbb{R}^m
  \]
  - \( A \in \mathbb{R}^{n \times n} \), \( \Lambda = \text{diag}(\lambda_1, \ldots, \lambda_m) \in \mathbb{R}^{m \times m} \) are constant \textit{unknown} matrices
  - \( B \in \mathbb{R}^{M \times m} \) is \textbf{known} constant matrix, and \( M \geq m \)
  - \( \forall i = 1, \ldots, m \) \( \text{sgn}(\lambda_i) \) is \textbf{known}

- **Approximation of uncertainty:**
  \[
  f(x) = W^T \tilde{\sigma}(V^T \mu) + \varepsilon_f(x), \quad \mu = \begin{pmatrix} x^T & 1 \end{pmatrix}^T, \quad \varepsilon_f(x) \in \mathbb{R}^m
  \]
  - matrix of constant \textit{unknown} \textbf{Inner-Layer} weights:
    \[
    V = \begin{bmatrix} \tilde{v}_1 & \cdots & \tilde{v}_N \\ \theta_1 & \cdots & \theta_N \end{bmatrix} \in \mathbb{R}^{(n+1) \times N}
    \]
  - matrix of constant \textit{unknown} \textbf{Outer-Layer} weights:
    \[
    W = \begin{bmatrix} \tilde{w}_1 & \cdots & \tilde{w}_m \\ c_1 & \cdots & c_m \end{bmatrix} \in \mathbb{R}^{(N+1) \times m}
    \]
  - vector of \( N \) \textit{sigmoid}s and a unity:
    \[
    \tilde{\sigma}(V^T \mu) = \begin{pmatrix} \sigma(\tilde{v}_1^T x + \theta_1) & \cdots & \sigma(\tilde{v}_N^T x + \theta_N) \end{pmatrix}^T, \quad \text{where:} \quad \sigma(s) = \frac{1}{1 + e^{-s}}
    \]
Adaptive Control using **Sigmoidal** NN

- Control Feedback:

  \[ u = \hat{K}_x^T x + \hat{K}_r^T r + \hat{W}^T \hat{\sigma}(\hat{V}^T \mu) \]

  \[- (m n + m^2 + (n + 1) N + (N + 1) m) \text{ - parameters to estimate: } \hat{K}_x, \hat{K}_r, \hat{W}, \hat{V} \]

- Adaptation with Projection, \((\Lambda > 0)\):

  \[
  \begin{align*}
  \dot{\hat{K}}_x &= \Gamma_x \text{Proj}(\hat{K}_x, -x e^T P B) \\
  \dot{\hat{K}}_u &= \Gamma_u \text{Proj}(\hat{K}_u, -r e^T P B) \\
  \dot{\hat{W}} &= \Gamma_w \text{Proj}(\hat{W}, (\hat{\sigma}(\hat{V}^T \mu) - \hat{\sigma}'(\hat{V}^T \mu)\hat{V}^T \mu)e^T P B) \\
  \dot{\hat{V}} &= \Gamma_v \text{Proj}(\hat{V}, \mu e^T P B\hat{W}^T \hat{\sigma}'(\hat{V}^T \mu))
  \end{align*}
  \]

- Provides *bounded* tracking
Design Example
Adaptive Reconfigurable Flight Control using RBF NN-s
Aircraft Model

• Flight Dynamics Approximation, (constant speed):

\[
\dot{x}_p = A_p x_p + B G \Lambda (\delta + K_0(x_p)) = A_p x_p + B_p \Lambda (\delta + K_0(x_p))
\]

– State: \( x_p = (\alpha \quad \beta \quad p \quad q \quad r)^T \)

– Control allocation matrix \( G \)

– **Virtual** Control Input: \( \delta \in \mathbb{R}^3 \)

– Modeling control uncertainty / failures by \( \Lambda \in \mathbb{R}^{3x3} \) diagonal matrix with positive elements

– Vector of actual control inputs:

\[
G \Lambda \delta = (\delta_{LOB} \quad \delta_{LMB} \quad \delta_{LIB} \quad \delta_{RIB} \quad \delta_{RMB} \quad \delta_{ROB} \quad \delta_{Tvec})^T \in \mathbb{R}^7
\]

– \( A_p, B_p \) are **known** matrices

  • represent nominal system dynamics

– **Matched** unknown nonlinear effects: \( K_0(x_p) \in \mathbb{R}^3 \)
Baseline Inner-Loop Controller

- **Dynamics:**
  \[ \dot{x}_c = A_c x_c + B_{1c} x_p + B_{2c} u \]

- **States:**
  \[ x_c = (q_I \quad p_I \quad r_I \quad r_w)^T \in \mathbb{R}^4 \]

- **Inner-loop commands, (reference input):**
  \[ u = \left( a_z^{cmd} \quad \beta^{cmd} \quad p^{cmd} \quad r^{cmd} \right)^T \]

- **System output:**
  \[ a_z = C_p x_p + D G \Lambda \left( \delta + K_0 \left( x_p \right) \right) = C_p x_p + D_p \Lambda \left( \delta + K_0 \left( x_p \right) \right) \]

- **Augmented system dynamics:**
  \[
  \begin{bmatrix}
  \dot{x}_p \\
  \dot{x}_c
  \end{bmatrix} =
  \begin{bmatrix}
  A_p & 0 \\
  B_{1c} & A_c
  \end{bmatrix}
  \begin{bmatrix}
  x_p \\
  x_c
  \end{bmatrix} +
  \begin{bmatrix}
  B_p \\
  0
  \end{bmatrix}
  \Lambda \left( \delta + K_0 \left( x_p \right) \right)
  +
  \begin{bmatrix}
  0 \\
  B_{2c}
  \end{bmatrix}
  u
  \]

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

- **Inner-Loop Control:**
  \[ \delta_L = K_x^T x + K_u^T u \]
Reference Model

- Assuming nominal data, \( (\Lambda = I_{3\times3}, \quad K_0(x_p) = 0_{3\times1}) \), and using baseline controller:

\[
\dot{x}_{ref} = \begin{pmatrix} A + B_1 K^T_x \end{pmatrix} x_{ref} + \begin{pmatrix} B_2 + B_1 K^T_u \end{pmatrix} u = A_{ref} x_{ref} + B_{ref} u
\]

- **Assumption**: Reference model matrix \( A_{ref} \) is Hurwitz, (i.e., baseline controller stabilizes nominal system)
Inner-Loop Control Objective
(Bounded Tracking)

- Design virtual control input such that, despite system uncertainties, the system state tracks the state of the reference model, while all closed-loop signals remain bounded

- Solution
  - Incremental, (i.e., adaptive augmentation), MRAC system with RBF NN, Dead-Zone, and Projection Operator
Adaptive Augmentation

- Total control input:

\[
\begin{align*}
\delta &= \hat{K}_x^T x + \hat{K}_u^T u - \hat{K}_0 (x_p) \pm \delta_L(x,u) \\
&= \delta_L(x,u) + \left( \hat{K}_x - K_x \right)^T x + \left( \hat{K}_u - K_u \right)^T u - \hat{K}_0 (x_p) \\
&= \delta_L(x_p, x_c, u) + \Delta \hat{K}_x^T x + \Delta \hat{K}_u^T u - \hat{\Theta}^T \Phi(x_p)
\end{align*}
\]

- Incremental adaptation with projection:

\[
\begin{align*}
\dot{\Delta \hat{K}_x} &= \Gamma_x \ \text{Proj}\left( \Delta \hat{K}_x, -x e^T P B_1 \right), \quad \Delta \hat{K}_x(0) = 0_{n \times 3} \\
\dot{\Delta \hat{K}_u} &= \Gamma_u \ \text{Proj}\left( \Delta \hat{K}_u, -u e^T P B_1 \right), \quad \Delta \hat{K}_u(0) = 0_{n \times 4} \\
\dot{\hat{\Theta}} &= \Gamma_{\Theta} \ \text{Proj}\left( \hat{\Theta}, \Phi(x_p) e^T P B_1 \right), \quad \hat{\Theta}(0) = 0_{N \times m}
\end{align*}
\]
Inner-Loop Block-Diagram

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

- MRAC requires model matching conditions

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

- Example that violates matching
  - System:
    \[
    \begin{pmatrix}
    \dot{x}_1 \\
    \dot{x}_2
    \end{pmatrix} =
    \begin{pmatrix}
    0 & 1 \\
    0 & 0
    \end{pmatrix}
    \begin{pmatrix}
    x_1 \\
    x_2
    \end{pmatrix} +
    \begin{pmatrix}
    0 \\
    1
    \end{pmatrix} u
    \]
  
  - Reference model:
    \[
    \begin{pmatrix}
    \dot{x}_1^m \\
    \dot{x}_2^m
    \end{pmatrix} =
    \begin{pmatrix}
    -1 & 1 \\
    0 & -2
    \end{pmatrix}
    \begin{pmatrix}
    x_1^m \\
    x_2^m
    \end{pmatrix} +
    \begin{pmatrix}
    0 \\
    1
    \end{pmatrix} r
    \]

  Matching conditions don’t hold

\[
A - A_m = \begin{pmatrix}
1 & 0 \\
0 & 2
\end{pmatrix} \neq b k_x^T = \begin{pmatrix}
0 & 0 \\
\ast & \ast
\end{pmatrix}
\]
Control Tracking Problem

• Consider 2\textsuperscript{nd} order cascaded system

\[
\begin{align*}
\dot{x}_1 &= f_1(x_1) + g_1(x_1) x_2 \\
\dot{x}_2 &= f_2(x_1, x_2) + g_2(x_1, x_2) u
\end{align*}
\]

• Control goal
  – Choose \( u \) such that: \( x_1(t) \rightarrow x_1^{\text{com}}(t), \text{ as } t \rightarrow \infty \)

• Assumptions
  – All functions are known
  – \( g_i \neq 0 \) does not cross zero

• Example: AOA tracking

\[
\begin{align*}
\dot{\alpha} &= -L_\alpha(\alpha)\alpha + \frac{1}{g_1} q \\
\dot{q} &= M_0(\alpha, q) + \frac{1}{g_2} \dot{q}_{\text{cmd}}
\end{align*}
\]
Backstepping Design

- Introduce pseudo control: \( x_{2}^{\text{com}} = x_{2}^{\text{com}}(t) \)
- Rewrite the 1\text{st} equation:
  \[
  \dot{x}_1 = f_1(x_1) + g_1(x_1)x_{2}^{\text{com}} + g_1(x_1)(x_2 - x_{2}^{\text{com}}) + \Delta x_2
  \]
- Dynamic inversion using pseudo control:
  \[
  x_{2}^{\text{com}} = \frac{1}{g_1(x_1)}\left(\dot{x}_1^{\text{com}} - f_1(x_1) - k_1 \Delta x_1\right)
  \]
- 1\text{st} state error dynamics:
  \[
  \Delta \dot{x}_1 = -k_1 \Delta x_1 + g_1(x_1) \Delta x_2
  \]
Backstepping Design (continued)

- Dynamic inversion using actual control
  \[ u = \frac{1}{g_2(x_1, x_2)} \left( \dot{x}_2^{com} - f_2(x_1, x_2) - k_2 \Delta x_2 - g_1(x_1) \Delta x_1 \right) \]

- 2\textsuperscript{nd} state error dynamics
  \[ \Delta \dot{x}_2 = -k_2 \Delta x_2 - g_1(x_1) \Delta x_1 \]

- \textit{Asymptotically stable} error dynamics
  \[
  \begin{pmatrix}
  \Delta \dot{x}_1 \\
  \Delta \dot{x}_2
  \end{pmatrix} =
  \begin{pmatrix}
  -k_1 & g_1(x_1) \\
  -g_1(x_1) & -k_2
  \end{pmatrix}
  \begin{pmatrix}
  \Delta x_1 \\
  \Delta x_2
  \end{pmatrix}
  \]

- Conclusion: \( x_i(t) \rightarrow x_i^{com}(t) \), as \( t \rightarrow \infty \)
Adaptive Backstepping Design

• 1st state dynamics: \( \dot{x}_1 = \hat{f}_1 + \hat{g}_1 x_2^{\text{com}} + \hat{g}_1 \Delta x_2 - \Delta f_1 - \Delta g_1 u \)
  
  – Function estimation errors:
  \( \Delta f_1 \triangleq \hat{f}_1 - f_1, \quad \Delta g_1 \triangleq \hat{g}_1 - g_1 \)

• Dynamic inversion using pseudo control and estimated functions:
  \( x_2^{\text{com}} = \frac{1}{\hat{g}_1(x_1)}(\dot{x}_1^{\text{com}} - \hat{f}_1(x_1) - k_1 \Delta x_1) \)

• 1st state error dynamics:
  \( \Delta \dot{x}_1 = -k_1 \Delta x_1 + \hat{g}_1 \Delta x_2 - \Delta f_1 - \Delta g_1 u \)
Adaptive Backstepping Design (continued)

• 2\textsuperscript{nd} 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:
    \[ u = \frac{1}{\hat{g}_2(x_1, x_2)} \left( \dot{x}_{2\text{com}} - \hat{f}_2(x_1, x_2) - k_2 \Delta x_2 - \hat{g}_1(x_1) x_1 \right) \]

• 2\textsuperscript{nd} state error dynamics:
  \[ \Delta \dot{x}_2 = -k_2 \Delta x_2 - \hat{g}_1 \Delta x_1 - \Delta f_2 - \Delta g_2 u \]
Adaptive Backstepping Design (continued)

• Combined error dynamics:

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

• Uncertainty parameterization, function and parameter estimation errors:

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

\[
\Delta \theta_{f_i} \triangleq \hat{\theta}_{f_i} - \theta_{f_i}
\]
\[
\Delta \theta_{g_i} \triangleq \hat{\theta}_{g_i} - \theta_{g_i}
\]
Adaptive Backstepping Design (continued)

• Tracking error dynamics:

\[
\begin{bmatrix}
\Delta \dot{x}_1 \\
\Delta \dot{x}_2
\end{bmatrix} =
\begin{bmatrix}
-k_1 & \hat{g}_1 \\
-\hat{g}_1 & -k_2
\end{bmatrix}
\begin{bmatrix}
\Delta x_1 \\
\Delta x_2
\end{bmatrix} -
\begin{bmatrix}
\Delta \theta_{f_1}^T \\
\Delta \theta_{f_2}^T \\
\Delta \theta_{g_1}^T \\
\Delta \theta_{g_2}^T
\end{bmatrix}
\begin{bmatrix}
\Phi_f \\
\Phi_f \\
\Phi_g u \\
\Phi_g u
\end{bmatrix} +
\begin{bmatrix}
\varepsilon_{f_1} + \varepsilon_{g_1} u \\
\varepsilon_{f_2} + \varepsilon_{g_2} u
\end{bmatrix}
\]

\[
\dot{e} = Ae - \Delta \Theta^T \Phi + \varepsilon
\]

• Stable robust adaptive laws:

\[
\dot{\Theta} = \Gamma \text{Proj}(\hat{\Theta}, \Phi e^T)
\]

• Conclusion: Bounded tracking
Adaptive Control in the Presence of Actuator Constraints

Overview

- **Problem:** Assure stability of an adaptive control system in the presence of actuator position / rate saturation constraints.

- **Solutions**
  - Ad-hoc
  - Proof-by-simulation
    - Lyapunov based
      - limited class of systems
      - sufficient conditions are often hard-to-verify
    
    - not acceptable
  
  - preferred

- **Need:** *Theoretically justified and verifiable* conditions for stable adaptation and control design 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

  - adaptive modifications: reference input
  - rigorous stability proof

  - pseudo control hedging (PCH)
    - fixed gain modification of reference input

  - positive $\mu$ – 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

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

- **Static actuator**
  \[
  u(t) = u_{\max} \text{ sat} \left( \frac{u_c(t)}{u_{\max}} \right) = \begin{cases} 
  u_c(t), & |u_c(t)| \leq u_{\max} \\
  u_{\max} \text{ sgn}(u_c(t)), & |u_c(t)| \geq u_{\max}
  \end{cases}
  \]

- **Ideal Reference model dynamics:**
  \[ \dot{x}_m^*(t) = A_m x_m^*(t) + b_m r(t), \quad x_m^* \in \mathbb{R}^n, r \in \mathbb{R} \]

- **Hurwitz**
- **bounded reference input**
- **amplitude saturation**
- **commanded input**
- **battle damage**
- **control failures**
Preliminaries

- Define: $u^\delta_{\text{max}} = u_{\text{max}} - \delta$, where: $0 < \delta < u_{\text{max}}$

- Commanded control deficiency: $\Delta u_c = u^\delta_{\text{max}} \text{ sat} \left( \frac{u_c}{u^\delta_{\text{max}}} \right) - u_c$

- Adaptive control with $\mu$– modification, (implicit form):
  
  $u_c = k^T_x x + k_r r + \mu \Delta u_c$

- Need explicit form of $u_c$
Positive $\mu$ – modification

- Adaptive control with $\mu$ – mod is given by a **convex combination** of $u_{\text{lin}}$ and $u_{\text{max}}^\delta$ sat\(\frac{u_{\text{lin}}}{u_{\text{max}}^\delta}\)

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

continuous in time but *not* continuously differentiable

$$\delta = 0 \land (\mu = 0 \lor \mu = \infty) \Rightarrow u = u_{\text{max}} \text{ sat} \left( \frac{u_{\text{lin}}}{u_{\text{max}}} \right)$$
Closed-Loop Dynamics

- **μ – mod control:**
  \[ u_c = u_{\text{lin}} + \mu \Delta u_c \]

- **System dynamics:**
  \[ \dot{x} = Ax + b \lambda u_c + b \lambda (u - u_c) \]

- **Closed-loop system:**
  \[ \dot{x} = Ax + b \lambda u_{\text{lin}} + b \lambda (\mu \Delta u_c + \Delta u) \]

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

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

\( \Delta u \) does not depend on \( \mu \) explicitly

Linear control deficiency

Closed-loop dynamics
Adaptive Reference Model Modification

- Closed-loop system:
  \[ \dot{x} = (A + b \lambda k_x^T) x + b \lambda (k_r r + \Delta u_{\text{lin}}) \]

- Leads to consideration of **adaptive** reference model:
  \[ \dot{x}_m = A_m x_m + b_m \left( r(t) + k_u \Delta u_{\text{lin}} \right), \quad |r(t)| \leq r_{\text{max}} \]

- Matching conditions:
  \[ \forall \lambda > 0 \exists \left( k_x^* \in R^n, \ k_r^* \in R, \ k_u^* \in R \right) \]
  \[ \begin{align*}
  A + b \lambda (k_x^T)^* &= A_m \\
  b \lambda k_r^* &= b_m \\
  b \lambda = b_m k_u^*
  \end{align*} \Rightarrow k_u^* k_r^* = 1 \]
Adaptive Laws Derivation

- **Tracking error:**  \( e = x - x_m \)
- **Parameter errors:**
  \[
  \begin{align*}
  \Delta k_x &= k_x - k_x^* \\
  \Delta k_r &= k_r - k_r^* \\
  \Delta k_u &= k_u - k_u^*
  \end{align*}
  \]

- **Tracking error dynamics:**
  \[
  \dot{e} = A_m e + b \lambda \left( \Delta k_x^T x + \Delta k_r r \right) - b_m \Delta k_u \Delta u_{lin}
  \]

- **Lyapunov function:**
  \[
  V \left( e, \Delta k_x, \Delta k_r, \Delta k_u \right) = e^T P e + \lambda \left( \Delta k_x^T \Gamma_x^{-1} \Delta k_x + \gamma_r^{-1} \Delta k_r^2 + \gamma_u^{-1} \Delta k_u^2 \right)
  \]
  where:  \( P A_m + A_m P = -Q < 0 \)
Stable Parameter Adaptation

- Adaptive laws derived to yield stability:

\[
\begin{align*}
\dot{k}_x &= -\Gamma_x x e^T P b \\
\dot{k}_r &= -\gamma_r r(t) e^T P b \\
\dot{k}_u &= \gamma_u \Delta u_{\text{lin}} e^T P b_m
\end{align*}
\]

\[\iff \dot{V} = -e^T Q e < 0 \implies \dot{V}(e, \Delta k_x, \Delta k_r, \Delta k_u) \leq 0\]

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

- Choose “safety zone” \( 0 < \delta < u_{\text{max}} \) and sufficiently large \( \mu > 0 \)
- Define virtual constraint: \( u_{\text{max}}^\delta = u_{\text{max}} - \delta \)
- Linear component of adaptive control signal: \( u_{\text{lin}} = k_x^T x + k_r r(t) \)

**Total adaptive control with \( \mu – \text{mod} \):**

\[
\dot{u}_c = \frac{1}{1 + \mu} \left( u_{\text{lin}} + \mu u_{\text{max}}^\delta \text{sat} \left( \frac{u_{\text{lin}}}{u_{\text{max}}} \right) \right)
\]

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

**Adaptive laws**

\[
\begin{align*}
\dot{k}_x &= -\Gamma_x x e^T P b \\
\dot{k}_r &= -\gamma_r r(t) e^T P b \\
\dot{k}_u &= \gamma_u \Delta u_{\text{lin}} e^T P b_m
\end{align*}
\]

**Modified reference model**
Simulation Example

- Unstable open-loop system:

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

- Choose: \( \delta = 0.2 \, u_{\text{max}} \)

- Ideal reference model:

\[ \dot{x}_m = -6 \, (x_m - r(t)) \]

- Reference input:

\[ r(t) = 0.7 \, (\sin(2t) + \sin(0.4t)) \]

- Adaptation rates set to unity

- System and reference model start at zero
Simulation Data

$\mu = 0$

- Tracking Performance: $\mu = 0$
  - System
  - Ref Model
  - Ref model with inv-model

- Control Input: $\mu = 0$
  - Commanded
  - Actual

$\mu = 1$

- Tracking Performance: $\mu = 1$
  - System
  - Ref Model
  - Ref model with inv-model

- Control Input: $\mu = 1$
  - Commanded
  - Actual

$\mu = 10$

- Tracking Performance: $\mu = 10$
  - System
  - Ref Model
  - Ref model with inv-model

- Control Input: $\mu = 10$
  - Commanded
  - Actual

$\mu = 100$

- Tracking Performance: $\mu = 100$
  - System
  - Ref Model
  - Ref model with inv-model

- Control Input: $\mu = 100$
  - Commanded
  - Actual
\[ \mu \text{ – mod Design Summary} \]

- Lyapunov based
- Provides closed-loop stability and bounded tracking
  - convex combination of linear adaptive control and
    its \( u_{\text{max}}^\delta \) – limited value
  - adaptive reference model modification
- Verifiable sufficient conditions
- **Future Work**
  - MIMO systems
  - Dynamic actuators
  - Nonaffine-in-control dynamics
  - Flight control applications
Adaptive Flight Control Applications, Open Problems, and Future Work
Autonomous Formation Flight Flight, (AFF)

References:

AFF: Program Overview

- **Program participants:**
  - NASA Dryden
  - Boeing - Phantom Works
  - UCLA

- **Flight test program**
  - Completed in December of 2001
  - 2 F/A-18 Hornets, 45 flights
    - **Demonstrated up to 20% induced aerodynamic drag reduction**

- **AFF Autopilot**
  - **Baseline** linear classical design to meet stability margins
  - **Adaptive** incremental system to counteract unknown vortex effects and environmental disturbances
  - **On-line extremum seeking** command generation
AFF: Lead Aircraft Wingtip Vortex Effects
Induced Drag Ratio & Rolling Moment Coefficient

**Drag Reduction** ($\phi = 0$)

$C_{D_{form}} / C_{D}$

**Induced Rolling Moment** ($\phi = 0$)

$\Delta C_l$

“sweet” spot

close to vortex induced roll reversal
AFF: Trailing Aircraft Dynamics in Formation

- Trailing Aircraft:

  Longitudinal Dynamics

  \[
  \begin{align*}
  \dot{V} &= V \\
  \dot{\gamma} &= -g \sin \gamma + \frac{\rho V^2}{2m} S \left( C_{\alpha_s} \delta_T - C_D (M, \alpha) \eta(y, \phi) \right)
  \end{align*}
  \]

  Lateral Dynamics

  \[
  \begin{align*}
  \dot{\phi} &= p \\
  \dot{\phi} &= -a_p p + b_\delta \delta_a + \xi(y, \phi)
  \end{align*}
  \]

- Trailing Aircraft Modeling Assumptions
  - SCAS yields 1st order roll dynamics & turn coordination
  - \( a_p, b_\delta, C_{T_\delta} \) are unknown positive constants
  - \( C_D (M, \alpha), \eta(y, \phi), \xi(y, \phi) \) are unknown bounded functions of known arguments and shapes

- Lead aircraft trimmed for level flight
AFF: Vortex Seeking Formation Flight Control

**Problem**: Using *throttle* and *aileron* inputs

- Track desired longitudinal displacement command $l_c$
- Generate on-line and track lateral separation command $y_c$ in order to:
  - Minimize unknown vortex induced drag coefficient $\eta(y, \phi)$ with respect to its 1st argument, (lateral separation)
  \[
  \dot{V} = -g \sin \gamma + \frac{\rho V^2}{2m} S\left(C_{T_{\delta r}} \delta_T - C_D(M, \alpha)\eta(y, \phi)\right)
  \]

**Remarks**:

- Aileron controls lateral separation
- Throttle controls longitudinal separation
  - depends on lateral separation through unknown function $\eta(y, \phi)$

**Solution**

- Using Direct Adaptive Model Reference Control
- Radial Basis Functions for approximation of uncertainties
- Extremum Seeking Command Generation
  \[
  \dot{y}_r = -\gamma \left. \frac{\partial \hat{\eta}(y, \phi)}{\partial y} \right|_{y=y^*_r}, \quad \gamma > 0
  \]
**AFF: Simulation Data**
Open Problems and Future Work
Task 1: Validation & Verification (V&V) of Adaptive Systems

• Significant industry effort going into development of adaptive / reconfigurable GN&C systems
• Methods to test and certify flight critical systems are not readily available
• There exists a necessity to develop V&V methods and certification tools that are similar to and extend the current process for conventional, non-adaptive GN&C systems
• Theoretically justified V&V technologies are needed to:
  – provide a standard process against which adaptive GN&C systems can be certified
  – offer certification guidelines during the early design cycle of such systems
**Task 1: V&V of Adaptive Systems**

Road Map to Solution (Issue Paper)

- **Preliminary V&V Design Concept**
- **Intelligent System Certification Working Group**
- **Certification Authorities (FAA/DoD/NASA)**
- **Certification Guidelines Requirements**
- **Certification Process**
- **Certification of V&V Tool and Process**
- **Verification**
  - Plan/Conduct Verification Events (Test Plan)
- **Design and Build**
  - Certifiable
    - Design
    - Hardware
    - Software
- **Definition of Intelligent System Certification Requirement**
- **Implementation Requirements (Cert. Plan)**

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

**Two Major Tasks**
- Stability Margins / Robustness Analysis
- S/W V&V Procedures
Task 1: V&V of Adaptive Systems
Subtask: Theoretical Stability / Robustness Analysis

- Establish adaptive control design guidelines
  - Define rates of adaptation
  - Calculate stability / robustness margins
  - Determine bounds on control parameters that correspond to stability / robustness margins
- Perform system validation using the derived margins
- Incorporate modifications that lead to improvement (if required) in the stability / robustness margins
- Validate closed-loop system tracking performance
Task 2: Integrated Vehicle Health Management (IVHM) and Composite Adaptation

• Aerodynamic parameters are of paramount importance to IVHM system functionality
• Examine different sources of on-line aerodynamic parameter estimation
  – Tracking errors
  – Prediction errors
• Composite Adaptive Flight Control = (Indirect + Direct) MRAC
Task 3: Persistency of Excitation in Flight Mechanics

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

\[
\begin{align*}
\dot{V} &= \frac{T \cos \alpha - D}{m} - g \sin (\theta - \alpha) \\
\dot{\alpha} &= q - \frac{T \sin \alpha + L}{mV} + g \cos (\theta - \alpha) \\
\dot{q} &= \frac{M}{I_y} \\
\dot{\theta} &= q
\end{align*}
\]

\[
\begin{align*}
T &= \bar{q} S C_T \approx \bar{q} S C_{T_{eq}} \delta_T \\
L &= \bar{q} S C_L \approx \bar{q} S C_{L} (\alpha, q) \\
D &= \bar{q} S C_D \approx \bar{q} S C_{D} (\alpha, q) \\
M &= \bar{q} S \bar{c} C_M \approx \bar{q} S \bar{c} \left( C_{M_{0}} (\alpha, q) + C_{M_{\delta_e}} (\alpha, \delta_e) \delta_e \right)
\end{align*}
\]

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

\[
\begin{align*}
\begin{bmatrix}
\dot{e}_q \\
\dot{\alpha} \\
\dot{q} \\
\dot{x}
\end{bmatrix}
&=egin{bmatrix}
0 & 0 & 1 \\
0 & \frac{Z_\alpha}{V} & 1 \\
0 & M_\alpha & M_q \\
\hat{A} & \hat{B} & \hat{C}
\end{bmatrix}
\begin{bmatrix}
e_q \\
\dot{\alpha} \\
\dot{q} \\
\dot{x}
\end{bmatrix}
+ \begin{bmatrix}
0 \\
\frac{Z_\delta}{V} \\
M_\delta
\end{bmatrix}
\begin{bmatrix}
\dot{e}_q \\
\dot{\alpha} \\
\dot{q} \\
\dot{u}
\end{bmatrix}
\implies \dot{x} = \hat{A}\dot{x} + \hat{B}\dot{u}
\end{align*}
\]
LQR PI Baseline Controller (continued)

- LQR design for the “wiggle” system
  - Optimal feedback solution: \( \tilde{u} = -\tilde{K}\tilde{x} \)
  - Using original states:
    \[
    \dot{\delta}_e^{bl} = -\begin{pmatrix} K_q' & K_\alpha & K_q \end{pmatrix} \begin{pmatrix} e_q \\ \dot{\alpha} \\ \dot{q} \end{pmatrix} = -K_q e_q - K_\alpha \dot{\alpha} - K_q \dot{q}
    \]
  - Integration yields LQR PI feedback:
    \[
    \delta_e^{bl} = -10 e_q' - 3.2433 \alpha - 10.7432 q
    \]

\[
\begin{bmatrix}
\hat{A} = \\
0 & 0 & 1 \\
0 & -1.0189 & 0.9051 \\
0 & 0.8223 & -1.0774
\end{bmatrix}, \quad \hat{B} = \\
0 & 0 \\
-0.0022 & 0 \\
-0.1756 & 0
\end{bmatrix}, \quad \hat{Q} = \\
100 & 0 & 0 \\
0 & 1 & 0 \\
0 & 0 & 100
\]

<table>
<thead>
<tr>
<th>Eigenvalue</th>
<th>Damping</th>
<th>Freq. (rad/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>-7.97e-001 + 3.45e-001i</td>
<td>9.18e-001</td>
<td>8.68e-001</td>
</tr>
<tr>
<td>-7.97e-001 - 3.45e-001i</td>
<td>9.18e-001</td>
<td>8.68e-001</td>
</tr>
<tr>
<td>-2.39e+000</td>
<td>1.00e+000</td>
<td>2.39e+000</td>
</tr>
</tbody>
</table>
Short-Period Dynamics with Uncertainties

- **System:**
  \[
  \begin{bmatrix}
  \dot{e}_q^T \\
  \dot{\alpha} \\
  \dot{q}
  \end{bmatrix} =
  \begin{bmatrix}
  0 & 0 & 1 \\
  0 & \frac{Z_\alpha}{V} & 1 \\
  0 & M_\alpha & M_q
  \end{bmatrix}
  \begin{bmatrix}
  e_q^T \\
  \alpha \\
  q
  \end{bmatrix} +
  \begin{bmatrix}
  0 \\
  \frac{Z_\delta}{V} \\
  M_\delta
  \end{bmatrix}
  \Lambda \left( \delta_e + K_0(\alpha, q) \right) +
  \begin{bmatrix}
  -1 \\
  0 \\
  0
  \end{bmatrix}
  q_{cmd}
  \]

- **Reference model:**
  \[
  \dot{x} = A x + B_1 \Lambda \left( \delta_e + K_0(\alpha, q) \right) + B_2 q_{cmd}
  \]
  - no uncertainties
  - (Plant + Baseline LQR PI)
  \[
  \dot{x}_{ref} = \left( A + B_1 K_x^T \right) x_{ref} + B_2 q_{cmd} = A_{ref} x_{ref} + B_{ref} q_{cmd}
  \]

- **Control Goal**
  - Model following pitch rate tracking:
  \[
  \| x - x_{ref} \| \to 0
  \]
Adaptive Augmentation Design

• Total elevator deflection:

\[
\delta_e = \delta_e^{bl} + \delta_e^{ad} = K_q e_q I + K_x \alpha + K_q q + \hat{k}_q e_q I + \hat{k}_\alpha \alpha + \hat{k}_q q - \hat{\Theta}^T \Phi (\alpha, q)
\]

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

• Adaptive laws:

\[
\begin{align*}
\dot{k}_x &= \Gamma_x \text{Proj} \left( \dot{k}_x, -x e^T P B_1 \right) \\
\dot{\Theta} &= \Gamma_\Theta \text{Proj} \left( \dot{\Theta}, \Phi \left( x_p \right) e^T P B_1 \right) \\
\end{align*}
\]
Adaptive Augmentation Design (continued)

- **Free design parameters**
  - symmetric positive definite matrices: \((Q, \Gamma_x, \Gamma_\Theta)\)
- **Need to solve algebraic Lyapunov equation**
  \[ PA_{ref} + A_{ref}^T P = -Q \]
- **Using Dead-Zone modification and Projection Operator**
Adaptive Design Data

- **Design parameters**
  - using 11 RBF functions:
  - Rates of adaptation:
    \[ \Gamma_x = 0, \quad \Gamma_{\theta} = 1 \]
  - Solving Lyapunov equation with:
    \[ Q = \text{diag}([0 \quad 1 \quad 800]) \]

- **Zero initial conditions**

- **Pitch rate command input**

- **System Uncertainties**
  - 50% elevator effectiveness failure, \( 0.5 * M^{bl}_\delta \)
  - 50% increase in static instability, \( 1.5 * M^{bl}_\alpha \)
  - 80% decrease in pitch damping, \( 0.2 * M^{bl}_q \)
  - nonlinear pitching moment

\[
M(\alpha) = 1.5 * M^{bl}_\alpha + e^{\frac{(\alpha - 2\pi \frac{180}{180})^2}{0.0116^2}}
\]
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**
  \[
  \dot{\Theta}_i = (\Gamma_{\Theta})_{ii} \Phi_i(\alpha, q)(k_{1i}(q - q_{Iref}) + k_{2i}(\alpha - \alpha_{ref}) + k_{3i}(q - q_{ref}))
  \]

- **Fixed RBF NN gains**
  - simulation data
    \[
    k_{1i} = 0, \quad k_{2i} = -1.1266, \quad k_{3i} = -24.0516
    \]

- **Projection Operator**
  - keeps parameters bounded
  - nonlinear extension of anti-windup integrator logic

- **Dead-Zone modification**
  - freezes adaptation process if: \[\|x - x_{ref}\| \leq \mathcal{E}\]
  - separates adaptive augmentation from baseline controller