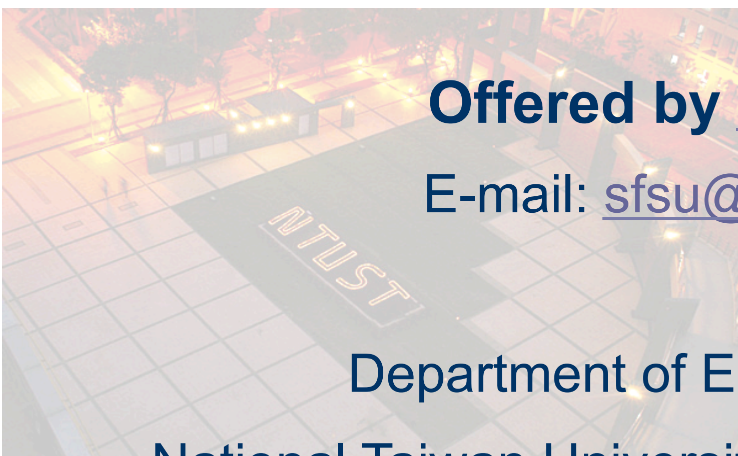
Learning Control

Ideas and Problems in Adaptive Fuzzy Control



Offered by Shun-Feng Su,

E-mail: sfsu@mail.ntust.edu.tw

Department of Electrical Engineering,

National Taiwan University of Science and Technology

July 26, 2016

Preface

- Intelligent control is a promising way of control design in recent decades.

Intelligent means to use knowledge in the process.

- Thus, Intelligent control design needs some knowledge of the system considered.
- However, such knowledge usually may not be available.
- Learning becomes an important mechanism for acquiring such knowledge.

Preface

- Learning control seems a good idea for control design for unknown or uncertain systems.
- To learn controllers is always a good idea, but somehow like a dream. It is because learning is to learn from something. But when there is no good controller, where to learn from?
- This talk is to discuss fundamental ideas and problems in one learning controller -- **adaptive fuzzy control**.

Outline

- Preface
- What is learning and what is learning control
- Introduction to fuzzy and its applications
- Basic ideas in adaptive fuzzy control
- Problems and possible approaches for resolving those problems
- Conclusive remarks

What is Learning?

In the literature, there are two important definitions for learning:

H. Simon defined learning as – "***any change in a system*** *that allows it to perform better the second time on the repetition of the same task or on another task drawn from the same population.*"

B. Kosko defined learning as **change** in all cases. "*A system learns if and only if the system parameter vector or matrix has a nonzero time derivative.*"

Concept of Machine Learning

- The first definition is to ask the system with learning should always **behave better** as learning continues.
- The second definition is mainly for **numerical learning**.
- Both definitions give a fundamental idea for learning – to change the system to make output differences.
- The fundamental problem for learning is ***how to change*** **the system to make the system's behaviors as required**.

Learning algorithm

Learning Control

Traditional learning control is to estimate (or successively approximate) some unknown quantities.

Categories of targets for learning in control:

- Learning about the **plant**;
- Learning about the **environment**;
- Learning about the **controller**; and
- Learning new **design goal and constraints**.

Learning Control

- The first two categories are more like **modeling**.

→It is easy to achieve by using supervised learning schemes, but have difficulties in designing controller (model based control) due to nonlinearity and learning insufficiency.

→Most learning control research efforts are in these two categories.

- The last one is AI related issue and is usually considered for general systems instead of control systems.

Learning Control

- To **learn controllers** is always a good idea, but somehow like a dream. It is because learning is to learn from something. But when there is no good controller, Where to learn from?
- Nevertheless, there still exist approaches, such adaptive fuzzy control, that can facilitate such an idea. ← **performance based learning**

To define a performance index and then change the system (learning) so as to optimize this index.

Learning Control

Performance based approach — find a way of optimizing the performance index.

- Reinforcement learning is to find parameters in the controller in a trial-and-error manner to optimize the performance index (external reinforcement).
- Lyapunov stability is to derive update rules of parameters so that the derivative of the considered Lyapunov function is negative.

- Preface
- What is learning and what is learning control
- **Introduction to fuzzy and its applications**
- Basic ideas in adaptive fuzzy control
- Problems and possible approaches for resolving those problems
- Conclusive remarks

Introduction of Fuzzy

Fuzzy have been widely used in various applications.

In fact, the fundamental idea behind fuzzy systems is to include uncertainty in the process. Such an inclusion provides extra information so that the systems can be more accurate.

In other words, fuzzy is vagueness by meaning, but can provides accurate due to this extra information.

Fuzzy

Fuzzy Logic Control

A Fuzzy Logic Controller (FLC) is a controller described by a collection of fuzzy rules (e.g. **IF-THEN** rules) involving linguistic variables.

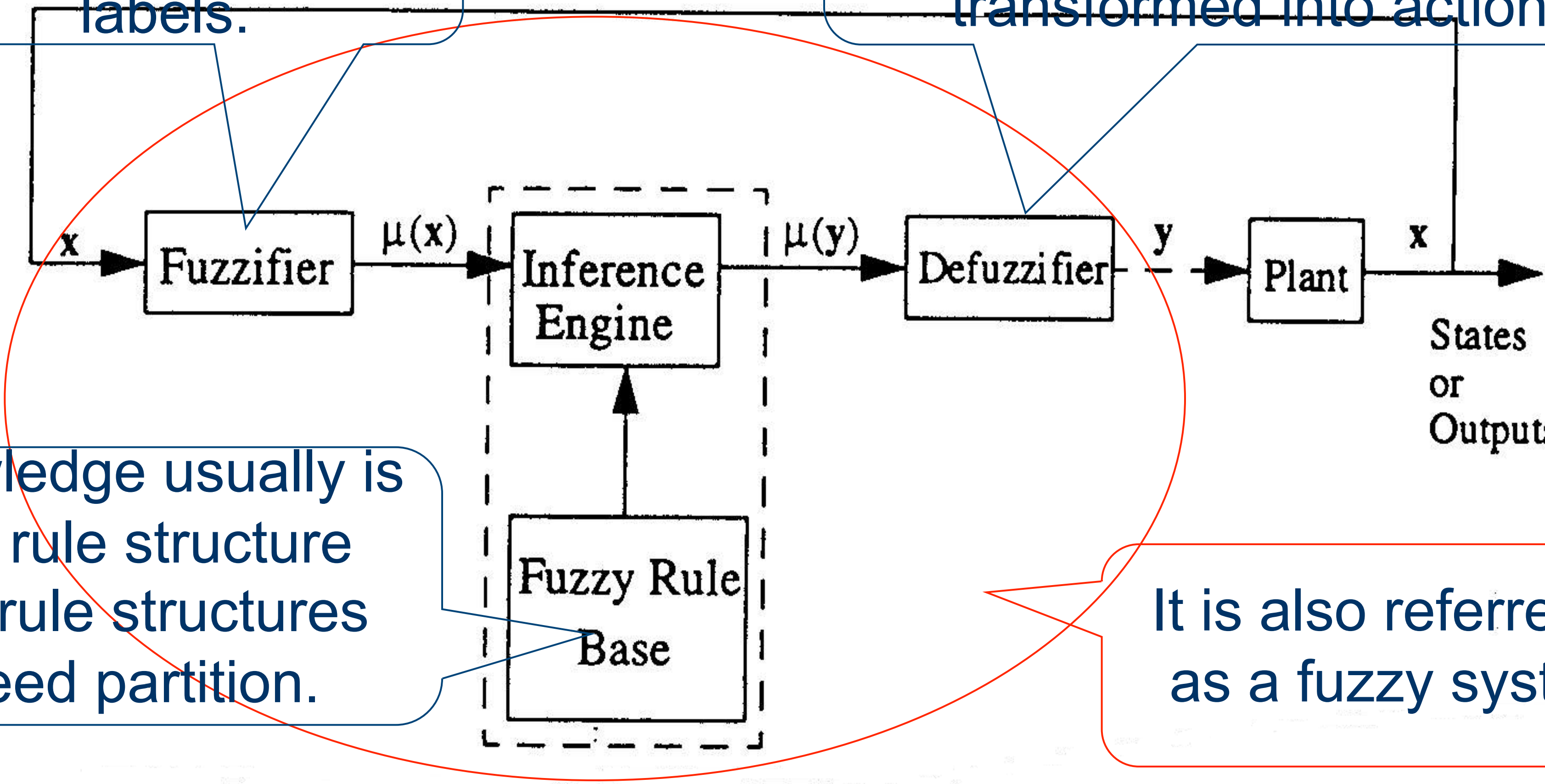
The original idea of using FLC is to incorporate the "expert experience" of a human operator into the design of the controller in controlling a process.

The utilization of linguistic variables, fuzzy control rules and approximate reasoning provides a means to incorporate human expert experience in designing the controller.

Fuzzy Logic Control

use rules, a value
ust be defined into
labels.

The consequences of all
matched rules must be
transformed into actions.



owledge usually is
a rule structure
d rule structures
need partition.

It is also referred to
as a fuzzy system.

Rationale behind FLC

In an FLC, the rule structure provides the adaptation among control strategies, and then the fuzzy mechanism provides the interpreting capability among rules.

With the interpreting capability, the transition between rules is gradual rather than abrupt. It is the so-called softening process.

Fuzzy Systems

In recent development, fuzzy systems have been considered as an alternative of a nonlinear system but with a linear system in each rule so that approaches for linear systems can also be applied.

Besides, various parameters are needed in fuzzy systems. Those parameters can be tuned to have excellent performance (by users or by learning mechanisms). ← adaptive fuzzy control

Fuzzy Systems

Mamdani *fuzzy rules* :

If (***X*** *is* ***A***) and (***Y*** *is* ***B***) … then (***Z*** *is* ***C***)

TSK (in modeling) **or** ***TS*** (in control) *fuzzy rules* :

If (***X*** *is* ***A***) and (***Y*** *is* ***B***) … then ***Z***=*f*(***X***, ***Y***).

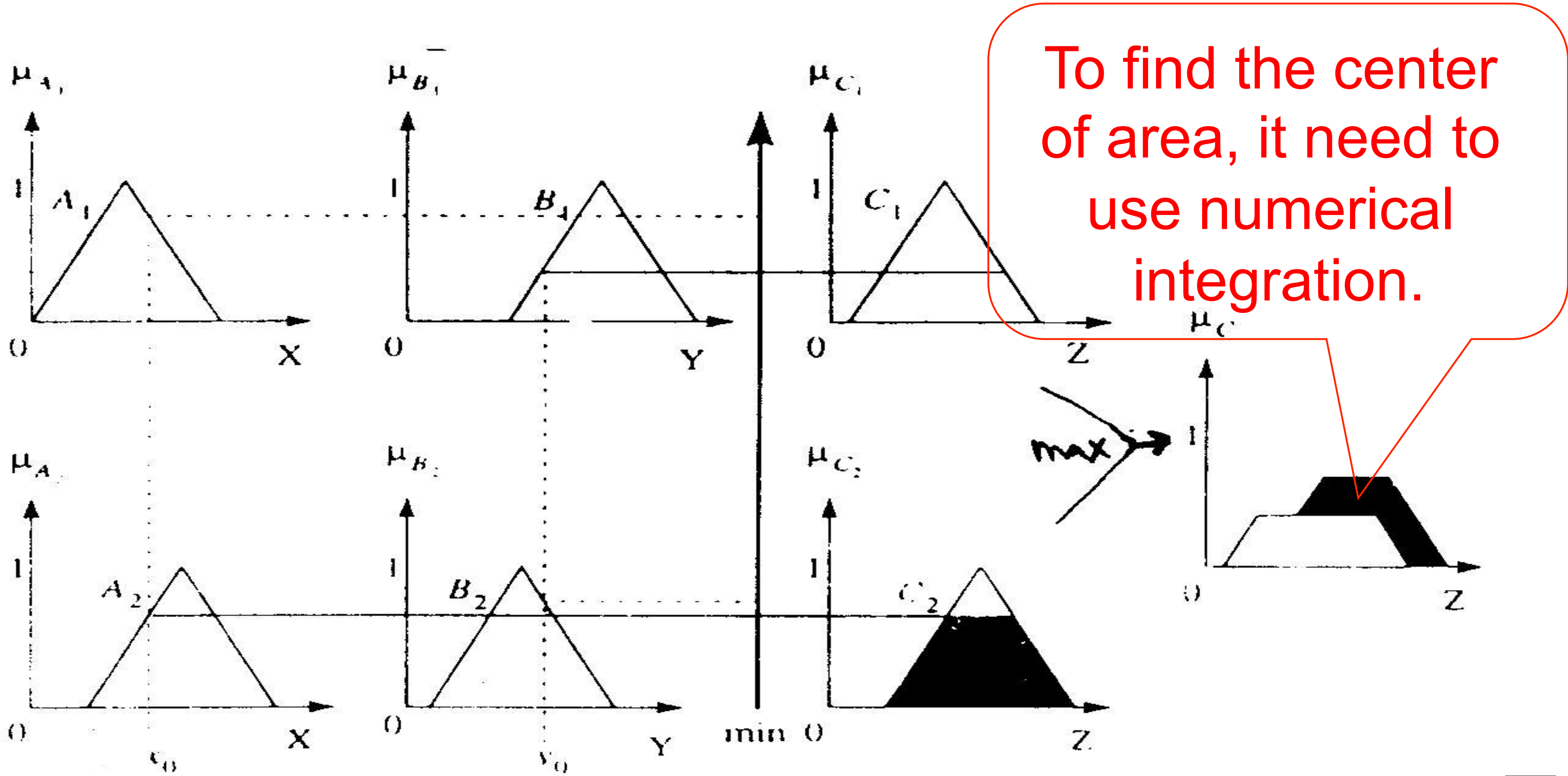
Note that ***C*** is a fuzzy set and *f*() is a crisp function.

$$\mu_{A\circ R}(v) = \max_u \min(\mu_A(u), t(\mu_B(u), \mu_C(v))).$$

Obtained from extension principle.

Fuzzy Systems

Mamdani fuzzy rules : COA defuzzification



Fuzzy Systems

TS fuzzy rules : Somewhat is also called COA. But without numerical integration. It is obtained as

$$z=\frac{\sum_{i=1}^{m}\alpha_i f_i}{\sum_{i=1}^{m}\alpha_i},$$

Simple and easy to calculate. Most importantly, it can be used in any mathematical operations, such as derivative.

where α_i and f_i are the firing strength and the fired result for the *i*-th rule and *m* is the rule number.

Fuzzy System

A fuzzy approximator is constructed by a set of fuzzy rules as

$$R^l : \text{IF } x_1 \text{ is } A_1^l, \text{ and } \cdots, \text{ and } x_n \text{ is } A_n^l \text{ THEN } y_F \text{ is } \theta^l,$$

$$\text{for } \; l = 1, 2, \cdots, M$$

Generally, θ^l is a fuzzy singleton (TS fuzzy model).

Another type is to use a linear combination of input variables. In that case, usually, the recursive least square (RLS) approach (or recursive Kalman filter) can be used to identify those coefficients.

Fuzzy System

The fuzzy systems with the center-of area like defuzzification and product inference can be obtained as

$$y_f(\mathbf{x}) = \frac{\sum_{l=1}^{M} \theta^l \left(\prod_{i=1}^{n} \mu_{A_i^l}(x_i)\right)}{\sum_{l=1}^{M} \left(\prod_{i=1}^{n} \mu_{A_i^l}(x_i)\right)}$$

t-norm operation for all premise parts

It is a universal function approximator and is written as $y_f(\mathbf{x}|\boldsymbol{\theta}) = \boldsymbol{\theta}^T \boldsymbol{\omega}$.

Fuzzy System

It should be noted that the above system is a nonlinear system. But, it can be seen that the form is virtually linear.($y_f(\mathbf{x}|\boldsymbol{\theta}) = \boldsymbol{\theta}^T \boldsymbol{\omega}$)

Thus, various approaches have been proposed to handle nonlinear systems by using the linear system techniques for the linear property bearing in each rule, such as common P stability or LMI design process.

- Preface
- What is learning and what is learning control
- Introduction to fuzzy and its applications
- **Basic ideas in adaptive fuzzy control**
- Problems and possible approaches for resolving those problems
- Conclusive remarks

Adaptive Fuzzy Control

Consider the following n*th* order nonlinear system :

$$N : \begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = x_3 \\ \vdots \\ x^{(n)} = f(\mathbf{x}) + g(\mathbf{x})u = f + gu \end{cases}$$

$y = x_1$ is the system output.

Adaptive Fuzzy Control

The objective is to design an controller such that y tracks a desired signal $y_m(t)$.

Let $e = y_m - y$ be the tracking error and can be written as $\mathbf{e} = [e, \dot{e}, \cdots, e^{(n-1)}]^T = [e_1, e_2, \cdots, e_n]^T$

Based on the **feedback linearization** method, if and are known, the reference controller is [1]

$$u^* = \frac{1}{g}(-f + y_m^{(n)} + \mathbf{k}^T \mathbf{e})$$

This is also referred to as the perfect control law

Feedback linearization

For tracking control, we need to obtain an perfect control law referred to "*Applied Nonlinear Control*" a method which is called "*Feedback Linearization Method.*"

$$u = \frac{1}{g(\mathbf{x})}[-f(\mathbf{x}) + x_d^{(n)} + \mathbf{K}^T\mathbf{E}]$$

$$\lim_{t \to \infty} e(t) \to 0$$

Hurwitz

substitute

$$\dot{x}_n = f(\mathbf{x}) + g(\mathbf{x})u$$

$$e^{(n)} + k_1 e^{(n-1)} + \cdots + k_n e = 0$$

Assume g(x)≠0

Adaptive Fuzzy Control

where $\mathbf{k} = [k_n, k_{n-1}, \cdots, k_1]^T$ is selected such that all roots of $s^{(n)} + k_{n-1}s^{(n-1)} + \cdots + k_1 s + k_0 = 0$ are in the open left-half plane.

The tracking error dynamics $x^{(n)} - y_m^{(n)} - \mathbf{k}^T \mathbf{e} = 0$ can have $\lim_{t \to \infty} \mathbf{e} = 0$.

If f and g are known, the control law can be fulfilled and then the control performance can be guaranteed.

Adaptive Fuzzy Control

Usually, f and g are unknown or subject to some uncertainty, thus the perfect control may not work.

Adaptive fuzzy control is then use fuzzy approximator (systems) to approximate them.

Direct adaptive fuzzy control – to estimate directly the controller u^*.

the perfect control law

$$u^* = \frac{1}{g}(-f + y_m^{(n)} + \mathbf{k}^T \mathbf{e})$$

To use $y_f(\mathbf{x}|\boldsymbol{\theta}) = \boldsymbol{\theta}^T \boldsymbol{\omega}$ to model

Adaptive Fuzzy Control

- Usually, f and g are unknown or subject to some uncertainty, thus the perfect control may not work.
- Adaptive fuzzy control is then use fuzzy approximator (systems) to approximate them.
- ***Direct adaptive fuzzy control*** – to estimate directly the controller u^*.
- ***Indirect adaptive fuzzy control*** – to estimate f and g.

the perfect control law $u^* = \frac{1}{g}(-f + y_m^{(n)} + \mathbf{k}^T\mathbf{e})$

Direct Adaptive Fuzzy Control

To approximate the controller by using a fuzzy system as $\hat{u}(\mathbf{x}|\boldsymbol{\theta}) = \boldsymbol{\theta}_D{}^T \boldsymbol{\omega}$ [3].

Consider the following Lyapunov function

$$V = \frac{1}{2}\mathbf{e}^T \mathbf{P}\mathbf{e} + \frac{1}{2\alpha}\tilde{\boldsymbol{\theta}}_D^T \tilde{\boldsymbol{\theta}}_D$$

where $\tilde{\boldsymbol{\theta}}_D = (\boldsymbol{\theta}_D^* - \boldsymbol{\theta}_D)$ is the error of the estimated parameter and $\boldsymbol{\theta}_D^*$ is the optimal parameter vector and is defined as

$$\boldsymbol{\theta}_D^* \equiv \arg\min_{\boldsymbol{\theta}_D \in \Omega_{\theta_d}} \{ \sup_{\mathbf{x} \in \Omega_x} \left| u^* - \hat{u}_D^*(\mathbf{x}|\boldsymbol{\theta}_D) \right| \}$$

Direct Adaptive Fuzzy Control

- The idea is to let the derivative of the Lyapunov function is negative. In that case, the system can be said to be stable and the error will eventually become zero if possible.
- The second term of the Lyapunov function can be view as to minimize the approximation errors.
- In fact, it is to generate the derivative of $\boldsymbol{\theta}_D$, which will be used to form the update rule for $\boldsymbol{\theta}_D$:

$$\ldots + \frac{d}{dt}(\boldsymbol{\theta}_D^T \boldsymbol{\theta}_D) = \ldots + \boldsymbol{\theta}_D^T \dot{\boldsymbol{\theta}}_D + \ldots = \ldots + \boldsymbol{\theta}_D^T(\dot{\boldsymbol{\theta}}_D + \text{terms}) + \ldots$$

$$\dot{\boldsymbol{\theta}}_D = \text{terms} \quad \Leftarrow \quad \text{update law}$$

Direct Adaptive Fuzzy Control

The idea is to let the derivative of the Lyapunov function is negative. In that case, the system can be said to be stable and the error will eventually become zero if possible.

The second term of the Lyapunov function can be view as to minimize the approximation errors.

In fact, it is to generate the derivative of θ_D, which will be used to form the update rule for θ_D.

This kind of approach can be seen in lots of learning or adaptive control schemes.

Direct Adaptive Fuzzy Control

$$\dot{V} = \frac{1}{2}\dot{\mathbf{e}}^T\mathbf{P}\mathbf{e} + \frac{1}{2}\mathbf{e}^T\mathbf{P}\dot{\mathbf{e}} - \frac{1}{\alpha}\tilde{\boldsymbol{\theta}}_D^T\dot{\boldsymbol{\theta}}_D$$

$$= -\frac{1}{2}\mathbf{e}^T\mathbf{Q}\mathbf{e} + \mathbf{e}^T\mathbf{P}\mathbf{B}\varepsilon_D + \frac{1}{\alpha}\tilde{\boldsymbol{\theta}}_D^T(\alpha\mathbf{e}^T\mathbf{P}\mathbf{B}\boldsymbol{\omega}_D - \dot{\boldsymbol{\theta}}_D)$$

$\Lambda^T\mathbf{P} + \mathbf{P}\Lambda = -\mathbf{Q}$

Traditional Lyapunov derivation

$\varepsilon_D = u^* - \hat{u}_D^*$ is approximate error (assumed to be small enough.)

The idea is to let $\dot{\boldsymbol{\theta}}_D = \alpha\mathbf{e}^T\mathbf{P}\mathbf{B}_1\boldsymbol{\omega}_D$ and to prove the remaining terms are negative in general.

Then, it can be claimed that $\dot{V}$ is negative.

$\Delta\boldsymbol{\theta}_D = \alpha\mathbf{e}^T\mathbf{P}\mathbf{B}_1\boldsymbol{\omega}_D\Delta t$, it is the update law for $\boldsymbol{\theta}$

Direct Adaptive Fuzzy Control

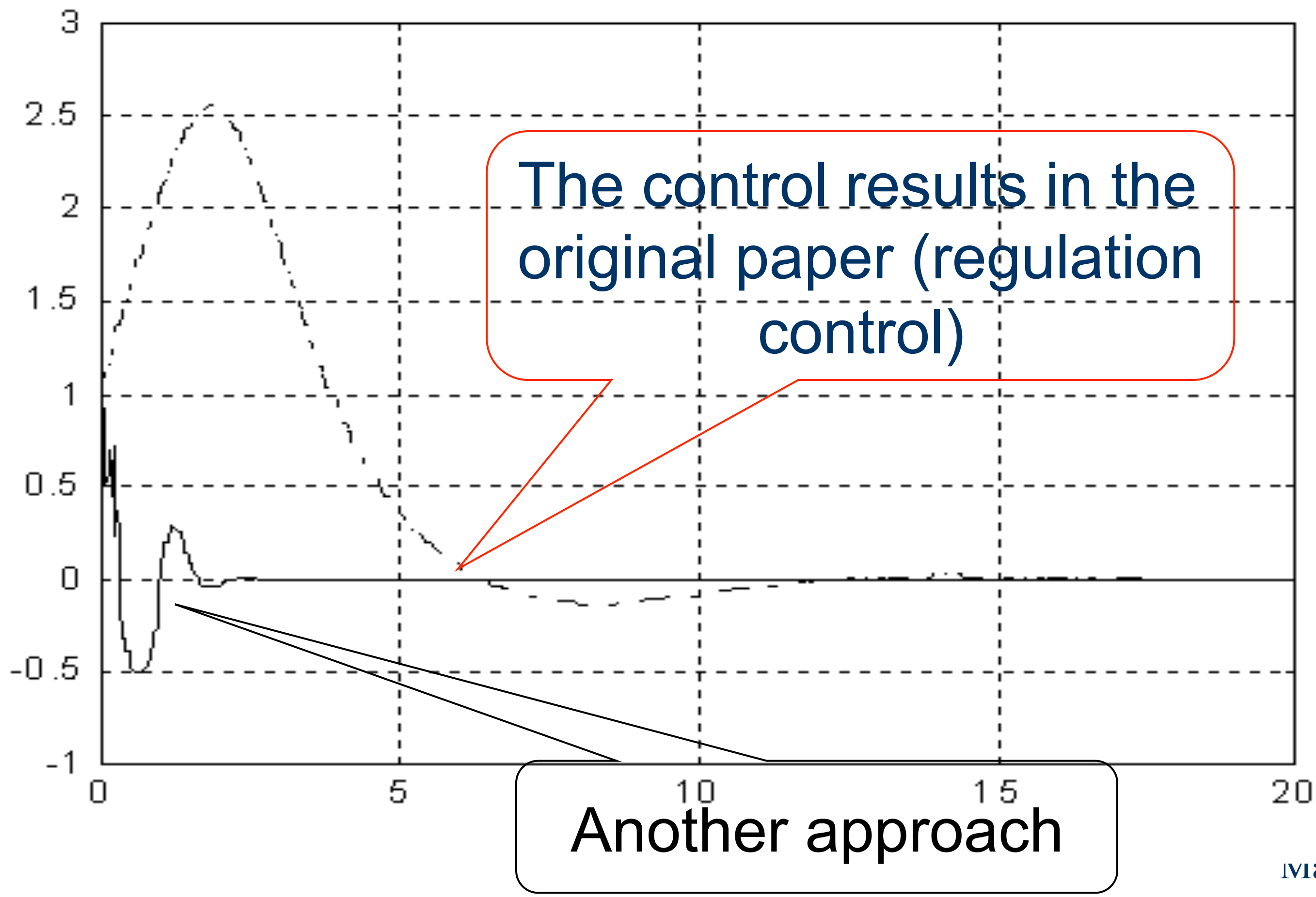
If you actually use this approach, the results may not be satisfactory.

The example shown in the paper is only for regulation control.

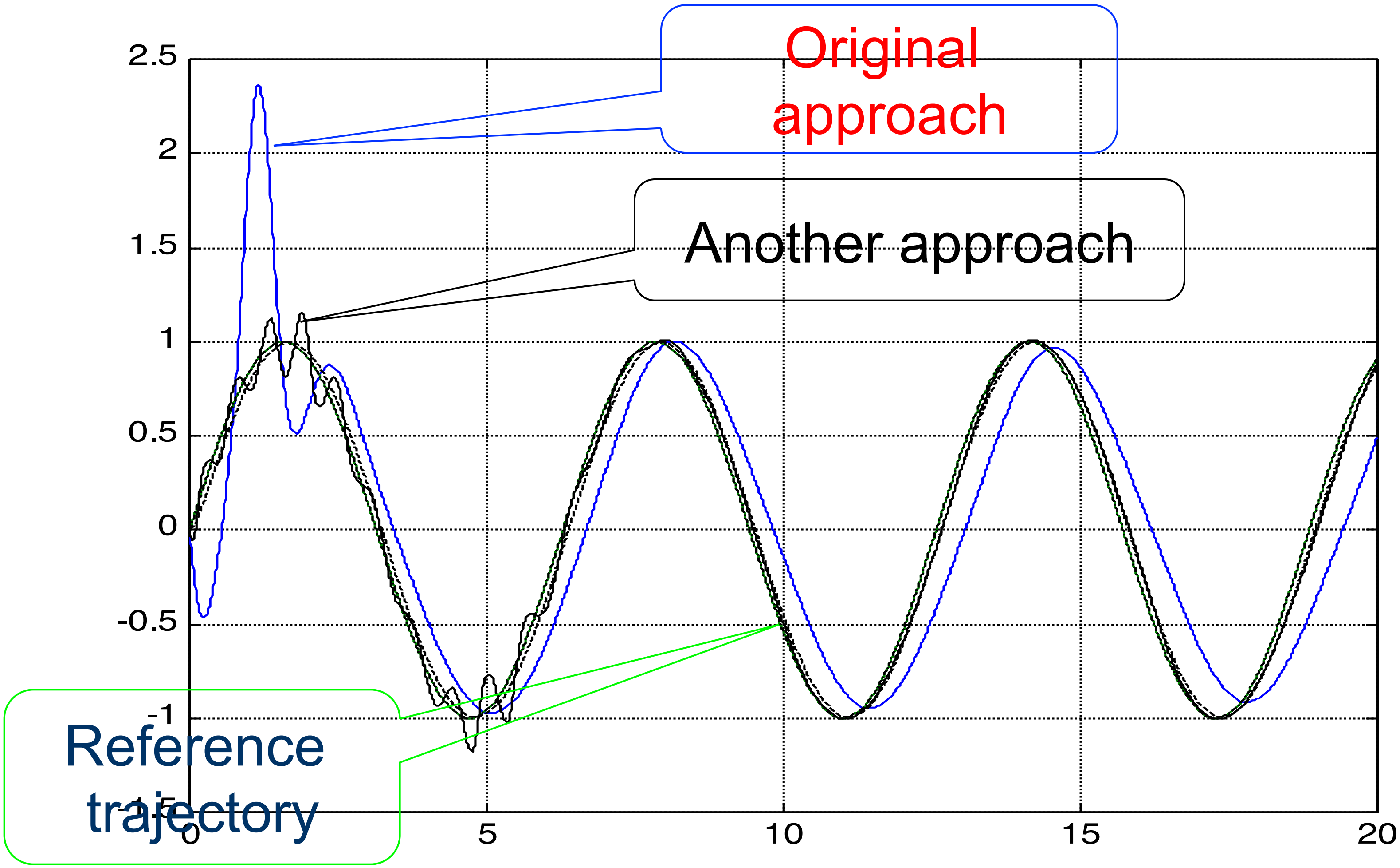
Less further work has been reported in the literature.

The main problem is whether there exist the optimal control and whether it can be approximated by the fuzzy approximator; that is, whether $\varepsilon_D = u^* - \hat{u}_D^*$ is small enough?

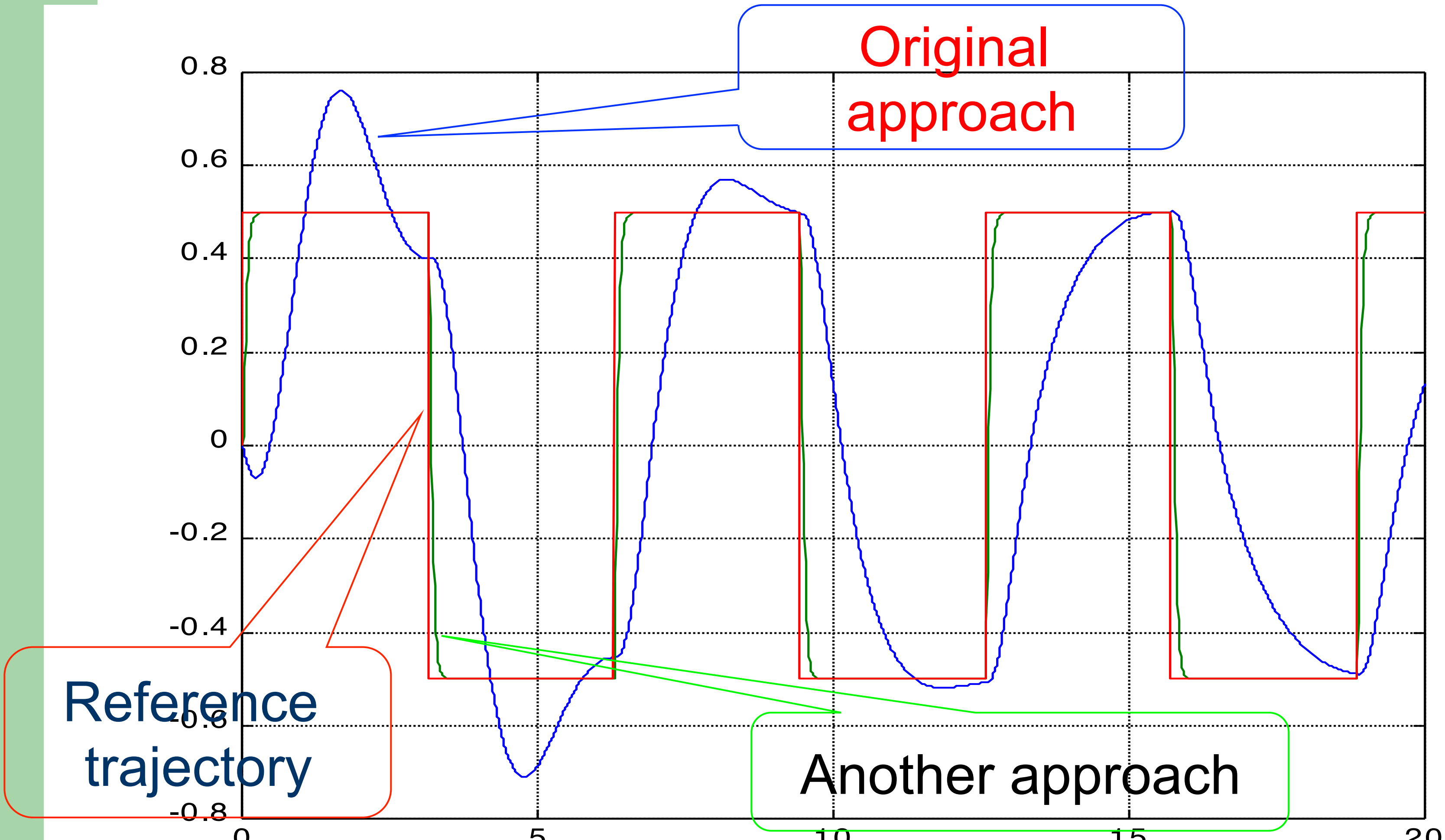
Direct Adaptive Fuzzy Control (regulation control)



Direct Adaptive Fuzzy Control (Tracking Control)



Direct Adaptive Fuzzy Control [6]



Indirect Adaptive Fuzzy Control [4]

To approximate f and g by using two fuzzy systems as $\hat{f}(\mathbf{x}|\boldsymbol{\theta}_f) = \boldsymbol{\theta}_f^T \boldsymbol{\omega}_f = \hat{f}$ and $\hat{g}(\mathbf{x}|\boldsymbol{\theta}_g) = \boldsymbol{\theta}_g^T \boldsymbol{\omega}_g = \hat{g}$.

Consider the following Lyapunov function

$$V = \frac{1}{2}\mathbf{e}^T \mathbf{P}\mathbf{e} + \frac{1}{2\beta_1}\tilde{\boldsymbol{\theta}}_f^T \tilde{\boldsymbol{\theta}}_f + \frac{1}{2\beta_2}\tilde{\boldsymbol{\theta}}_g^T \tilde{\boldsymbol{\theta}}_g$$

where those variables are similar to those defined in direct adaptive control.

Indirect Adaptive Fuzzy Control

$$\dot{V} = -\frac{1}{2}\mathbf{e}^T\mathbf{Q}\mathbf{e} + \mathbf{e}^T\mathbf{P}\mathbf{B}_1\varepsilon_I$$

$$+\frac{1}{\beta_1}\tilde{\boldsymbol{\theta}}_f^T(\beta_1\mathbf{e}^T\mathbf{P}\mathbf{B}_1\boldsymbol{\omega}_f + \dot{\boldsymbol{\theta}}_f) + \frac{1}{\beta_2}\tilde{\boldsymbol{\theta}}_g^T(\beta_2\hat{u}_I\mathbf{e}^T\mathbf{P}\mathbf{B}_1\boldsymbol{\omega}_g + \dot{\boldsymbol{\theta}}_g)$$

The approximate error is $\varepsilon_I = (\hat{f}^* - f) + (\hat{g}^* - g)\hat{u}_I$

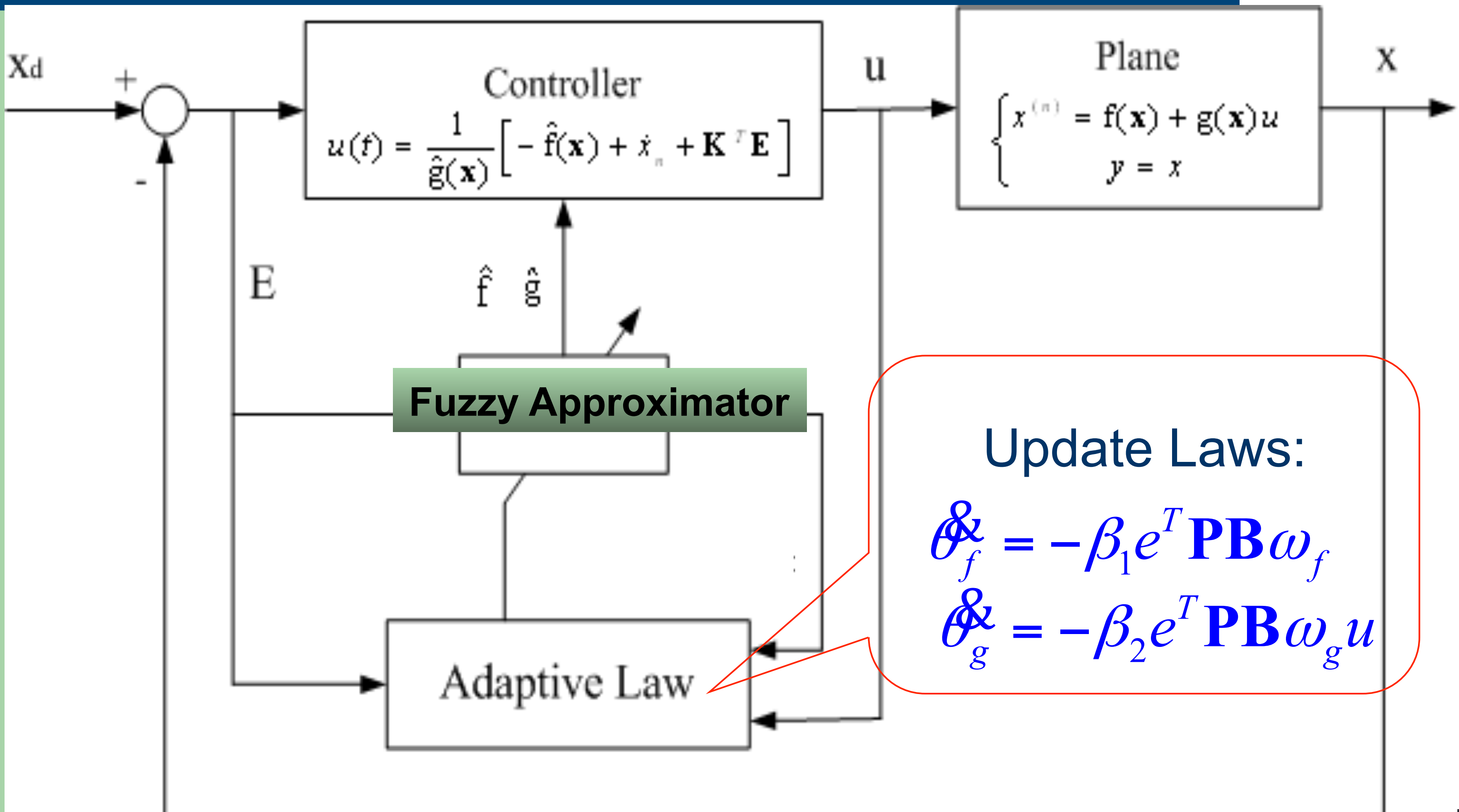
Similarly we can we the update rules as

$$\dot{\boldsymbol{\theta}}_f = -\beta_1\mathbf{e}^T\mathbf{P}\mathbf{B}_1\boldsymbol{\omega}_f$$

$$\dot{\boldsymbol{\theta}}_g = -\beta_2\hat{u}_I\mathbf{e}^T\mathbf{P}\mathbf{B}_1\boldsymbol{\omega}_g$$

Assume to be small enough

Original adaptive control scheme



Update Laws:

$$\dot{\theta}_f = -\beta_1 e^T \mathbf{PB}\omega_f$$

$$\dot{\theta}_g = -\beta_2 e^T \mathbf{PB}\omega_g u$$

Adaptive Fuzzy Control

Adaptive fuzzy control is to use fuzzy approximator

$$y_f(\mathbf{x}|\boldsymbol{\theta}) = \boldsymbol{\theta}^T \boldsymbol{\omega}$$.

Fuzzy systems are universal approximators [2].

Other universal approximators can also be used, such as:

- Radial Basis Functions;
- Cerebellar Model Articulation Controllers;
- Wavelets; etc.

As long as they can also be written as a linear form like $y_f(\mathbf{x}|\boldsymbol{\theta}) = \boldsymbol{\theta}^T \boldsymbol{\omega}$.

Adaptive Fuzzy Control

- Some approaches claimed that they also used neural networks to act as the approximator in their approach. In fact, it is one kind of radial basis function neural networks, which can be equivalent to a fuzzy system.
- If you want to use other approximators which are not of linear form, some linear approximation approaches (like first order of Taylor expansion) may be employed to make it workable in the framework.

Adaptive Fuzzy Control

Also, such an idea can be employed to find adaptive laws for other parameters.

Again, a linear form is needed (or some linear approximation mechanism is employed) to ensure a simple form of the update law.

Besides, the squared term of the parameter must be added into the Lyapunov function to have a basic formation of the update law.

Adaptive Fuzzy Control

Some approaches also adapt the idea of sliding control, by defining the sliding surface as the integral of the characteristic polynomial as

$S(t) = \int \dot{S}(t)dt$, where $\dot{S}(t) = e^{(n)} + \hat{k}^T \hat{e}$ [7].

Then, the idea is to replace all error terms by the sliding term. For example, the Lyapunov function is defined as:

$$V_2 = \frac{1}{2}\left(S^2 + \frac{1}{r_g}\tilde{\theta}_g^T \tilde{\theta}_g + \frac{1}{r_f}\tilde{\theta}_f^T \tilde{\theta}_f\right)$$

Similar results can be obtained.

- Preface
- What is learning and what is learning control
- Introduction to fuzzy and its applications
- Basic ideas in adaptive fuzzy control
- **Problems and possible approaches for resolving those problems**
- Conclusive remarks

Adaptive Fuzzy Control

There are problems in the above approaches:

- *Control aspect* — ε_D or ε_I (the approximate errors) may not be small. It may cause a *system* stability problem.
- *Learning aspect* — Large error (chattering phenomenon) in the Initial stage and convergence problem (parameter drifting) in the final stage.

Adaptive Fuzzy Control

There are problems in the above approaches:

- Approximate errors and robust control
- Initialization and supervisory control.
- Parameter drifting

Approximate Errors

Errors ε_D may exist due to rule resolution (rule numbers) and rule dependency (input-output deterministic).

Rule resolution may not be sufficient if the rule number used is small. ← universal approximator theorem

$$\dot{V} = -\frac{1}{2}\mathbf{e}^T\mathbf{Q}\mathbf{e} + \mathbf{e}^T\mathbf{P}\mathbf{B}\varepsilon_D + \frac{1}{\alpha}\tilde{\boldsymbol{\theta}}_D^T(\alpha\mathbf{e}^T\mathbf{P}\mathbf{B}\,\boldsymbol{\omega}_D - \dot{\tilde{\boldsymbol{\theta}}}_D)$$

When ε_D is large, then this part may not be negative.

With the update law, it is zero.

Approximate Errors

Another possible error-- Rule dependency may not be sufficient if the input variables used to define the input-output relationship is not sufficient. ← It is called nondeterministic in traditional learning.

For current published work, only the error and the error derivative are used as the input variables. If the system considered is more complicated, maybe more terms must be included.

Robust Control

Feedback linearization

$$u^* = g^{-1}(-f + y_m^{(n)} + \mathbf{k}^T \mathbf{e})$$

$$\dot{\mathbf{e}} = \mathbf{\Lambda e} + \mathbf{B}(u^* - u), \ \mathbf{\Lambda}^T \mathbf{P} + \mathbf{P\Lambda} = -\mathbf{Q}, \ u = u_{design} + u_{else}$$

Designed input → u_{design}

error → u_{else}

$$\psi : \begin{cases} \dot{\mathbf{e}} = \mathbf{\Lambda e} + \mathbf{B}(\varepsilon_{dI} - u_{design}) \\ y_e = \mathbf{e} \end{cases}$$

- Differential input (u_{design})
- An assignable stable inner linear system ($\mathbf{\Lambda e}$)
- System output (y_e)

$$\varepsilon_{dI} = u^* - u_{else}$$

- Unknown input (u_{else})

It can be viewed as an unknown input (u_{else})

It is ε_D or ε_I in the above. (ε_{dI})

Robust Control

$$\psi : \begin{cases} \dot{\mathbf{e}} = \mathbf{\Lambda e} + \mathbf{B}(\varepsilon_{dI} - u_{design}) \\ y_e = \mathbf{e} \\ \varepsilon_{dI} = u^* - u_{else} \end{cases}$$

Use a Lyapunov function to find the energy change of system Ψ.

$$V = \mathbf{e}^T \mathbf{P} \mathbf{e}$$

We have

$$\dot{V} = -\mathbf{e}^T \mathbf{Q} \mathbf{e} - 2\mathbf{e}^T \mathbf{PB}\, u_{design} + 2\mathbf{e}^T \mathbf{PB}\, \varepsilon_{dI}$$ (Energy dynamics equation)

How to design u_{design} to yield that the energy dynamics ($\dot{V}$) fits in with a **special form**.

Dissipative
H_∞ tracking performance
L_2-gain inequality

Robust Control

$$\dot{V} = -\mathbf{e}^T\mathbf{Q}\mathbf{e} - 2k_d|g|(\mathbf{e}^T\mathbf{P}\mathbf{B}_1)^2 + 2\mathbf{e}^T\mathbf{P}\mathbf{B}_1 g\,\varepsilon_u$$

$$\dot{V} = -\mathbf{e}^T\mathbf{Q}\mathbf{e} - \{(\sqrt{2k_d|g|})\mathbf{e}^T\mathbf{P}\mathbf{B}_1 - (\frac{\varepsilon_u g}{\sqrt{2k_d|g|}})\}^2 + (\frac{g}{\sqrt{2k_d|g|}})\varepsilon_u^T(\frac{g}{\sqrt{2k_d|g|}})\varepsilon_u$$

$$\dot{V} \le -\mathbf{e}^T\mathbf{Q}\mathbf{e} + (\frac{g_{up}}{\sqrt{2k_d g_{low}}})\varepsilon_u^T(\frac{g_{up}}{\sqrt{2k_d g_{low}}})\varepsilon_u$$

Supply rate

$$\omega(\varepsilon_u, \mathbf{e}) = -\mathbf{e}^T\mathbf{Q}\mathbf{e} + (\frac{g_{up}}{\sqrt{2k_d g_{low}}})\varepsilon_u^T(\frac{g_{up}}{\sqrt{2k_d g_{low}}})\varepsilon_u$$

Robust Control

$$\dot{V} \le -\mathbf{e}^T \mathbf{Q} \mathbf{e} + \left(\frac{g_{up}}{\sqrt{2k_d g_{low}}}\right)\varepsilon_u^T \left(\frac{g_{up}}{\sqrt{2k_d g_{low}}}\right)\varepsilon_u$$

Integral

- **H_∞ tracking performance**

$$\int_0^\infty \mathbf{e}^T \mathbf{Q} \mathbf{e}\, dt \le \mathbf{e}^T(0)\,\mathbf{P}\,\mathbf{e}(0) + \left(\frac{g_{up}}{\sqrt{2k_d g_{low}}}\right)^2 \int_0^\infty \varepsilon_u^T \varepsilon_u dt$$

- ***L_2*-gain-like inequality**

$$\frac{\|\mathbf{e}_Q\|_{L_2}^2}{\|\varepsilon_u\|_{L_2}^2} \le \left(\frac{g_{up}}{\sqrt{2k_d g_{low}}}\right)^2 = \delta^2\Big|_{V(\mathbf{e}(0))=0}$$

$$\|\varepsilon_u\|_{L_2}^2 = \int_0^\infty \varepsilon_u^T \varepsilon_u dt \ , \quad \|\mathbf{e}_Q\|_{L_2}^2 = \int_0^\infty \mathbf{e}^T \mathbf{Q}_d \mathbf{e}\, dt$$

***Controllable* attenuation level δ**

$$k_d = (g_{up}/\delta)^2 (2g_{low})^{-1}$$

Considered Example

Consider an often-used inverted pendulum system as

$$\dot{x}_1 = x_2$$

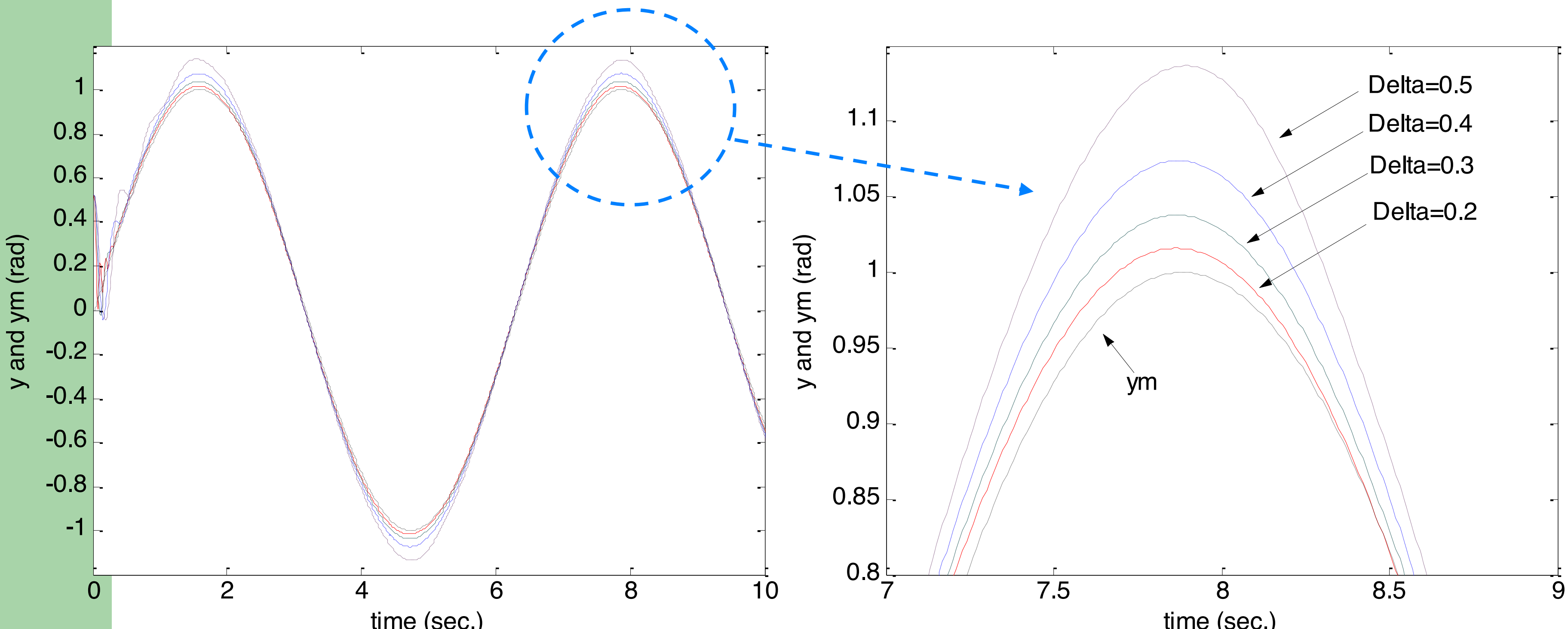
$$\dot{x}_2 = f + gu + d$$

$$f = \frac{g \sin x_1 - (mlx_2^2 \sin x_1 \cos x_1)/(m_c + m)}{l(4/3 - m\cos^2 x_1 / m_c + m)}$$

$$g = \frac{\cos x_1 /(m_c + m)}{l(4/3 - \cos^2 x_1 / m_c + m)}$$

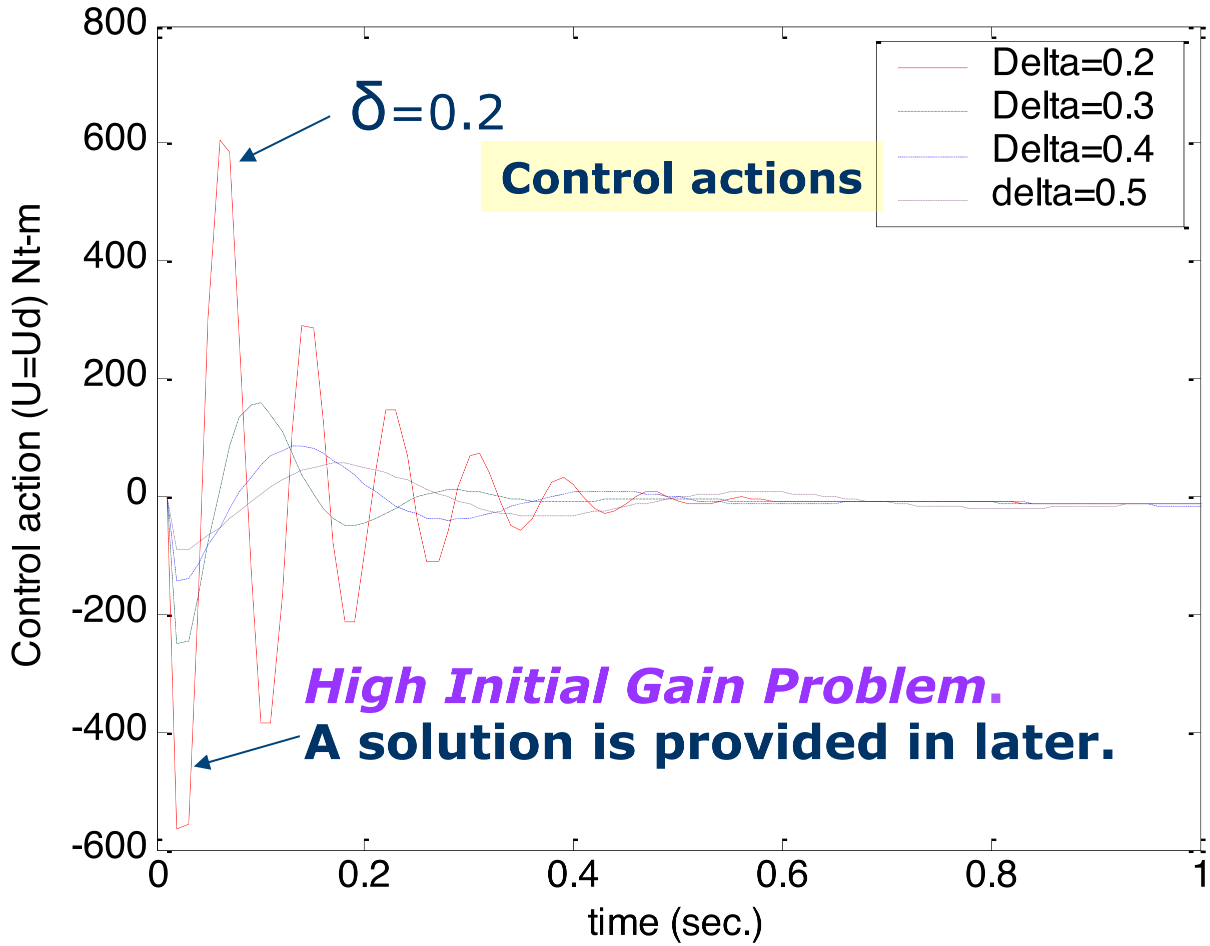
Robust Control

Simulation 1: The assignable control performance test. We let δ=0.2, 0.3, 0.4, and 0.5.



Tracking control performance

Robust Control



High Initial Gain Problem.

A solution is provided in later.

Robust Control

$u = K_c \operatorname{sgn}(g)\mathbf{e}^T\mathbf{PB}_1$

L_2-gain state feedback controller

$$\dot{V} = -\mathbf{e}^T\mathbf{Q}\mathbf{e} + \mathbf{e}^T\mathbf{PB}(u^* - u)$$

$$\dot{V} = -\mathbf{e}^T\mathbf{Q}\mathbf{e} - K_c|g|(\mathbf{e}^T\mathbf{PB}_1)^2 + \mathbf{e}^T\mathbf{PB}\,u^*$$

Negative define term

Bounded

A suitable value of K_c leads the equation to be minimum, which results in a more negative value of the derivation of V, *and the initial control action does not have the oscillation (high-gain) problem*.

How to find a suitable K_c ?

Robust Control

A large K_c will have nice control performance (small δ) but will have large initial control gain, but a small K_c may have a large error in the final stage.

A idea is to use a small K_c in the initial stage and a large K_c in the final stage. But how to change?

The research goal is that how to reduce the oscillation phenomenon of the initial control action and remain the satisfactory initial state response.

Use ***genetic algorithm*** to in adjusting K_c.

Search region

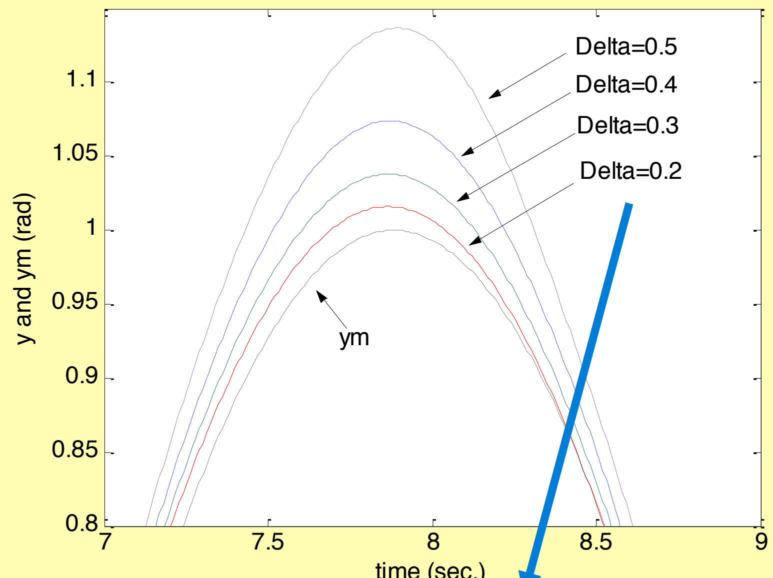
The **assignable** control performance is a inherent property of the L_2-gain control, which can be applied to define the **search region**.

The attenuation level δ determines the tracking control accuracy, and we can use the selection of K_c to adjust δ,

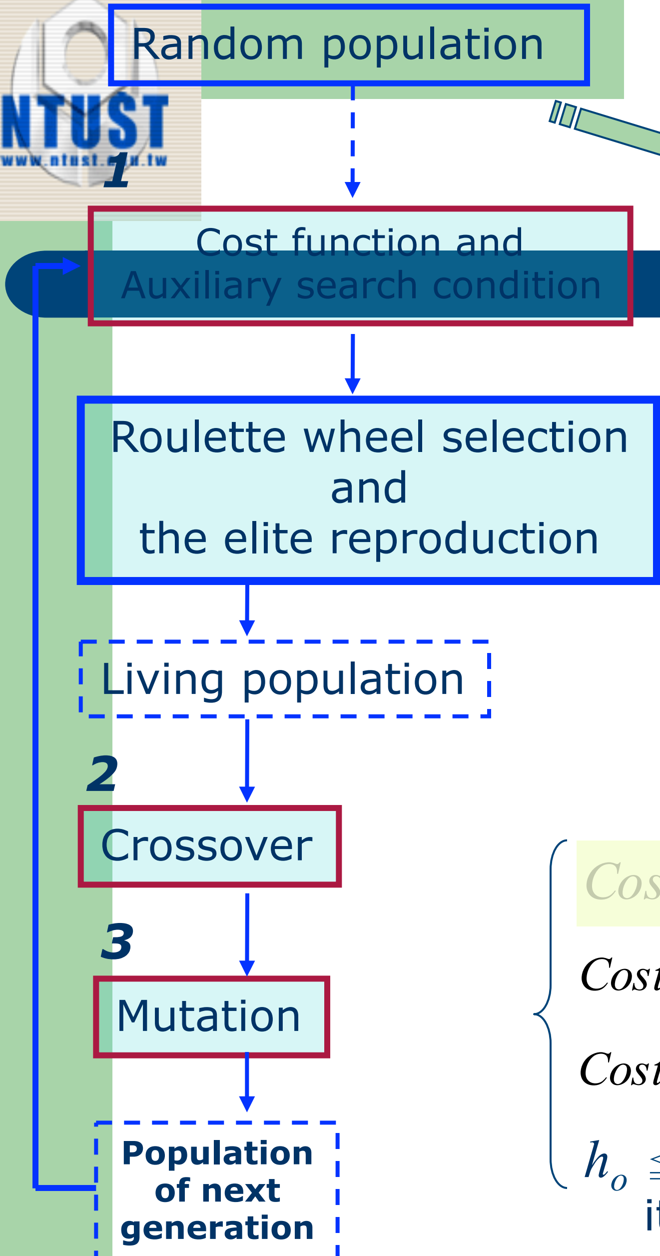
such as

$$\begin{cases} \delta = g_{up} / \sqrt{2K_c g_{low}} \\ g_{low} \le |g| \le g_{up} \end{cases}$$

Example:



$$\delta_{max} = 0.5, \ \delta_{min} = 0.2$$



The l-th chromosome is represented as

$$Ch^l = (K_c^l) \; , \quad l = 1,2,\cdots,m$$

m is the number of the used chromosomes.

K_c^l is a gene (solution) of the chromosome.

If a chromosome cannot satisfy the auxiliary search condition, or its cost is larger than a threshold cost ($CostT$) , then the chromosome will be replaced by another good chromosome.

$$CostT(k) = \{|CostMin(k)| - |CostAvg(k)|\}\, h_o + CostMin(k)$$

$CostMin(k)$ is the minimum cost of the k-th generation.

$CostAvg(k)$ is the average cost of the k-th generation.

$h_o \leq 1$ is a retaining constant;
it provides the multiplicity for the population.

Cost function

$$\dot{V} = -\mathbf{e}^T\mathbf{Q}\mathbf{e} - K_c|g|(\mathbf{e}^T\mathbf{P}\mathbf{B}_1)^2 + \mathbf{e}^T\mathbf{P}\mathbf{B}\,u^*$$

Negative definite term

Bounded

Cost function $CostE = -K_c(\mathbf{e}^T\mathbf{P}\mathbf{B}_1)^2$

A suitable value of K_c leads the cost function to be minimum.

In order to resolve the oscillation problem in the initial stage, we must avoid the minimum solution being found too early.

↓

The evolution speed is needed to be restricted that is the design basis of the auxiliary search condition.

↓

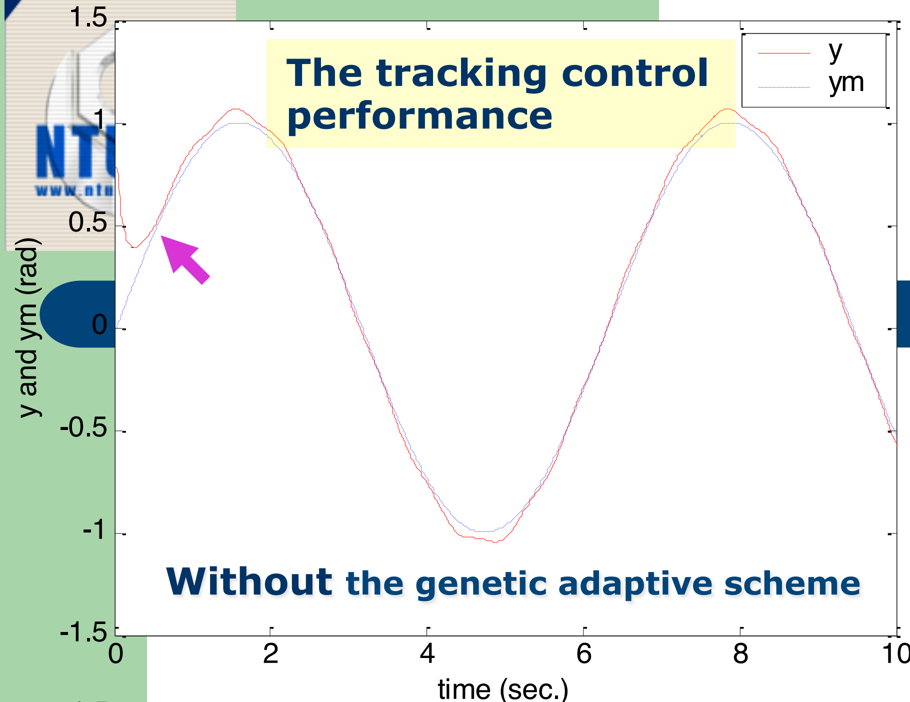
- **An auxiliary search condition** is defined under the change of the control action as

$$\left|\, |u(t)| - |u(t-\Delta t)| \,\right| < u_{\Delta}$$

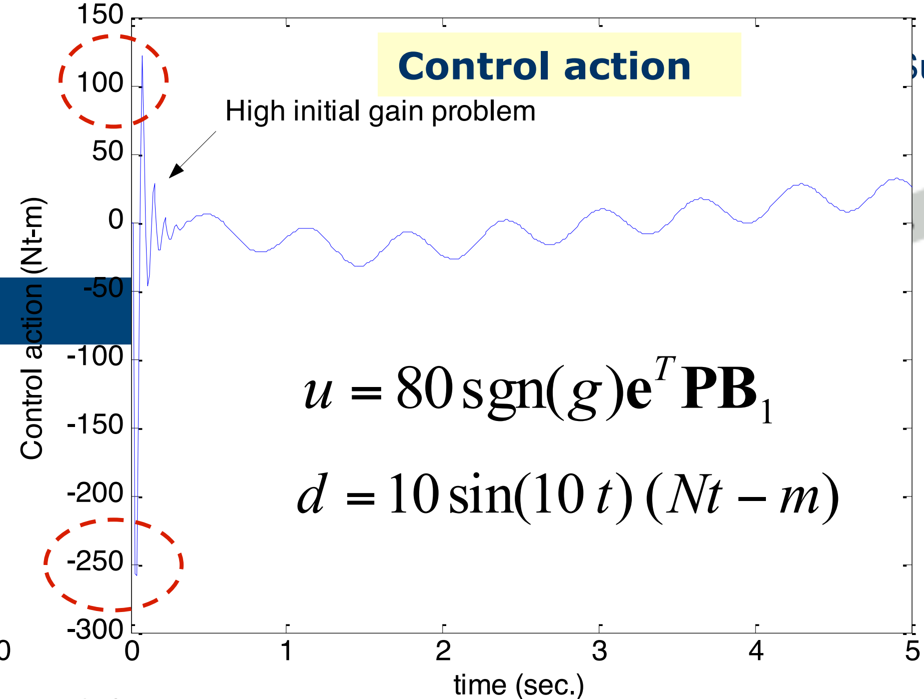
u_{Δ} → A constant which is used to restrict the evolution speed.

Δt → It is the sample time (0.01 seconds).

The tracking control performance

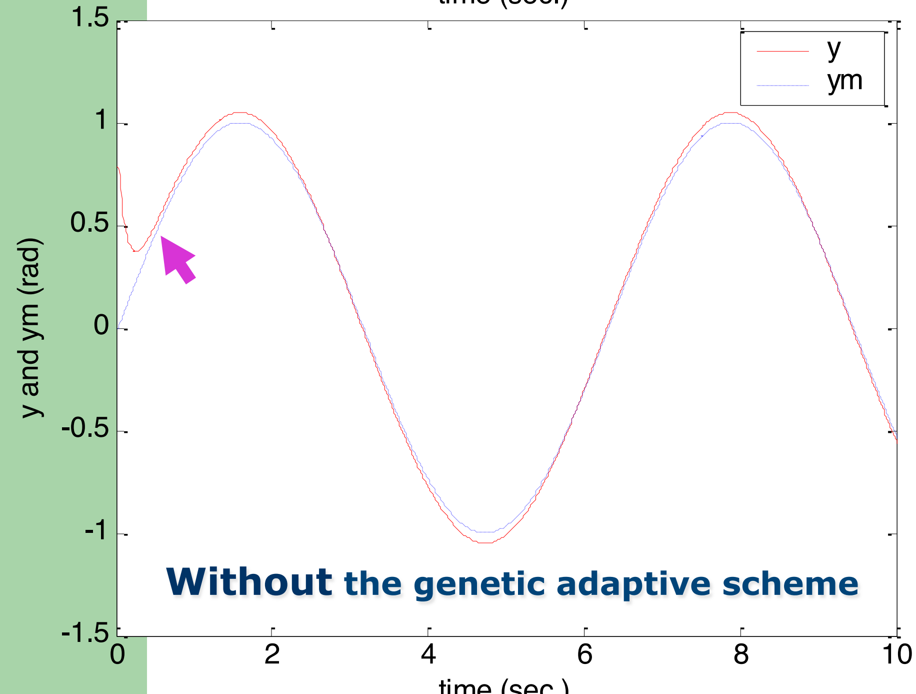


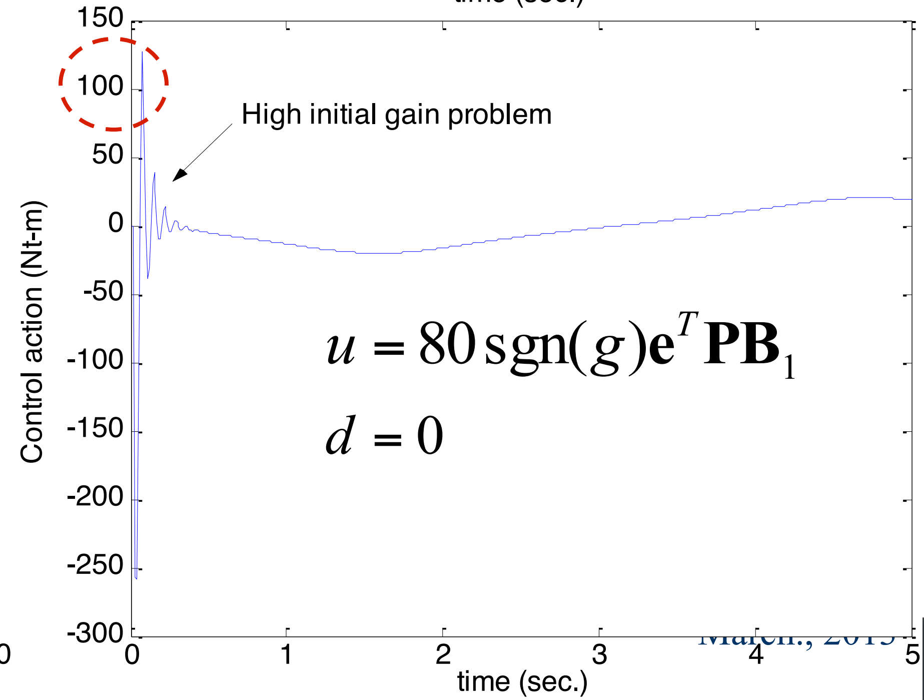
Control action



$$u = 80\,\mathrm{sgn}(g)\mathbf{e}^T\mathbf{P}\mathbf{B}_1$$

$$d = 10\sin(10\,t)\ (Nt-m)$$



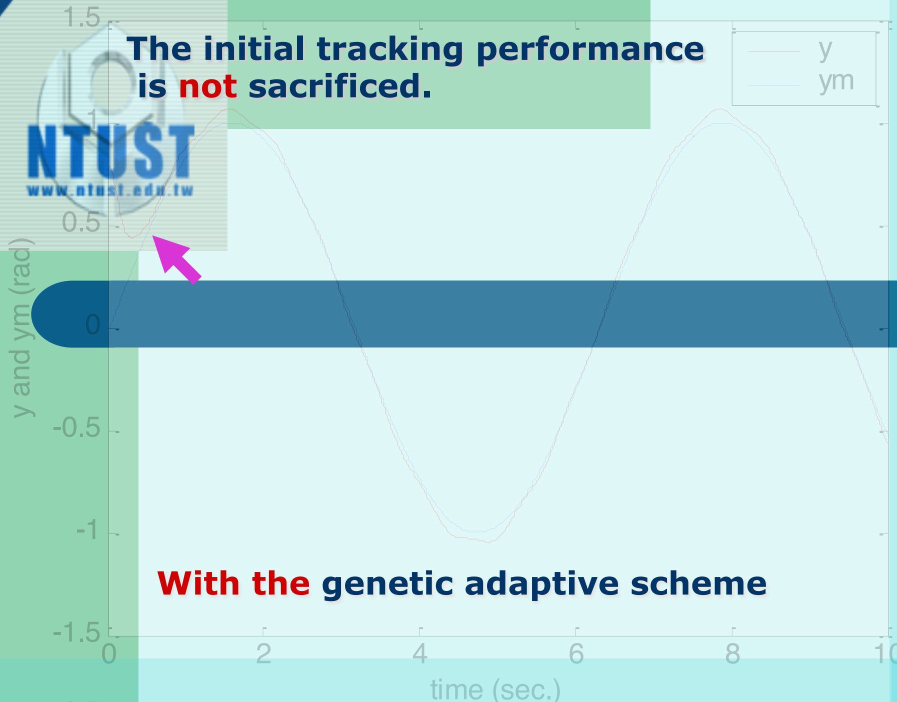


$$u = 80\,\mathrm{sgn}(g)\mathbf{e}^T\mathbf{P}\mathbf{B}_1$$

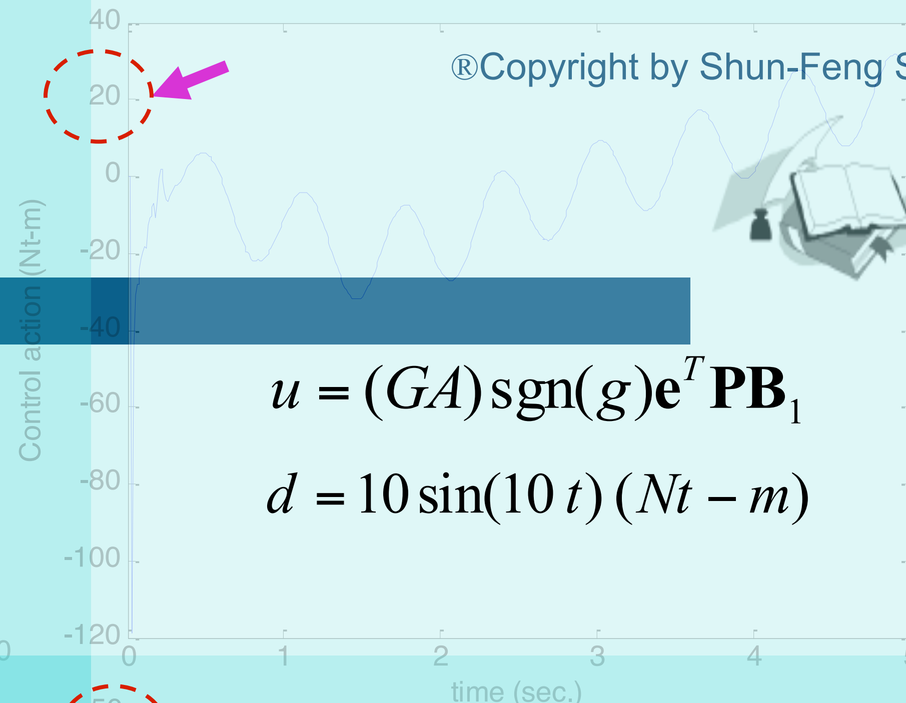
$$d = 0$$

®Copyright by Shun-Feng Su

The initial tracking performance is not sacrificed.



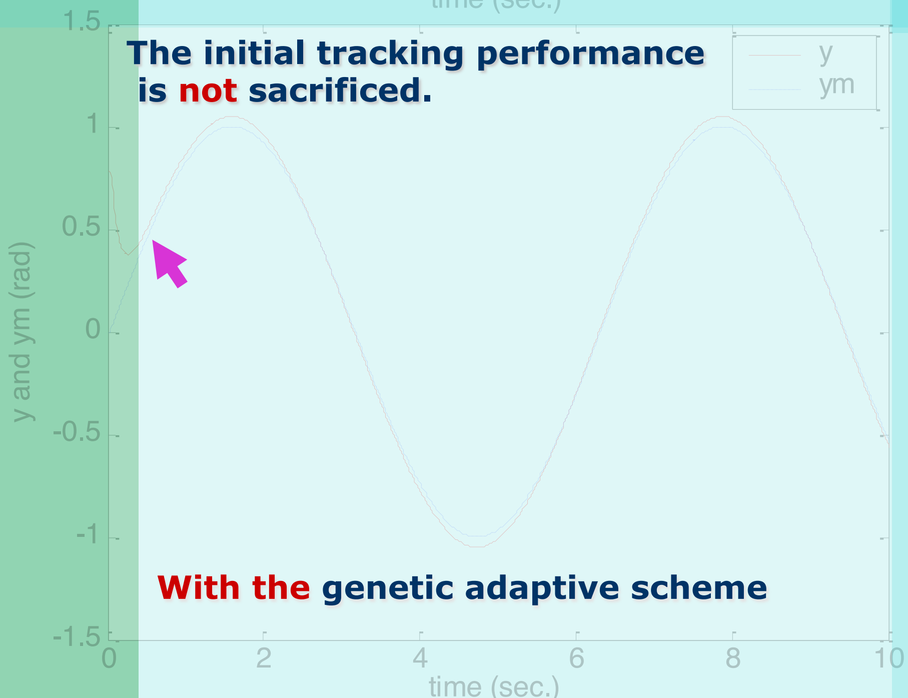
With the genetic adaptive scheme



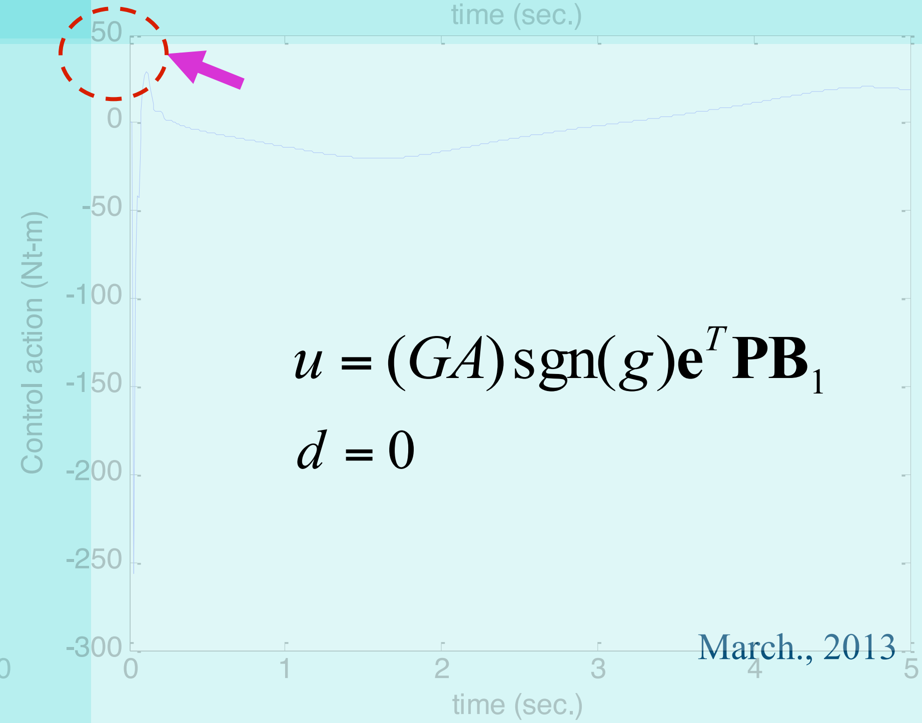
$$u = (GA)\,\mathrm{sgn}(g)\,\mathbf{e}^T \mathbf{P}\mathbf{B}_1$$

$$d = 10\sin(10\,t)\;(Nt - m)$$

The initial tracking performance is not sacrificed.



With the genetic adaptive scheme



$$u = (GA)\,\mathrm{sgn}(g)\,\mathbf{e}^T \mathbf{P}\mathbf{B}_1$$

$$d = 0$$

Robust Control

We can add an integral term to become more stable

The compensative controller is defined as

$$u_c = k_{c1}\,\mathrm{sgn}(g)\mathbf{e}^T\mathbf{PB}_1 + k_{c2}\int_0^t \mathrm{sgn}(g)\mathbf{e}^T\mathbf{PB}_1\,dt$$

$k_{c1} > 0$ $\quad$ $k_{c2} \geq 0$

$$\dot{V} = -\mathbf{e}^T\mathbf{Qe} + 2\mathbf{e}^T\mathbf{PB}\varepsilon_t - 2\mathbf{e}^T\mathbf{PB}u_c$$

By substituting u_c into $\dot{V}$.

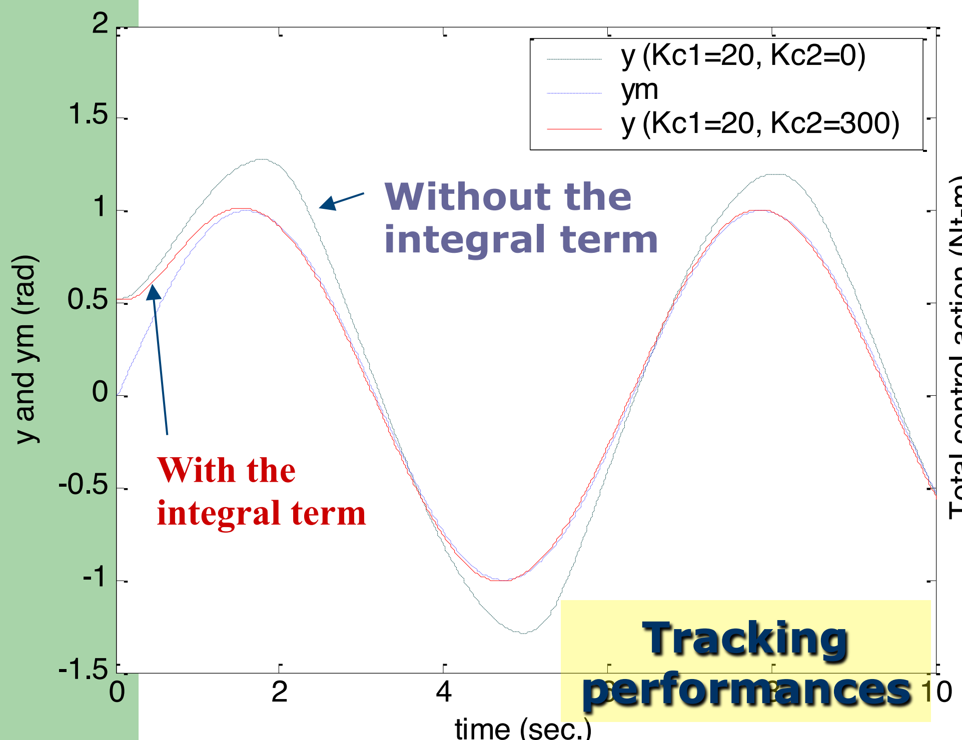
$$\dot{V} = -\mathbf{e}^T\mathbf{Qe} - \left(\sqrt{2k_{c1}|g|}\,\mathbf{e}^T\mathbf{PB}_1 - \frac{\varepsilon_t g}{\sqrt{2k_{c1}g}}\right)^2 + \left(\frac{g}{\sqrt{2k_{c1}|g|}}\right)^2\varepsilon_t^2$$

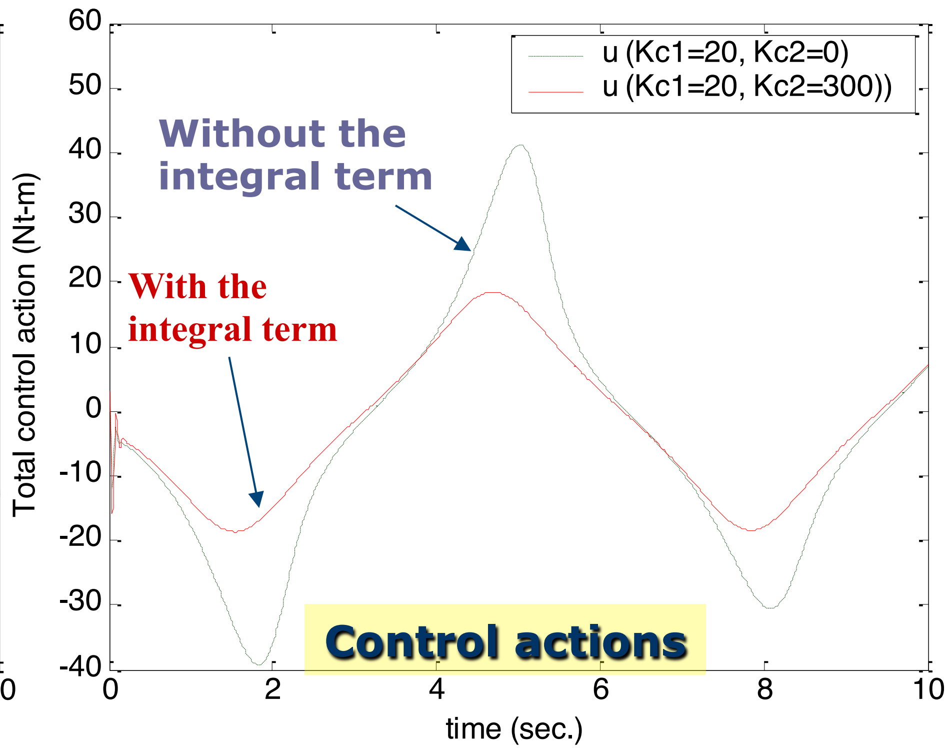
$$-2k_{c2}\int_0^t |g|(\mathbf{e}^T\mathbf{PB}_1)^2\,dt$$

Additional negative energy

Robust Control

An integral term provides a more stable edge to have better control performance.





Approximate Errors

- Another way of handling errors is to consider those errors in the controller (**error feedback controller**).
- For indirect adaptive fuzzy control, it is easy to find ways of estimating those errors and/or compensating them.
- For direct adaptive fuzzy control, it may be difficult to compensate the approximate error because it is difficult to define control errors.

Error Feedback Controller (Indirect)

An approach is proposed to improve the accuracy of estimated value. Define a modeling plant as $\hat{x}^{(n)} = \hat{f}(x) + \hat{g}(x)u(t)$

Estimated f by the fuzzy system

Estimated g by the fuzzy system

Define the estimated state error as $\tilde{x}^{(n)} = x^{(n)} - \hat{x}^{(n)}$

that is, $\tilde{x}^{(n)} = \tilde{f}(x) + \tilde{g}(x)u$

$\tilde{f} = f - \hat{f}$

$\tilde{g} = g - \hat{g}$

Approximate Errors (Indirect)

Now, define a new Lyapunov function as

$$V = \frac{1}{2} e^{T} \mathbf{P} e + \frac{1}{2} \tilde{x}_n^2 + \frac{1}{2\beta_1} \tilde{\theta}_f^T \tilde{\theta}_f + \frac{1}{2\beta_2} \tilde{\theta}_g^T \tilde{\theta}_g$$

New added term state estimated error

The idea is to minimize the modeling error while adaptive.

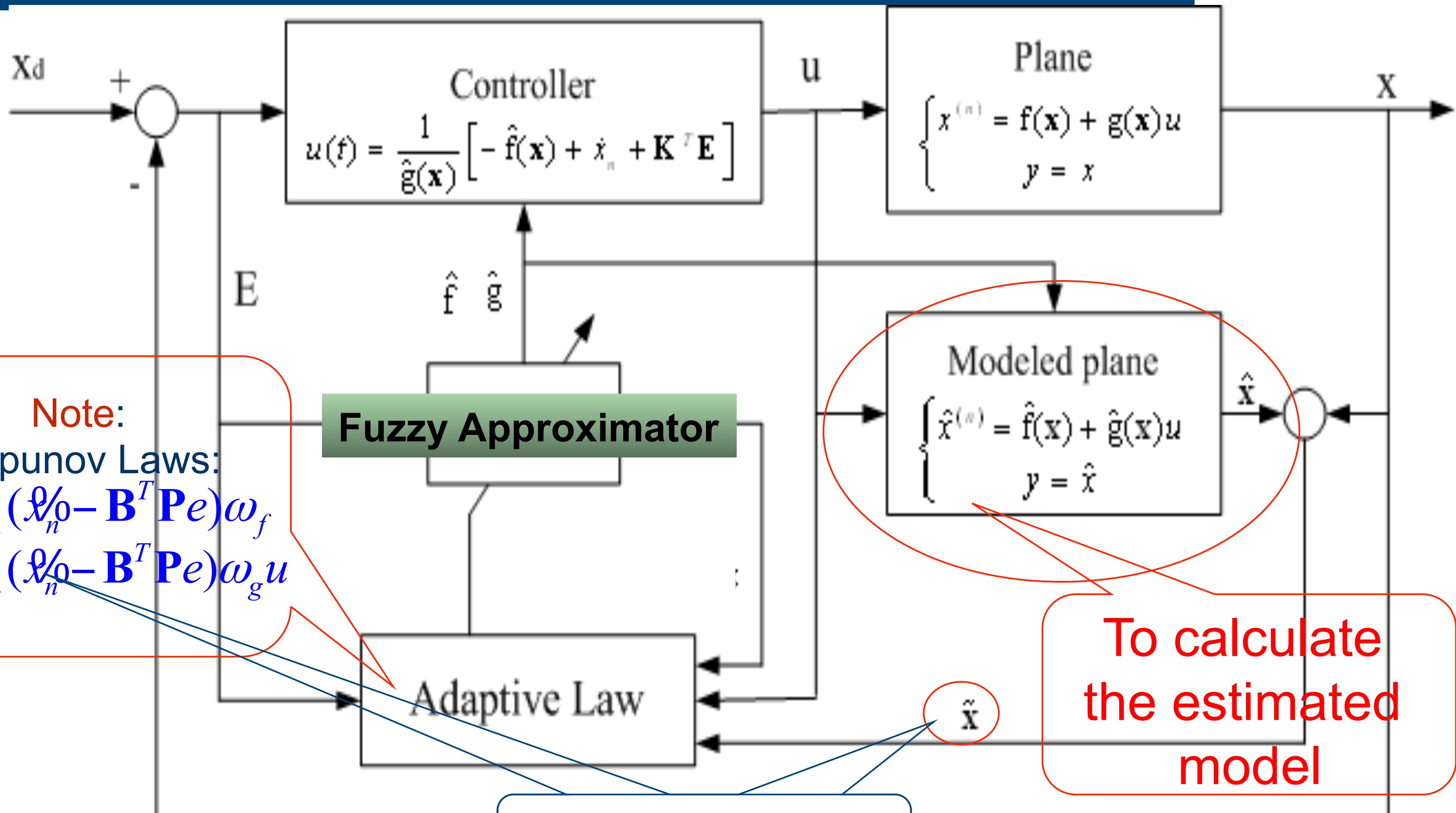
Similarly we can we the update rules as

$$\dot{\tilde{\theta}}_f = \beta_1 (\tilde{x}_n - \mathbf{B}^T \mathbf{P} e)\omega_f$$

$$\dot{\tilde{\theta}}_g = \beta_2 (\tilde{x}_n - \mathbf{B}^T \mathbf{P} e)\omega_g u$$

Approach I

Approximate Errors (Indirect)



Note:

Lyapunov Laws:

$$\dot{\theta}_f = \beta_1(\tilde{x}_n - \mathbf{B}^T\mathbf{P}e)\omega_f$$

$$\dot{\theta}_g = \beta_2(\tilde{x}_n - \mathbf{B}^T\mathbf{P}e)\omega_g u$$

To calculate the estimated model

Model error

Simulation – chaotic eq.

Simulate the chaotic equation $\ddot{x} + 0.1\dot{x} + x^3 - 12\cos(t) = u$

select vector k and matrix Q are $\mathbf{k} = \begin{bmatrix} 12 & 7 \end{bmatrix}^T, \mathbf{Q} = \begin{bmatrix} 30 & 5 \\ 5 & 30 \end{bmatrix}$

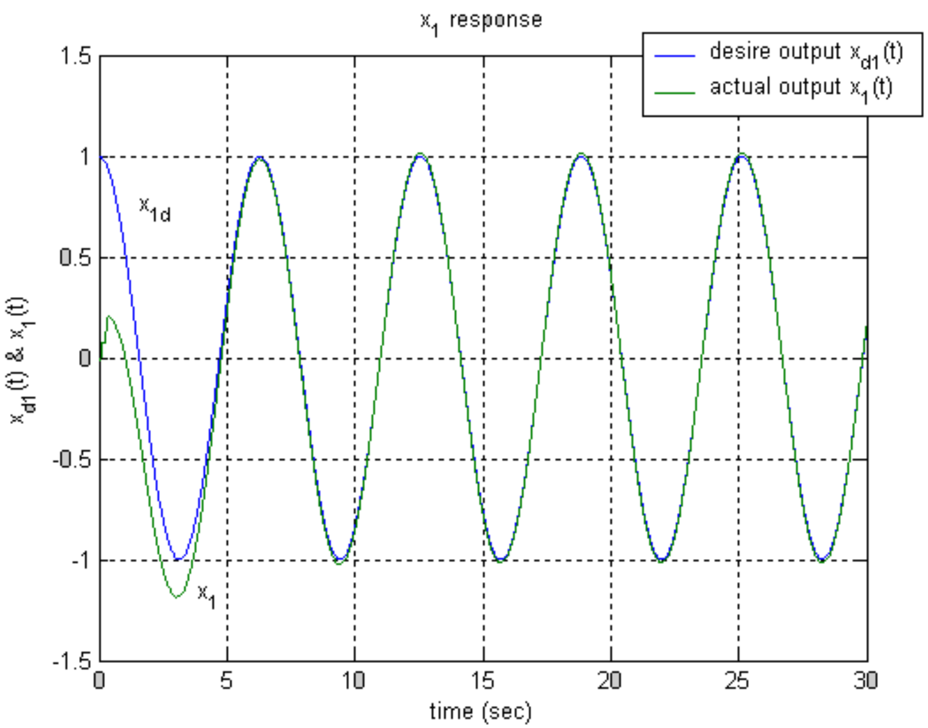
The desire output $x_d = \cos(t)$.

Some parameter β1=70,β2=0.01, gL=0.01.

There three conditions are simulated

1. *Simulation with noise-free*
2. *Simulation with disturbance*: with disturbance at 10 sec which function is $0.05\exp(-x^2 / 0.1^2)$
3. *Simulation with noise*: with noise whose mean is 0, and standard deviation is 0.01.

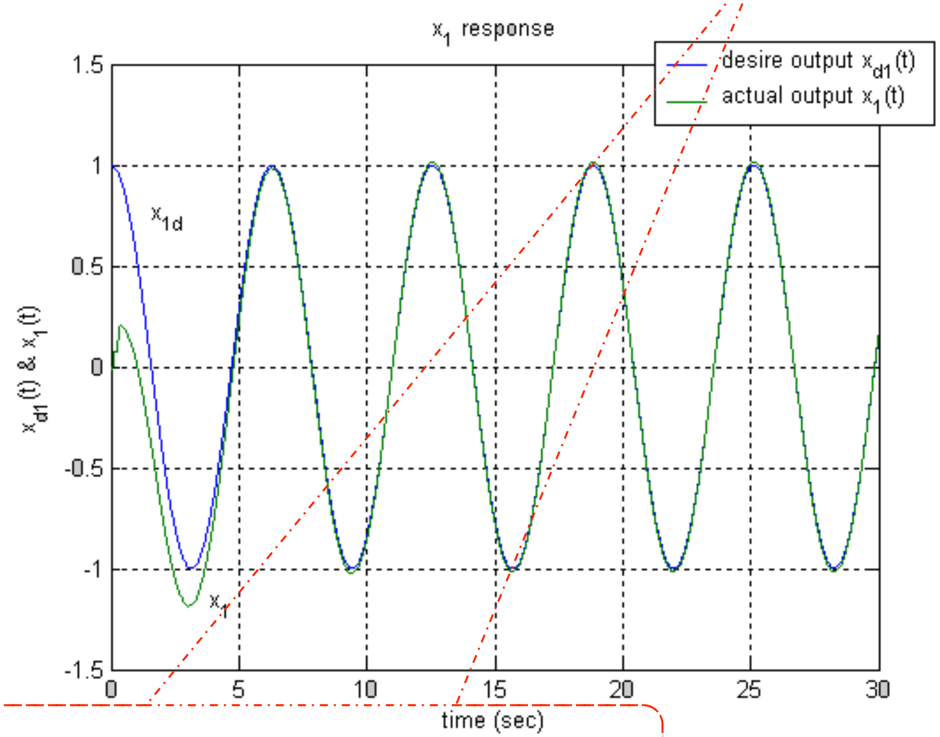
Simulation – chaotic eq.



Original approach
Tracking error converges at 0.003.

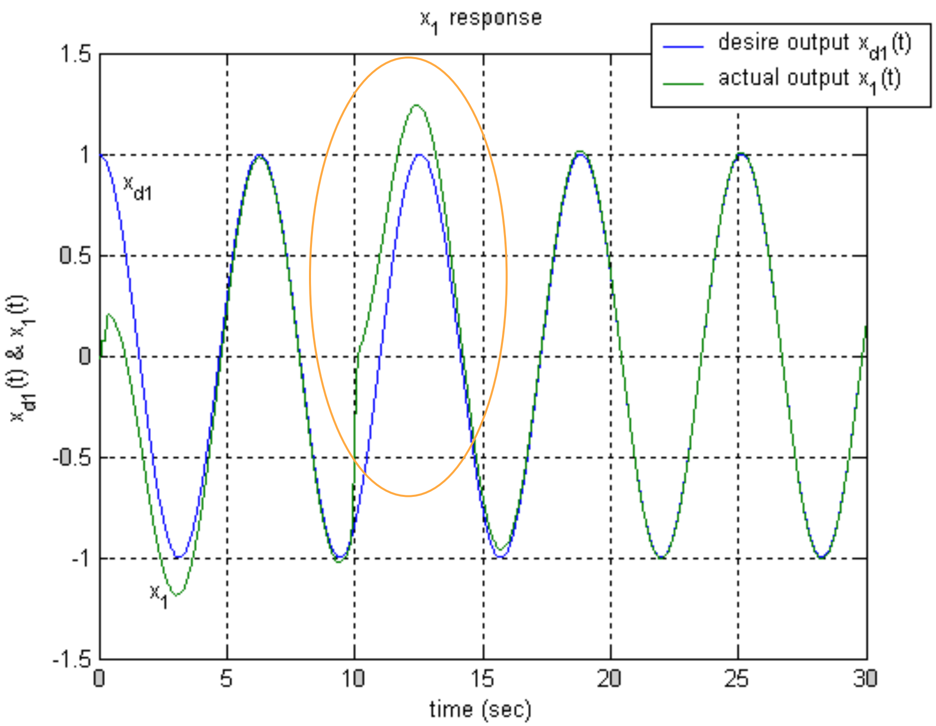
Condition 1.

Proposed approach
Tracking error converges at 0.0024.



20% improvement in error reduction

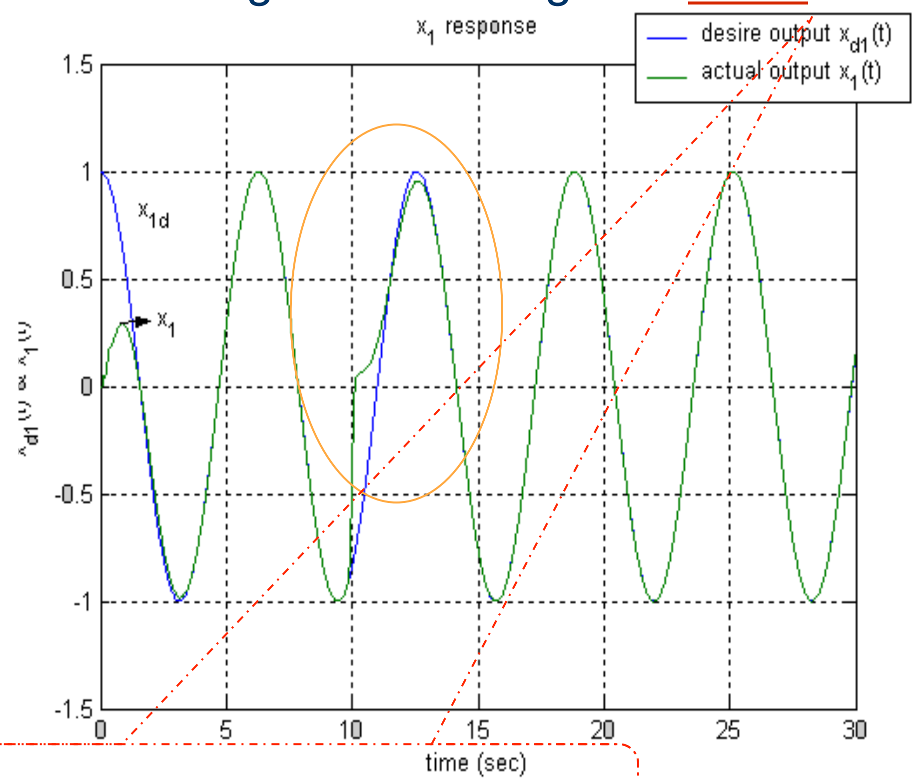
Simulation – chaotic eq.



Original approach
Tracking error converges at 0.0041.

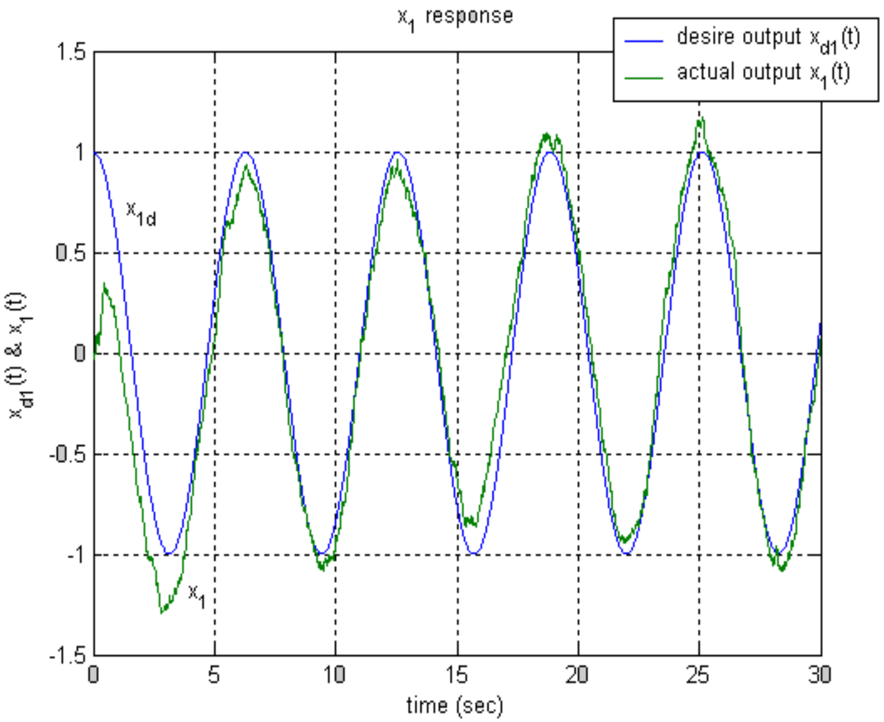
Condition 2.

Proposed approach
Tracking error converges at 0.003.



26.8% improvement in error reduction

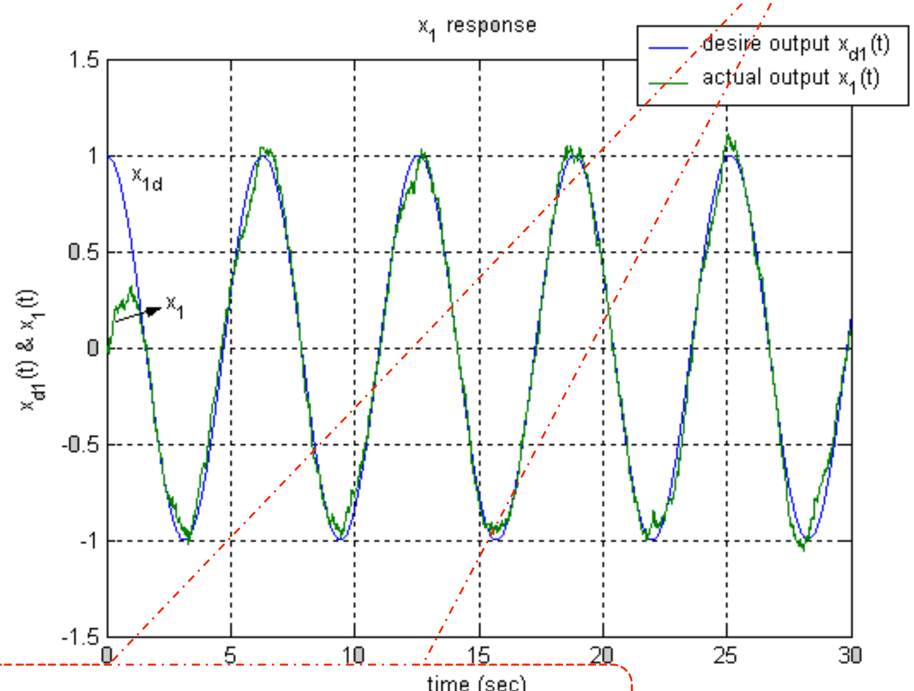
Simulation – chaotic eq.



Original approach
Tracking error converges at 0.0032.

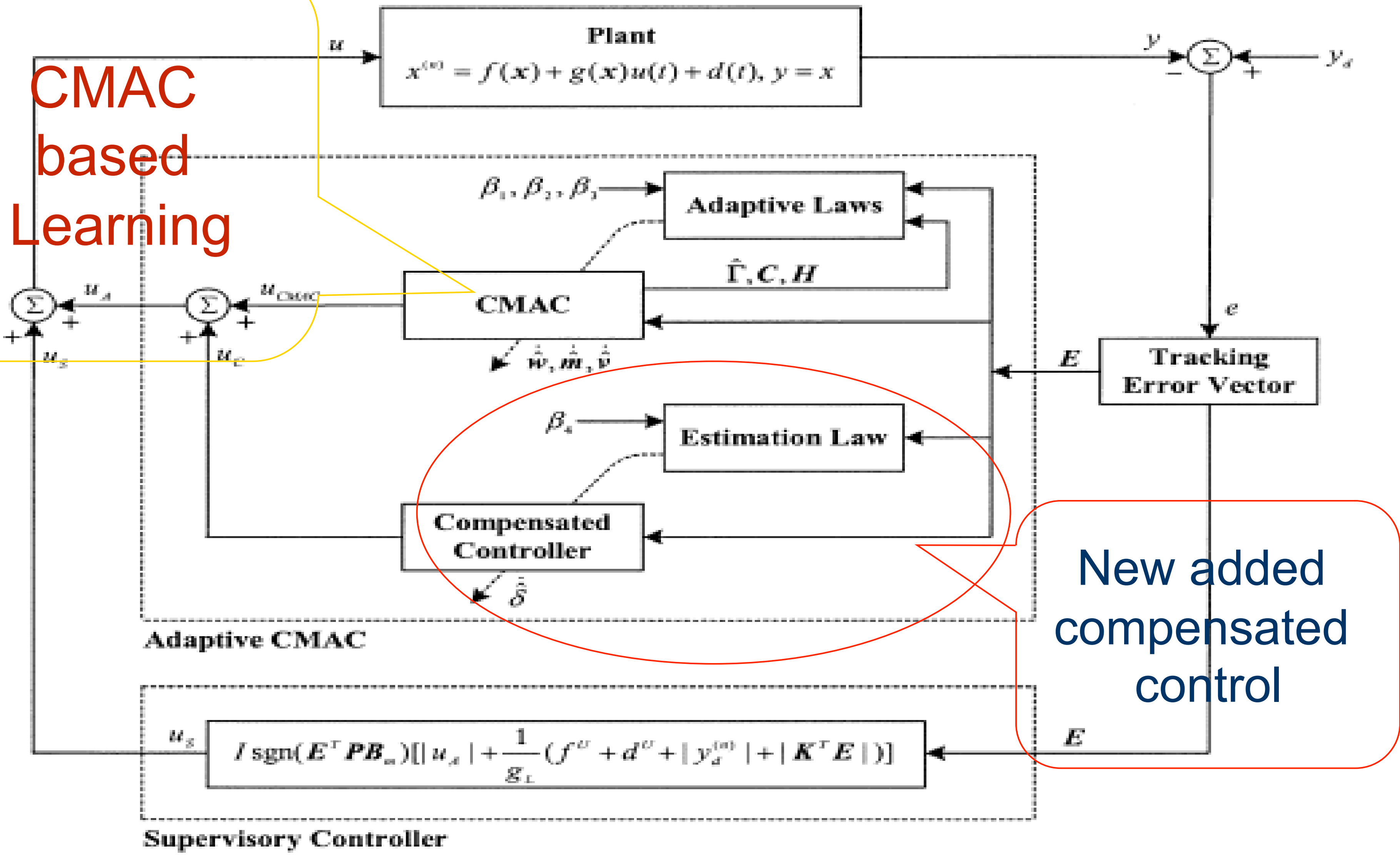
Condition 3.

Proposed approach
Tracking error converges at 0.0025.



21.9% improvement in error reduction

Another Approach [8]



Approach II

$$V(\mathbf{E}, \tilde{\mathbf{w}}, \tilde{\mathbf{m}}, \tilde{\mathbf{v}}, \tilde{\delta}, t) = \frac{1}{2}\mathbf{E}^T\mathbf{P}\mathbf{E} + \frac{1}{2\beta_1}\tilde{\mathbf{w}}^T\tilde{\mathbf{w}} + \frac{1}{2\beta_2}\tilde{\mathbf{m}}^T\tilde{\mathbf{m}} + \frac{1}{2\beta_3}\tilde{\mathbf{v}}^T\tilde{\mathbf{v}} + \frac{1}{2\beta_4}\tilde{\delta}^2$$

CMAC based parameter Learning

$$\dot{\hat{\mathbf{w}}} = \beta_1 \mathbf{E}^T\mathbf{P}\mathbf{B}_m\hat{\Gamma}$$

$$\dot{\hat{\mathbf{m}}} = \beta_2 \mathbf{E}^T\mathbf{P}\mathbf{B}_m\mathbf{C}\hat{\mathbf{w}}$$

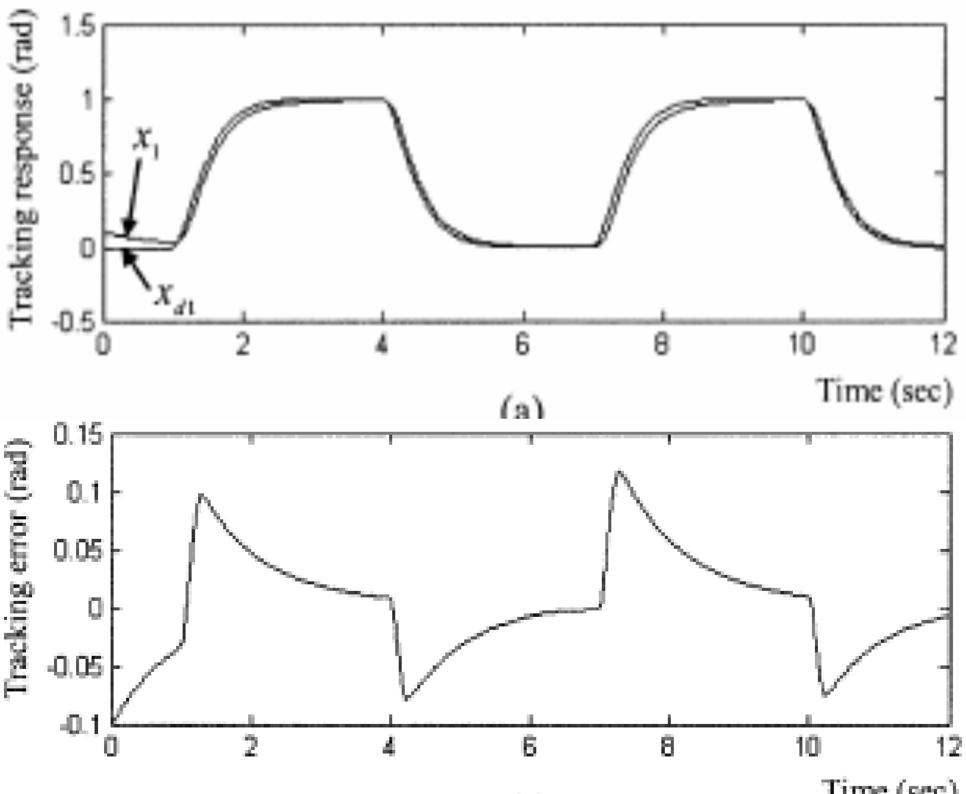
$$\dot{\hat{\mathbf{v}}} = \beta_3 \mathbf{E}^T\mathbf{P}\mathbf{B}_m\mathbf{H}\hat{\mathbf{w}}$$

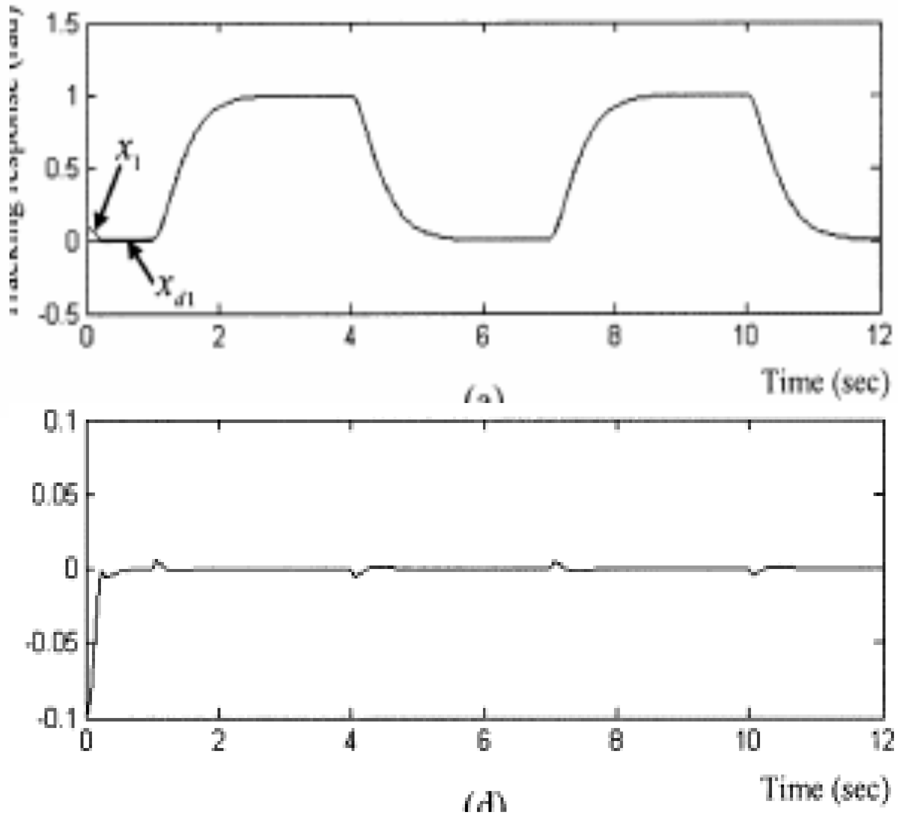
$$u_C = \hat{\delta}\,\mathrm{sgn}(\mathbf{E}^T\mathbf{P}\mathbf{B}_m)$$

$$\dot{\hat{\delta}} = \beta_4 |\mathbf{E}^T\mathbf{P}\mathbf{B}_m|.$$

compensated control also has some bounded adaptive effects (discussed later)

Simulation

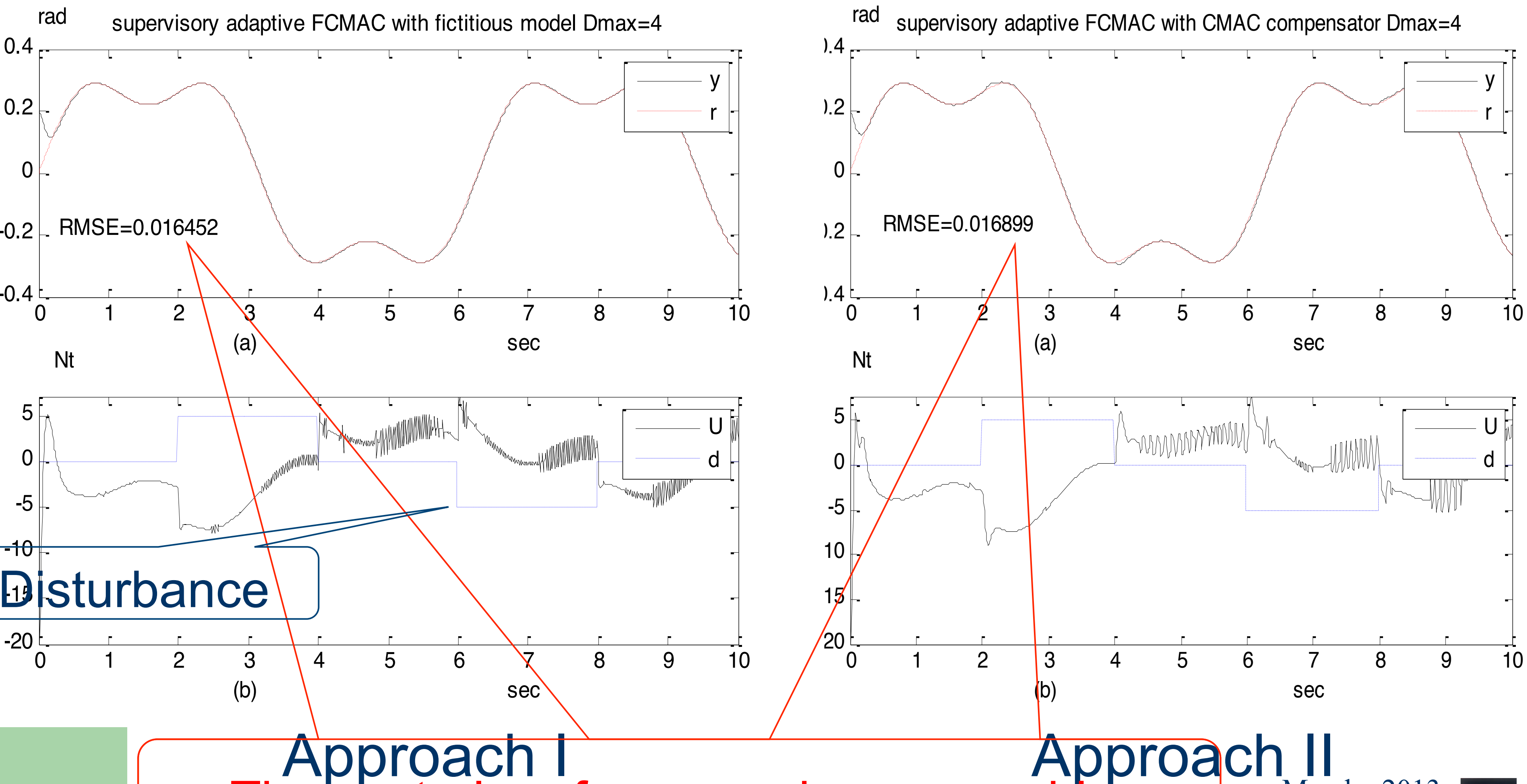




Original approach

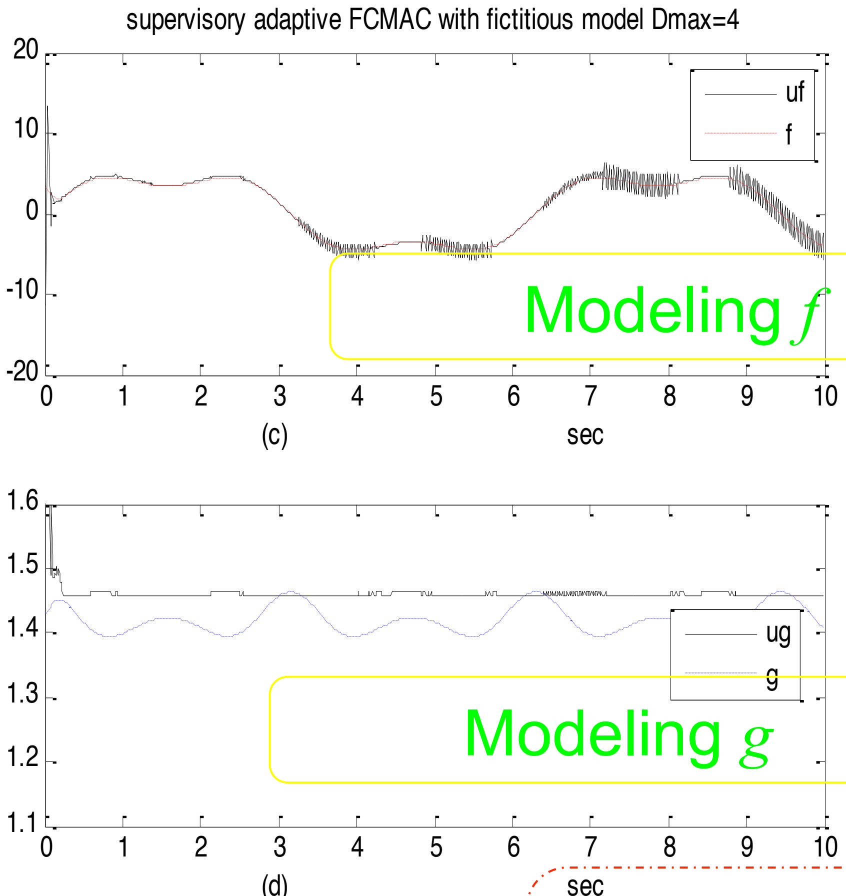
Approach II

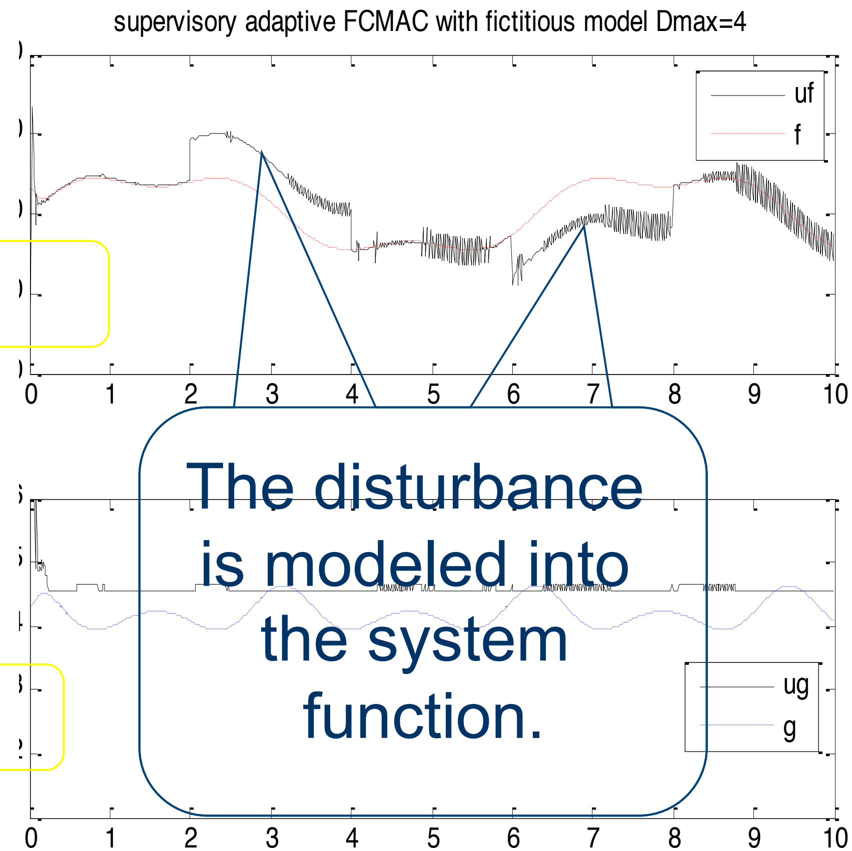
Simulation



The control performance is comparable.

Simulation – Approach I





Modeling f

Modeling g

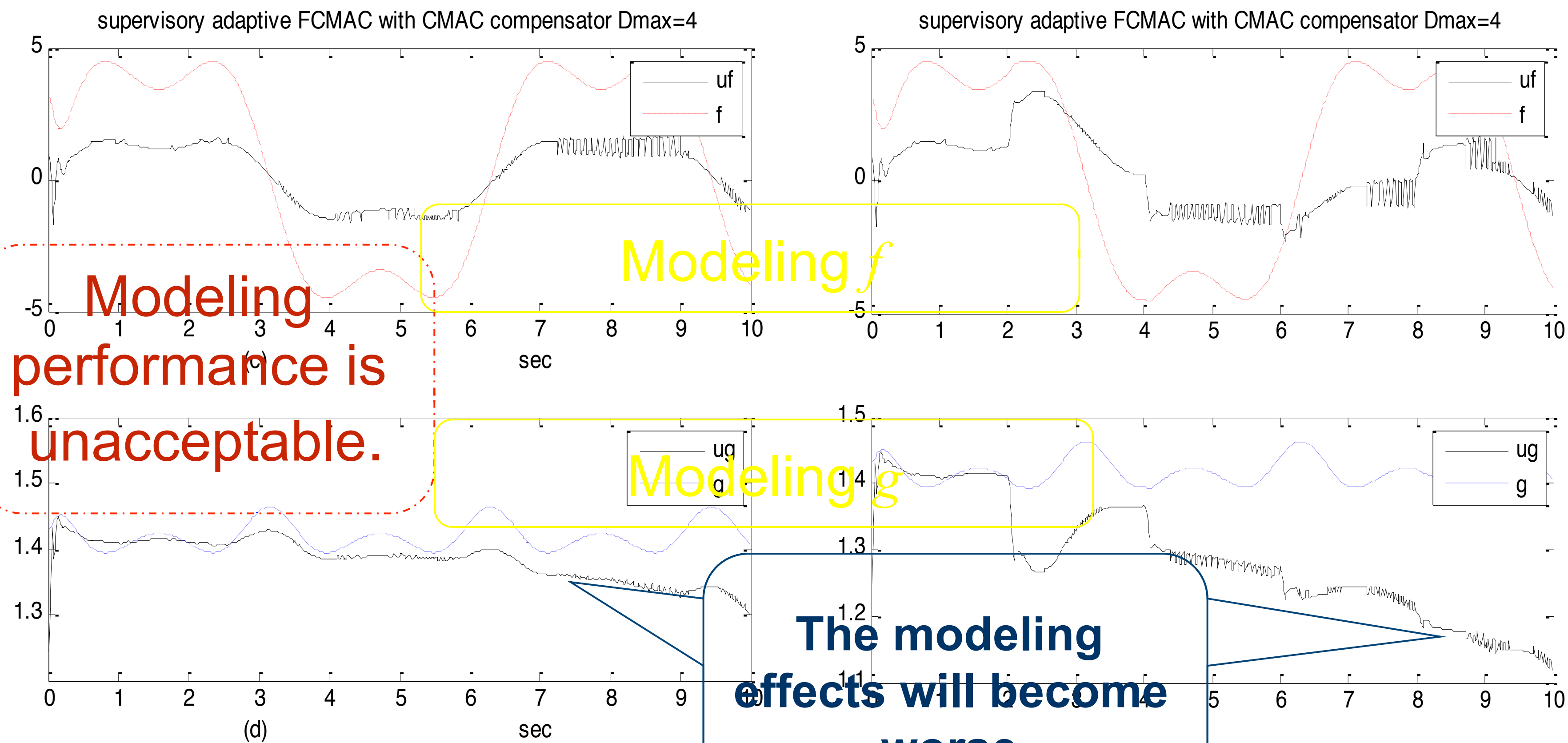
The disturbance is modeled into the system function.

Without disturbance

Modeling performance is acceptable.

With disturbance

Simulation – Approach II



Without disturbance

With disturbance

Error Feedback Controller (Direct)

A way of defining errors must be developed for direct adaptive fuzzy control.

***Definition*:**

A control system is said to have the finite L_2-gain property if there exists an assignable finite gain $\delta > 0$ and a bias constant $\gamma_{bias} \in \mathbb{R}^+$ representing the initial condition such that the following inequality holds $\left\|\mathbf{e}_{T_f}\right\|_{L_2} \le \delta \left\|\varepsilon_{t\,T_f}\right\|_{L_2} + \gamma_{bias}$

where $\left\|\mathbf{e}_{T_f}\right\|_{L_2} = \sqrt{\int_0^{T_f} \mathbf{e}^T \mathbf{e}\, dt}$ represents the system output energy and $\left\|\varepsilon_{t\,T_f}\right\|_{L_2} = \sqrt{\int_0^{T_f} \varepsilon_t^2 dt}$ represents the system input energy.

Error Feedback Controller (Direct)

The inequity $\left\|\mathbf{e}_{T_f}\right\|_{L_2} \leq \delta \left\|\varepsilon_{tT_f}\right\|_{L_2} + \gamma_{bias}$ indicates that -- the tracking control error is bounded in a region around origin, the size of the region can be arbitrarily small with the choice of δ. Thus, the following equation is guaranteed as **[6]**.

$$\lim_{t \to \infty} u_d + u_c = u^*$$

[6] E. Kim, "A fuzzy disturbance observer and its application to control," *IEEE Trans. Fuzzy Systems*, vol. 10, no. 1, Feb. 2002.

Consider that

$$u_c \cong u^* - u_d = \tilde{\boldsymbol{\theta}}_d^T \boldsymbol{\omega}_d$$

$\tilde{\boldsymbol{\theta}}_d$ is estimated as $\tilde{\boldsymbol{\theta}}_d = (u_c \boldsymbol{\omega}_d^{-1})^T$.

Error Feedback Controller (Direct)

The estimative value can be multiplied by another adaptive rate $\beta_m \|\mathbf{e}\|_2$ as

$$\dot{\tilde{\boldsymbol{\theta}}}_d = \beta_m \|\mathbf{e}\|_2 (u_c \boldsymbol{\omega}_d^{-1})^T$$

By substituting the estimative value into the earlier adaptive law, the proposed adaptive law is found as follows:

$$\dot{\boldsymbol{\theta}}_d = \alpha_m \operatorname{sgn}(g)\mathbf{e}^T \mathbf{P}\mathbf{B}_1 \boldsymbol{\omega}_d + \beta_m \|\mathbf{e}\|_2 (u_c \boldsymbol{\omega}_d^{-1})^T$$

$\|\mathbf{e}\|_2 = \sqrt{|e_1|^2 + |e_2|^2 + \cdots + |e_n|^2}$ is a simple adaptation scheme to enhance the learning stability more.

Q.E.D.

Simulations

Simulation 2: The *learning speed tests* are illustrated in this simulation.

Adaptive rate 2

Adaptive rate 1

$k_{c1} = 15$ is still used, the other parameters α_m and β_m will be adjusted to show the change of the learning speed.

1. If $\beta_m = 0$, then the approximate error feedback term $\beta_m \|\mathbf{e}\|_2 (u_c \boldsymbol{\omega}_d^{-1})^T$ is not used.

2. The value of α_m is also increased to illustrate the effects of the learning speed.

Simulate results

α_m	β_m	Reached Time (sec.) and Cycle
0.1	20	25.12 (4th cycle)
0.1	0	Unstable
1	20	25.12 (4th cycle)
1	0	Unstable
10	20	18.84 (3th cycle)
10	0	Unstable
20	20	18.84 (3th cycle)
20	0	43.96 (7th cycle)
30	20	18.84 (3th cycle)
30	0	31.40 (5th cycle)
40	20	18.84 (3th cycle)

$$\dot{\boldsymbol{\theta}}_d = \alpha_m \operatorname{sgn}(g)\mathbf{e}^T\mathbf{P}\mathbf{B}_1\boldsymbol{\omega}_d + \beta_m\|\mathbf{e}\|_2(u_c\boldsymbol{\omega}_d^{-1})^T$$

$$\dot{\boldsymbol{\theta}}_d = \alpha_m \operatorname{sgn}(g)\mathbf{e}^T\mathbf{P}\mathbf{B}_1\boldsymbol{\omega}_d$$

α_m

1. A suitable β_m provides **more stable learning speed** even if the adaptive rates are different.

2. It can be found that **the selection of the adaptive rate can be relaxed** because the proposed approach.

✓ *The stable learning speed is guaranteed.*

✓ *The initial learning stability is guaranteed.*

NTUST

Adaptive Fuzzy Control

There are problems in the above approaches:

- Approximate errors and robust control
- **Initialization and supervisory control**.
- Parameter drifting

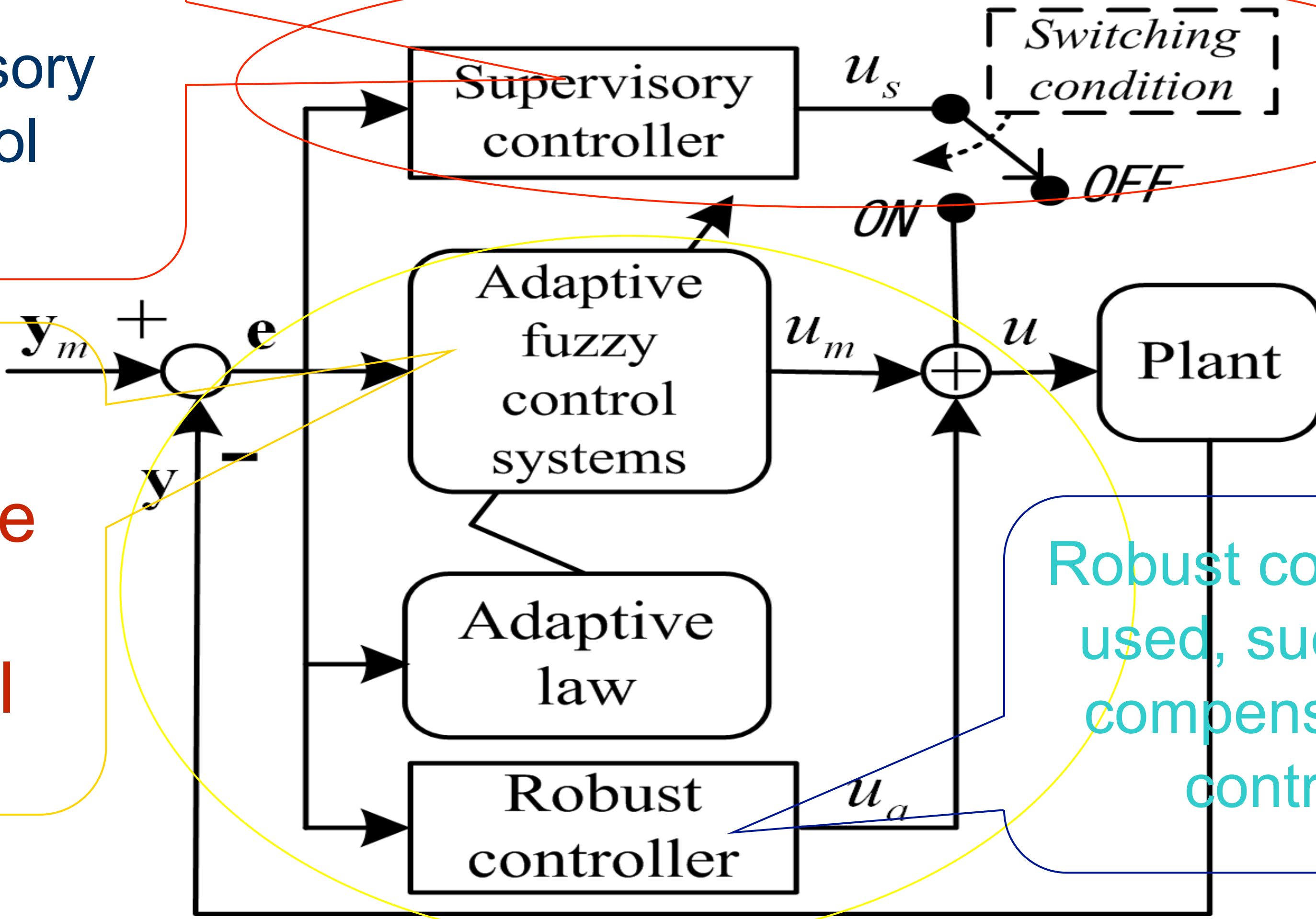
Initialization

- Initial status (initial states and initial parameter values) may cause various problems for a learning system.
- A so-called supervisory controller [3,5] is often used and the effects are satisfactory. In above examples, all use supervisory controllers. It is similar to hitting control for sliding model control.
- The supervisory controller is proposed in the early version of adaptive fuzzy control and can also act as one kind of robust control.

Supervisory Control

supervisory control

Adaptive fuzzy control



Robust control if used, such as compensated control

Supervisory Control

- The control in supervised control is $u = u_p + u_s$, where u_p is the approximated perfect control law and u_s is the supervisory controller.
- Consider the derivative of Lyapunov function $\dot{V}_s = -(1/2)\mathbf{e}^T\mathbf{Q}_s\mathbf{e} + \mathbf{e}^T\mathbf{P}_s\mathbf{B}(u^* - u_p - u_s)$, where u^* is the optimal control.
- Thus, if u_s is large enough, the derivate of *V* will be always negative.

It is ε_I

Supervisory Control

Controller $u = u_p + u_s$

Supervisory controller

$$u_s = \text{sgn}(g)(-C_{su})$$

$$\begin{cases} \dot{V}_s = -(1/2)\mathbf{e}^T\mathbf{Q}_s\mathbf{e} + \mathbf{e}^T\mathbf{P}_s\mathbf{B}_1|g|C_{su} + \mathbf{e}^T\mathbf{P}_s\mathbf{B}_1(gu_s^* - gu_m) \\ \mathbf{B}_1 = [0 \quad 0 \quad \cdots \quad 0 \quad 1]^T \\ gu_s^* = -f + y_m^{(n)} + \mathbf{k}_s^T\mathbf{e} \end{cases}$$

$$\dot{V}_s = -\frac{1}{2}\mathbf{e}^T\mathbf{Q}_s\mathbf{e} + \mathbf{e}^T\mathbf{P}_s\mathbf{B}_1|g|C_{su} + \mathbf{e}^T\mathbf{P}_s\mathbf{B}_1(-f + y_m^{(n)} + \mathbf{k}_s^T\mathbf{e} - gu_p)$$

Supervisory Control

$$\dot{V}_s = -\frac{1}{2}\mathbf{e}^T\mathbf{Q}_s\mathbf{e} + \mathbf{e}^T\mathbf{P}_s\mathbf{B}_1|g|C_{su} + \mathbf{e}^T\mathbf{P}_s\mathbf{B}_1(-f + y_m^{(n)} + \mathbf{k}_s^T\mathbf{e} - gu_p)$$

$$\dot{V}_s \le -(1/2)\mathbf{e}^T\mathbf{Q}_s\mathbf{e} + \mathbf{e}^T\mathbf{P}_s\mathbf{B}_1 K_{su} C_{su} + |\mathbf{e}^T\mathbf{P}_s\mathbf{B}_1|(f_{up} + |y_m^{(n)}| + |\mathbf{k}_s^T\mathbf{e}| + (|gu_m|)_{up})$$

Stable condition : $\dot{V}_s \le 0$

- $K_{su} \ge 1$: A positive constant.
- $f_{up} > 0$: The upper-bound of f.
- $(|gu_p|)_{up}$: The upper-bound of the gu_m.

Design

$$\begin{cases} C_{su} = -\operatorname{sgn}(\mathbf{e}^T\mathbf{P}_s\mathbf{B}_1)(f_{up} + |y_m^{(n)}| + |\mathbf{k}_s^T\mathbf{e}| + (|gu_p|)_{up}) \\ u_s = \operatorname{sgn}(g)(-C_{su}) \end{cases}$$

★**Design result :**

$$u_s = \operatorname{sgn}(g)\operatorname{sgn}(\mathbf{e}^T\mathbf{P}_s\mathbf{B}_1)\, K_{su}\, (f_{up} + |y_m^{(n)}| + |\mathbf{k}_s^T\mathbf{e}| + (|gu_p|)_{up})$$

Supervisory Control

Thus, the supervisory controller can be selected as

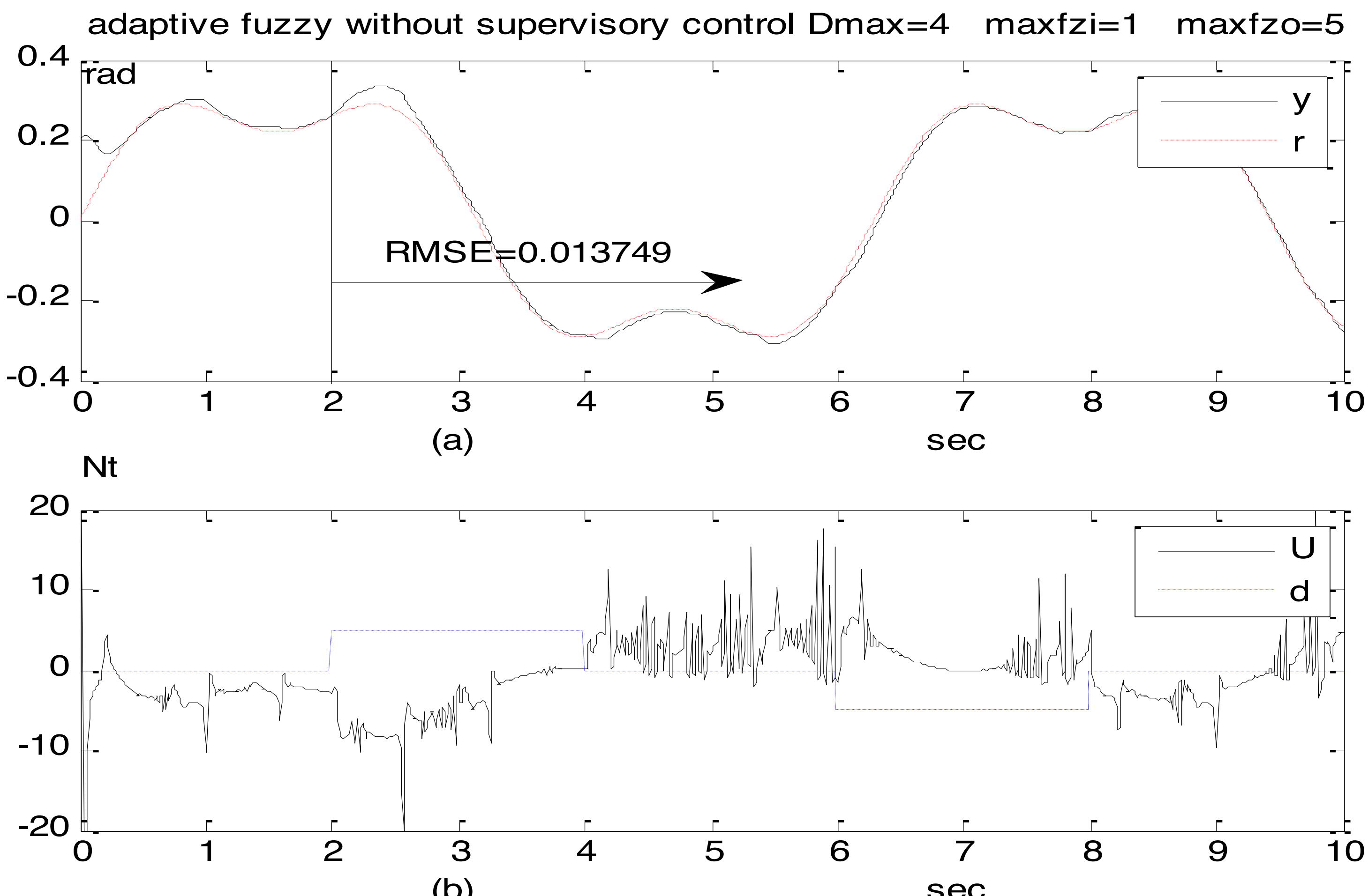
$$u_s = \text{sgn}(g)\,\text{sgn}(\mathbf{e}^T \mathbf{P}_s \mathbf{B}_1)\; K_s \;\left(f_{up} + \left|y_m^{(n)}\right| + \left|\mathbf{k}_s^T \mathbf{e}\right| + (\left|g u_p\right|)_{up}\right)$$

where the subscript *up* is the upper bound of that function and K_s is a constant.

It can be found that the supervisory controller is a function of the upper bound of the system function.

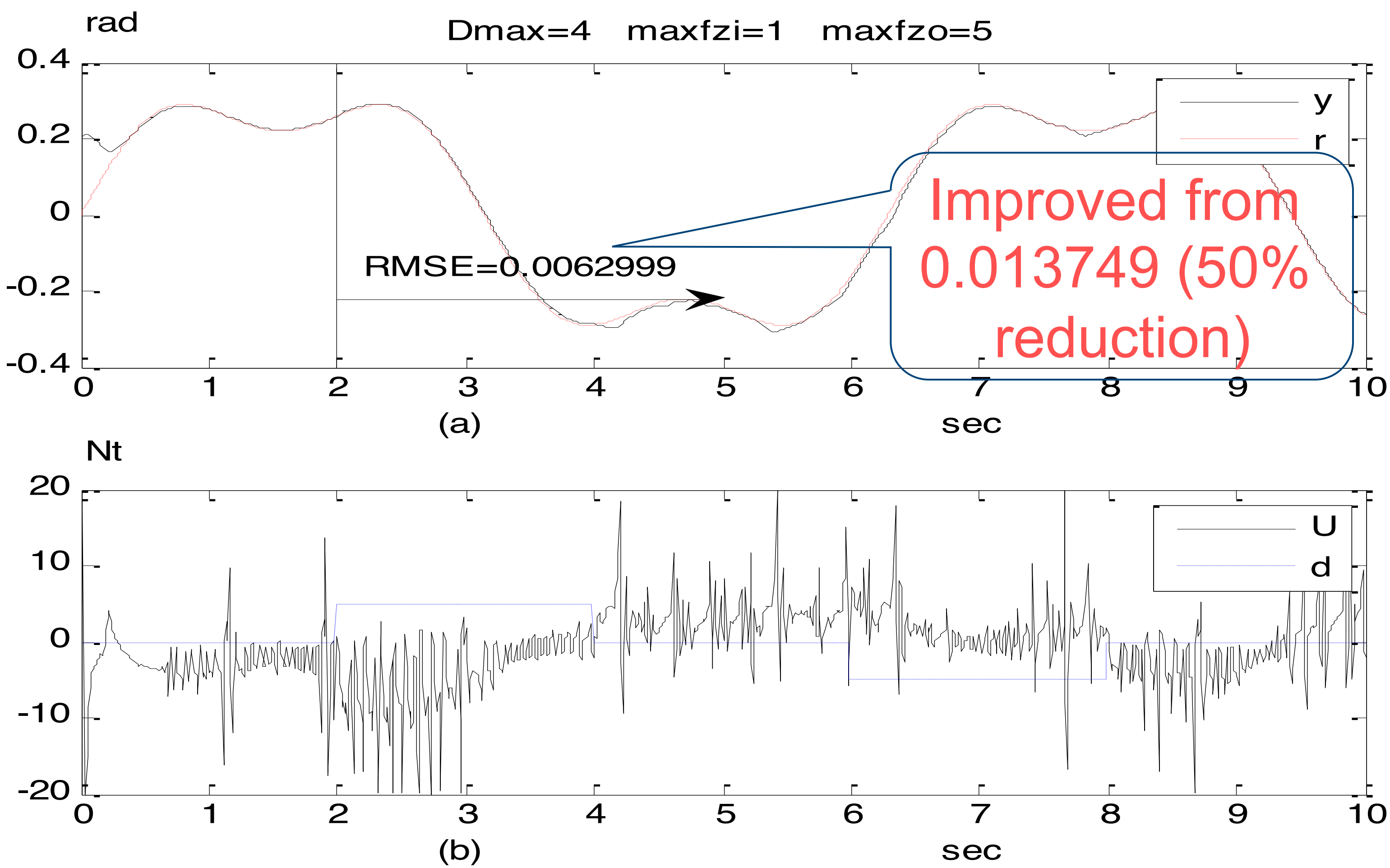
If the bound is not properly selected, the control performance may not be satisfactory.

Simulation



Without supervisory control

Simulation



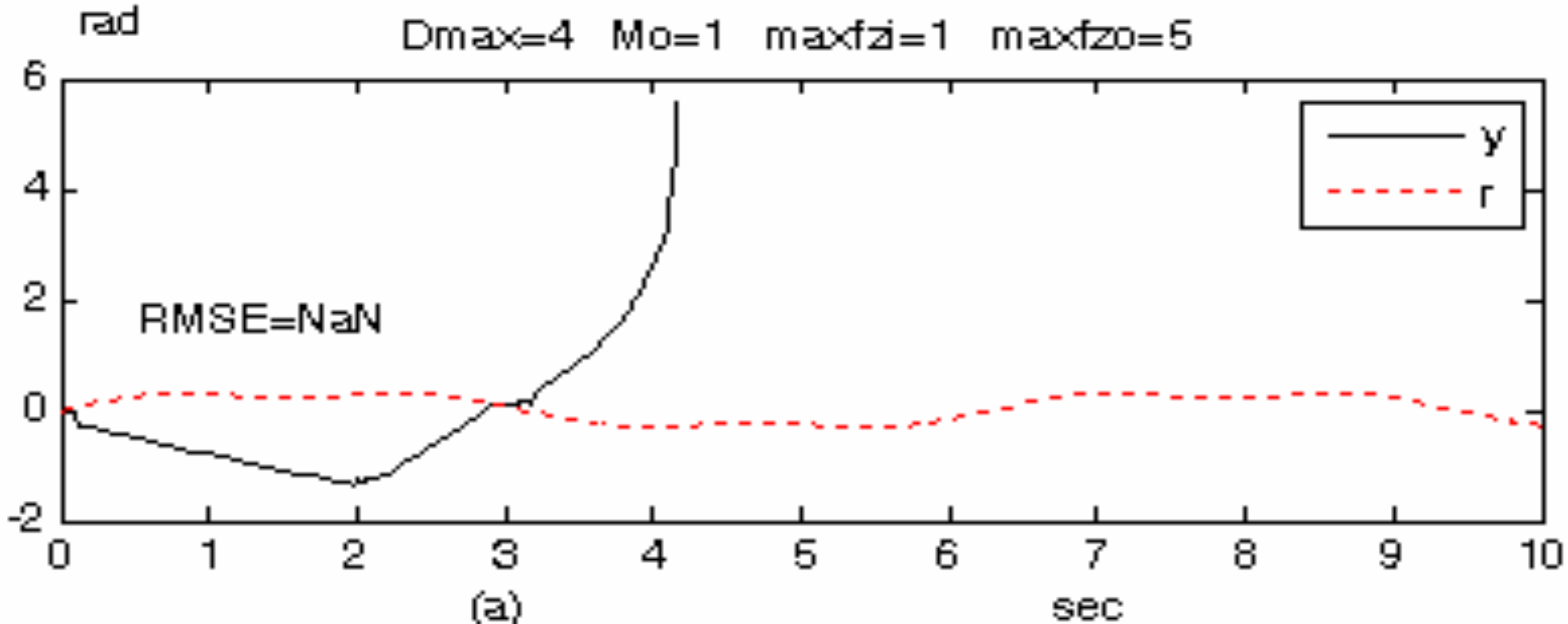
With supervisory control

Supervisory Control

Consider another system as

$$\dot{x} = e^x - 1 + u$$

This system do not have a bound for the system function. → The system diverges.



Supervisory Control

The problem is that the bound is a function of *f*.

$$u_s = \mathrm{sgn}(g)\,\mathrm{sgn}(\mathbf{e}^T \mathbf{P}_s \mathbf{B}_1)\; K_s \;\left(f_{up} + \left|y_m^{(n)}\right| + \left|\mathbf{k}_s^T \mathbf{e}\right| + (\left|g u_p\right|)_{up}\right)$$

The idea is to use previous control action so that the bounded for the perfect control law can be reduced so that the supervisory control can easily be implemented.

The term of the system function becomes the difference of the system function, of which the bound is much smaller than that of the system function.

Supervisory Control

$$u^* = u(k) = u(k-1) + g_0[y_m^{(n)}(k) - y_m^{(n)}(k-1) - f(k) + f(k-1) + \hat{k}^T(\hat{e}(k) - \hat{e}(k-1)) + err_g]$$

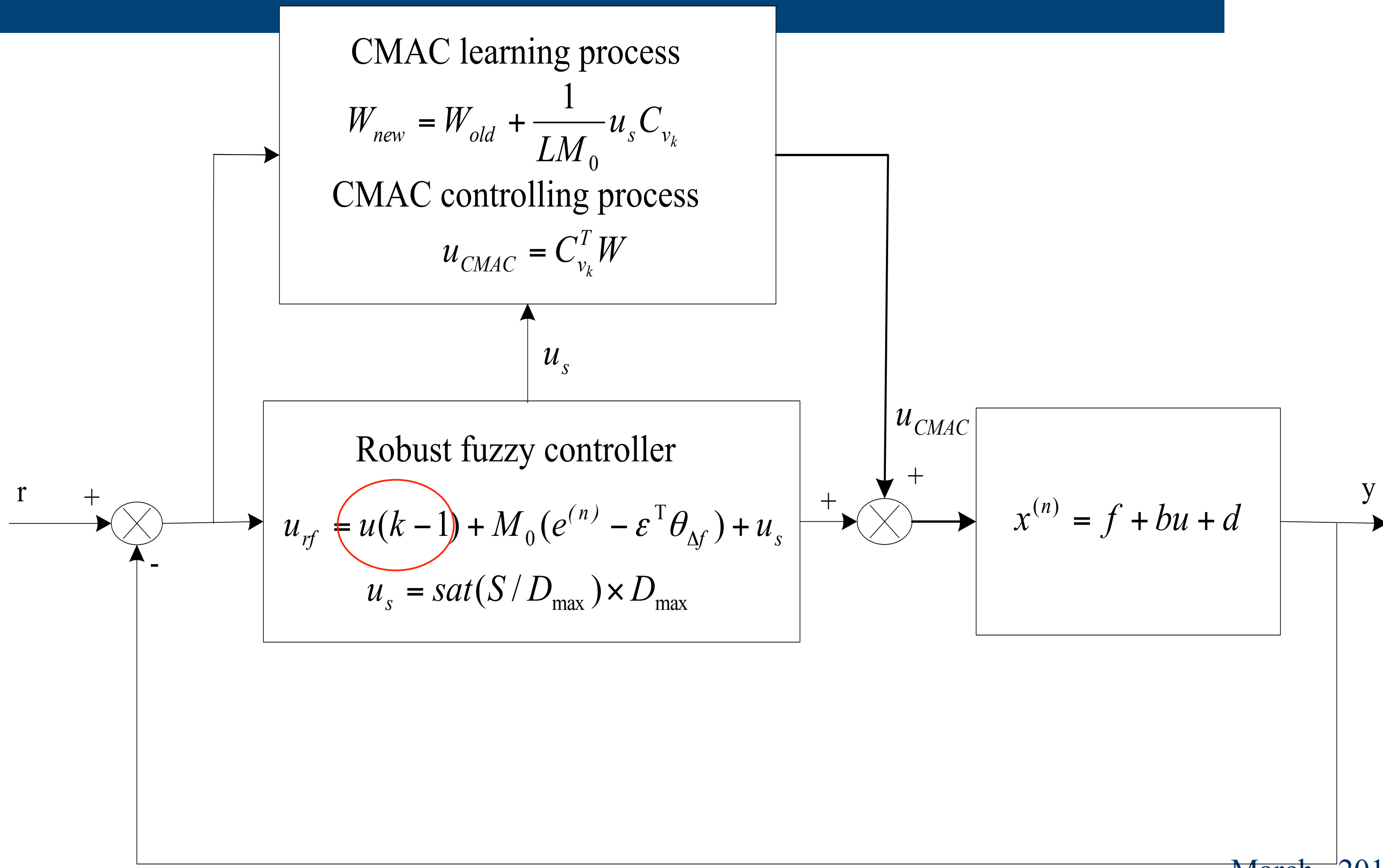
$$u^* = u(k-1) + g_0(e^{(n)} - \Delta f) + g_0 E$$

$$E = err_g + err_{tracking} + err_{transition}$$

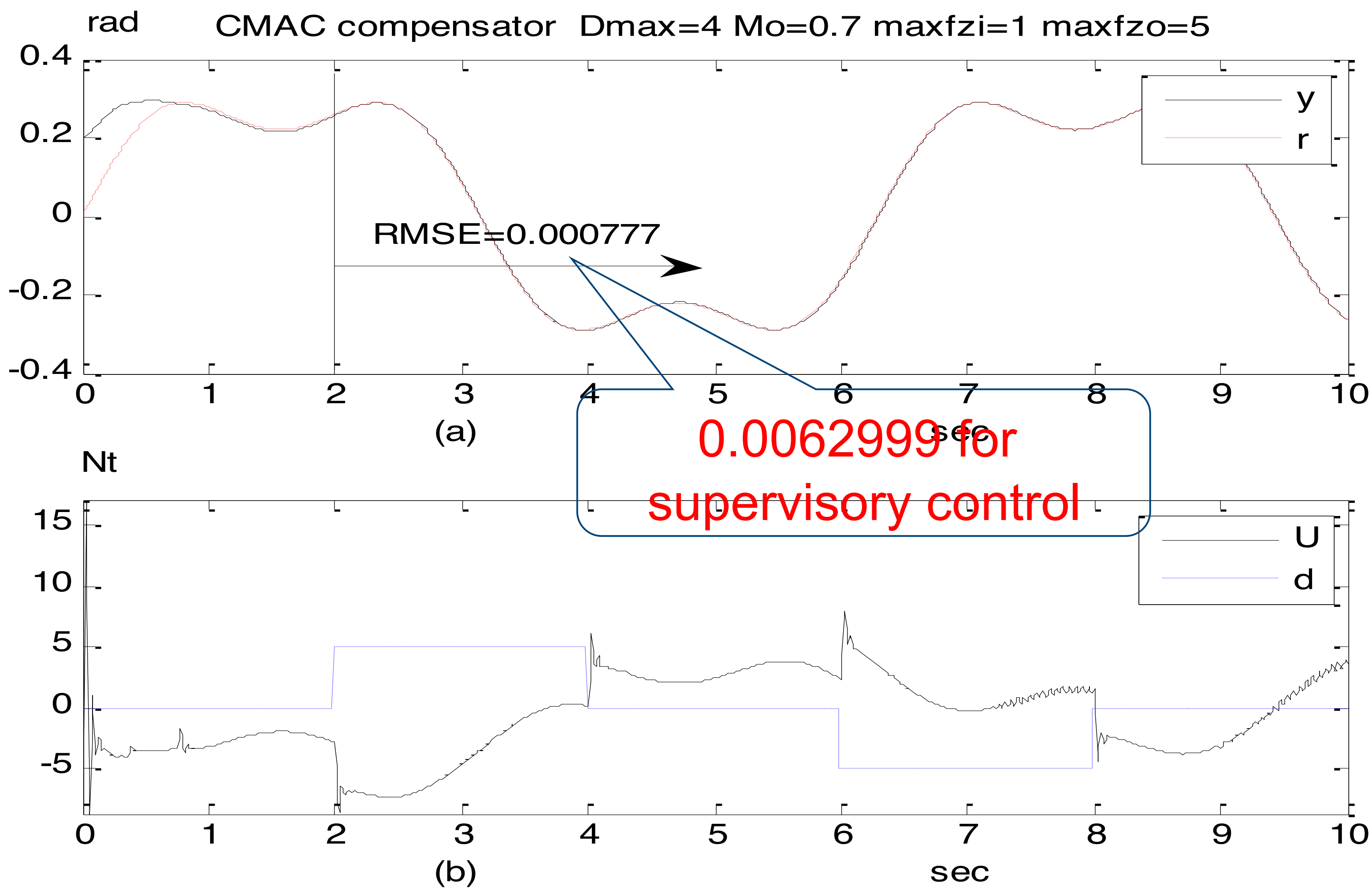
$$u = u(k-1) + M_0(e^{(n)} + \varepsilon_f^T \theta_{\Delta f}) + u_{CMAC} + u_s$$

Compensated learning for *E*.

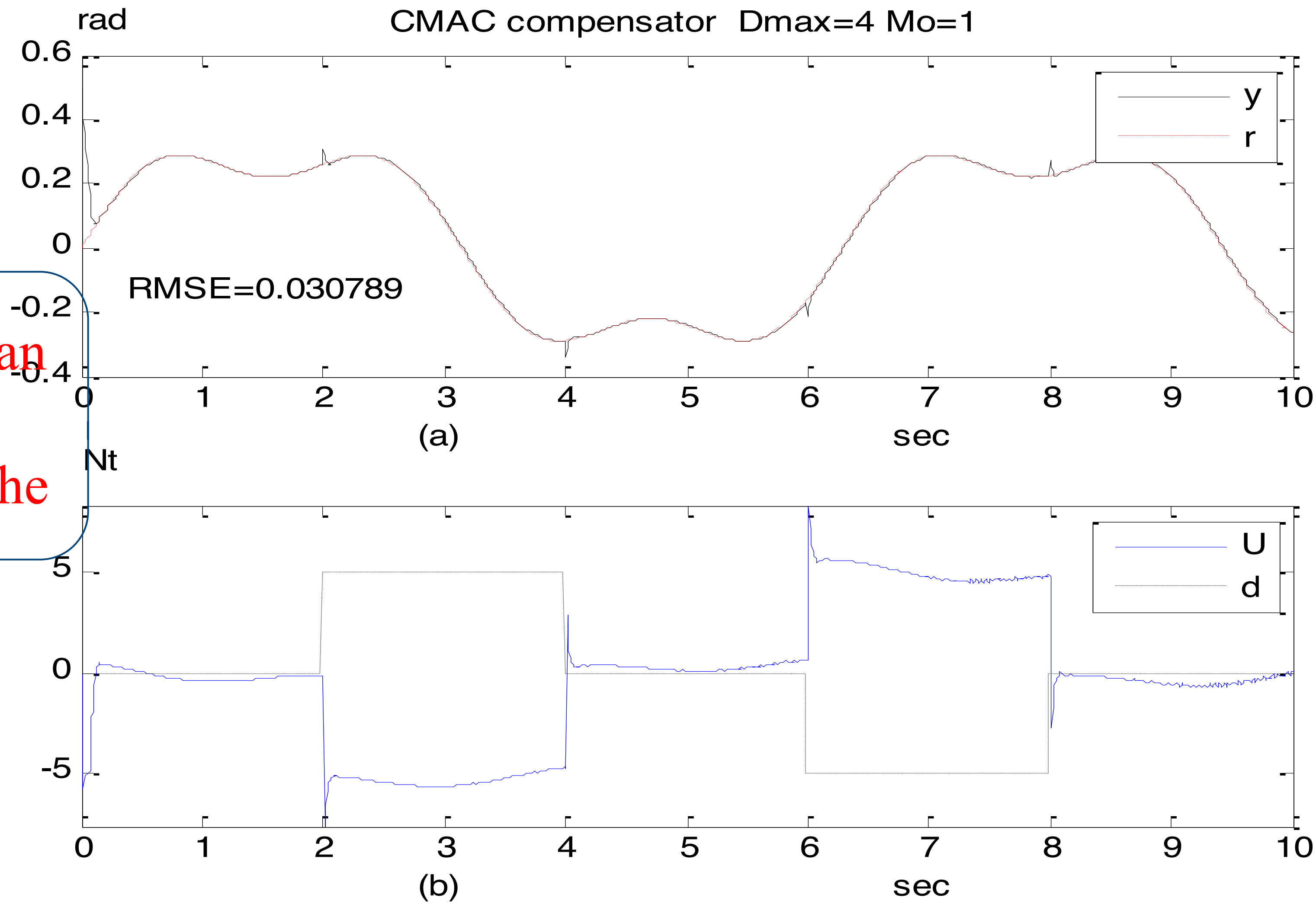
Supervisory Control



Inverted Pendulum



Exponential System



Much better than other approaches

Adaptive Fuzzy Control

There are problems in the above approaches:

- Modeling errors
- Initialization and supervisory control.
- Parameter drifting

Parameter Drifting

For adaptive fuzzy control, it can be found that the parameter is a function of errors:

$$\dot{\boldsymbol{\theta}}_f = -\beta_1 \mathbf{e}^T \mathbf{P}\mathbf{B}_1 \boldsymbol{\omega}_f$$

$$\dot{\boldsymbol{\theta}}_g = -\beta_2 \hat{u}_I \mathbf{e}^T \mathbf{P}\mathbf{B}_1 \boldsymbol{\omega}_g$$

When there are errors, the parameters will be changed. It can be expected that for tracking problems, there are always errors and the parameters are always changing.

This referred to as the **parameter drifting problem**.

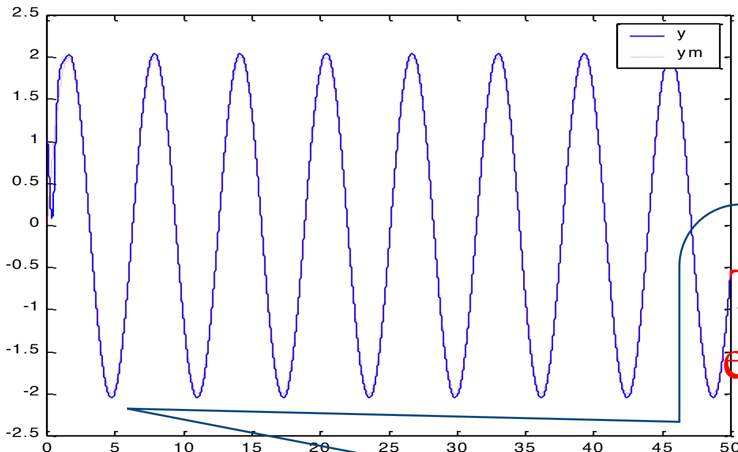
Parameter Drifting

Two situations occur:

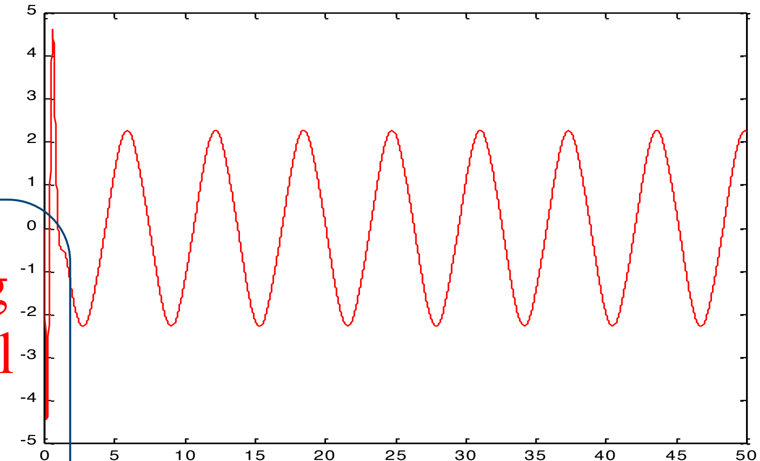
- The parameters may drift to some unwanted regions (in fact, some values may go unbounded.)
- The parameters in the optimal controller are not constants. This violates the basic assumption in the derivation of the update rules.

$\dot{\tilde{\theta}} = (\dot{\theta}^{*} - \dot{\theta}) = -\dot{\theta}$ ← no longer true!

Parameter Drifting



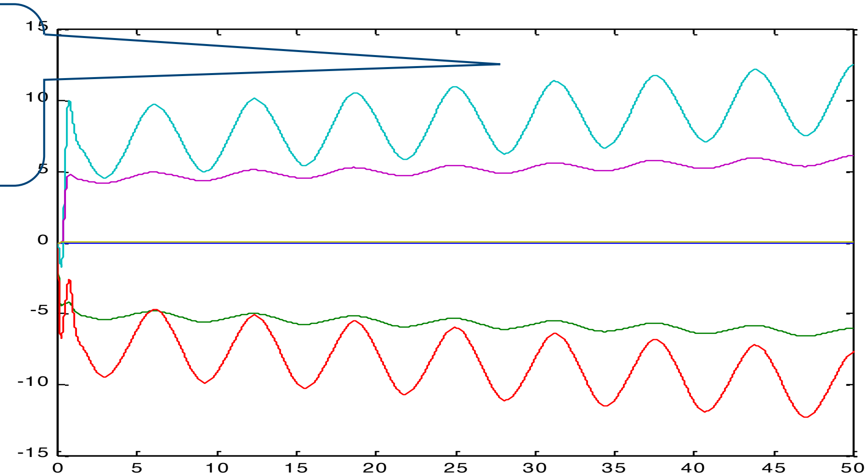
The tracking error is small enough in 5 sec.



The actual output y

The control u

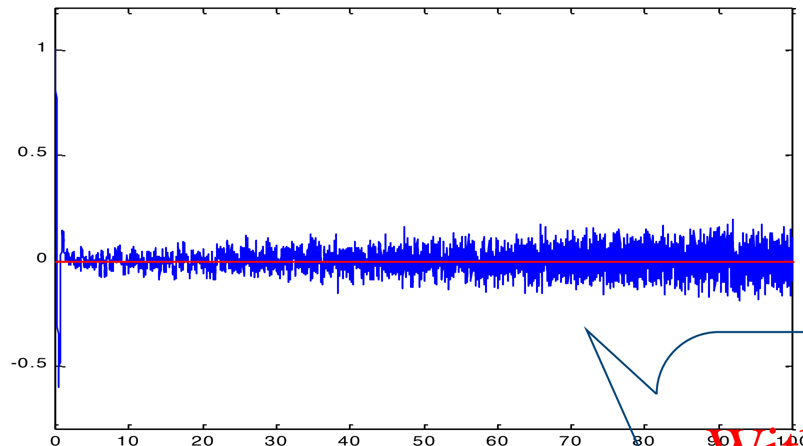
All parameters are still changing



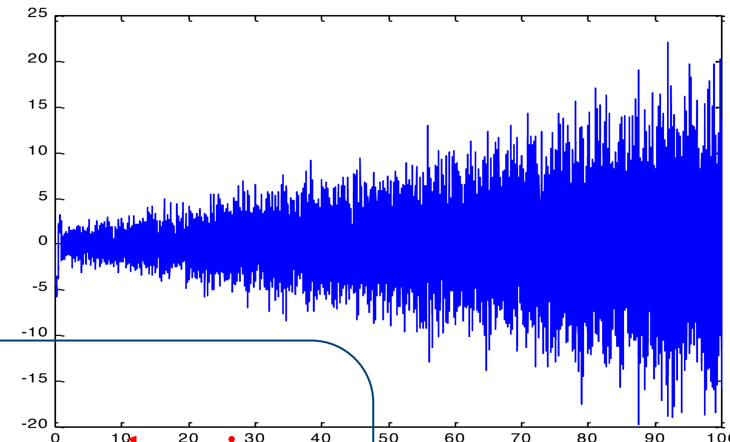
The consequence

$\underline{\theta}$

Parameter Drifting



The actual output y

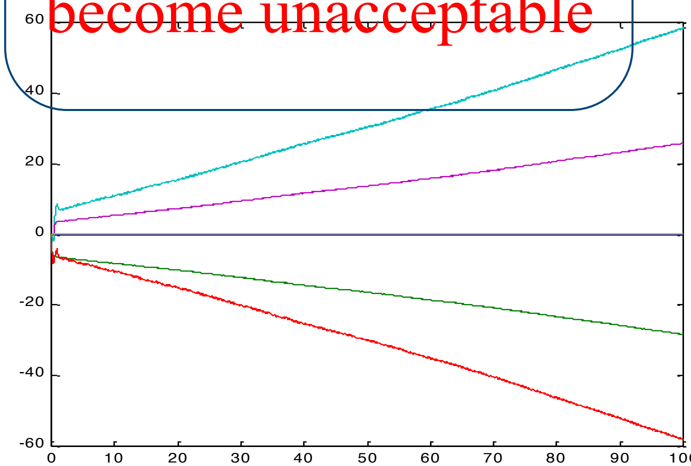


The control u

With external noise, the system may become unacceptable

The consequence

$\underline{\theta}$

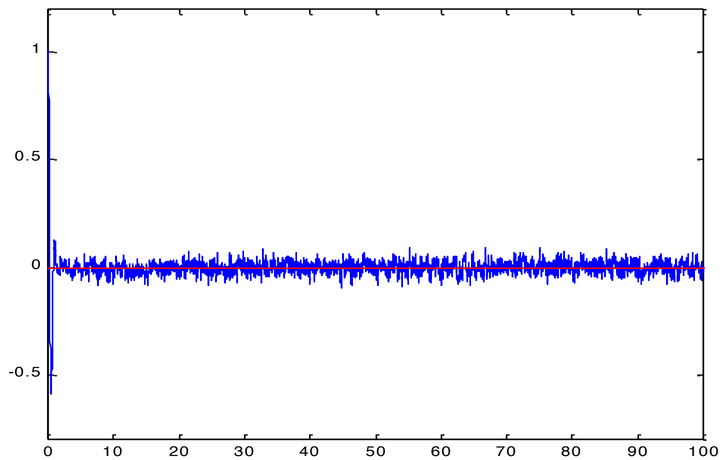


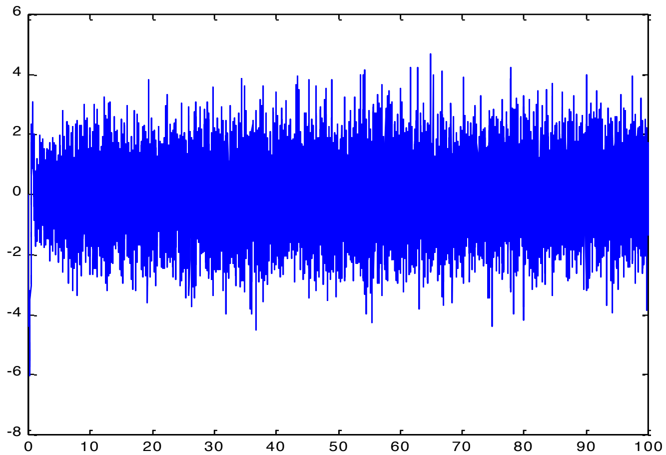
Parameter Drifting

For the unbounded phenomenon, the original adaptive fuzzy control [3] has proposed a simple way of restraining it.

- By simply clipping the bounded
- By using the projection onto the boundary surface. (Projection methods)

Clipping – Regulation control



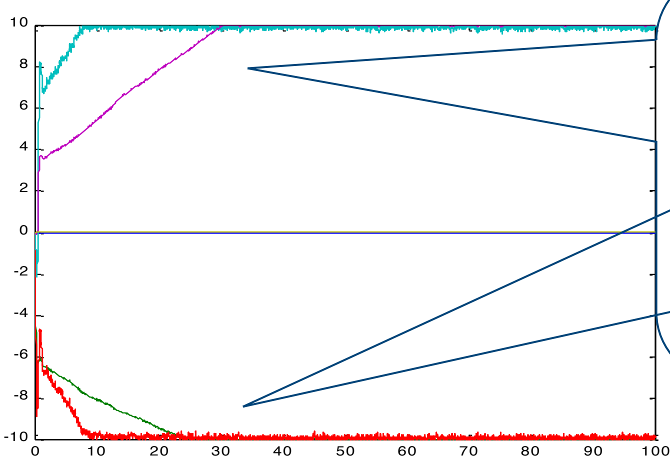


The actual output y

The control u

The consequence

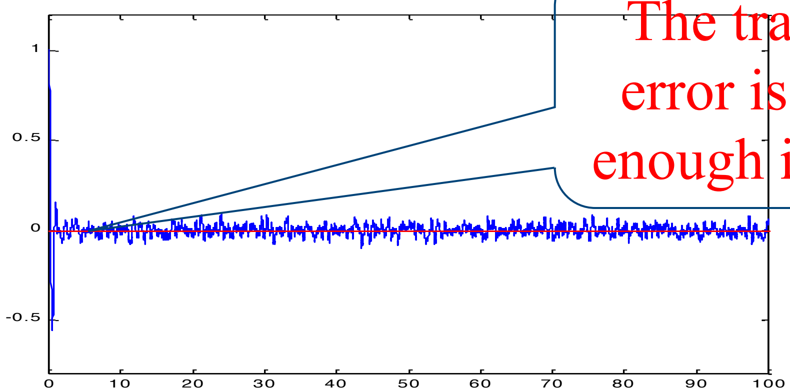
$\underline{\theta}$



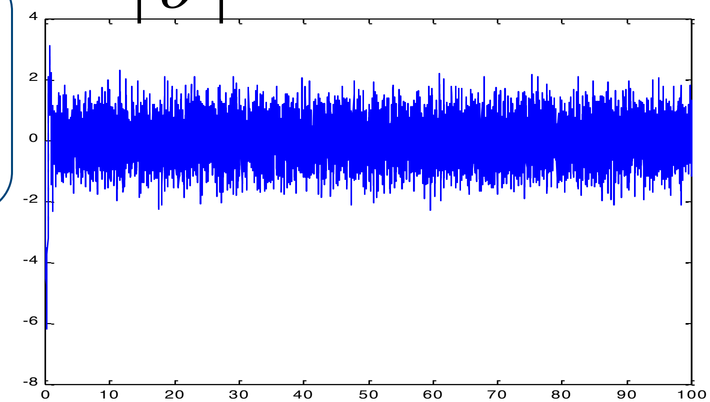
Most of the parameters go to the boundaries.
non-reasonable

Projection

The update parameters $\underline{\theta} = \underline{\theta}^* \dfrac{M_{n\theta}}{|\underline{\theta}|}$



The tracking error is small enough in 5 sec.

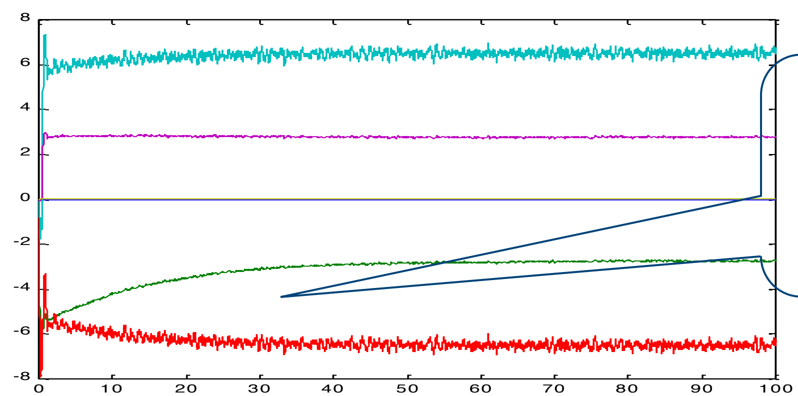


The actual output v

The control u

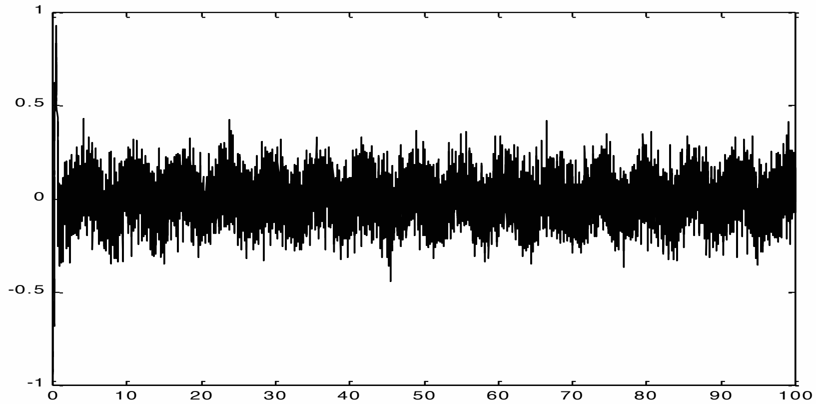
The consequenc

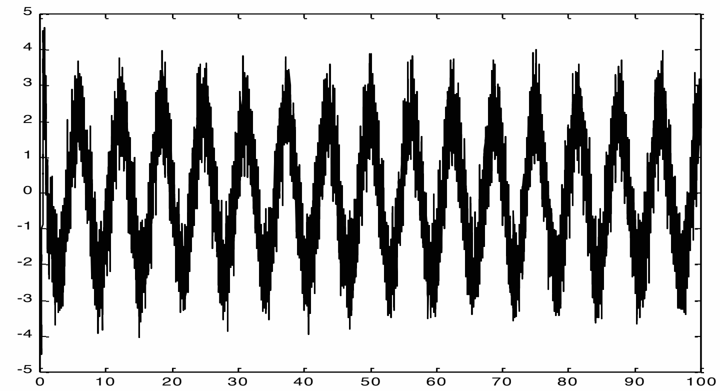
$\underline{\theta}$



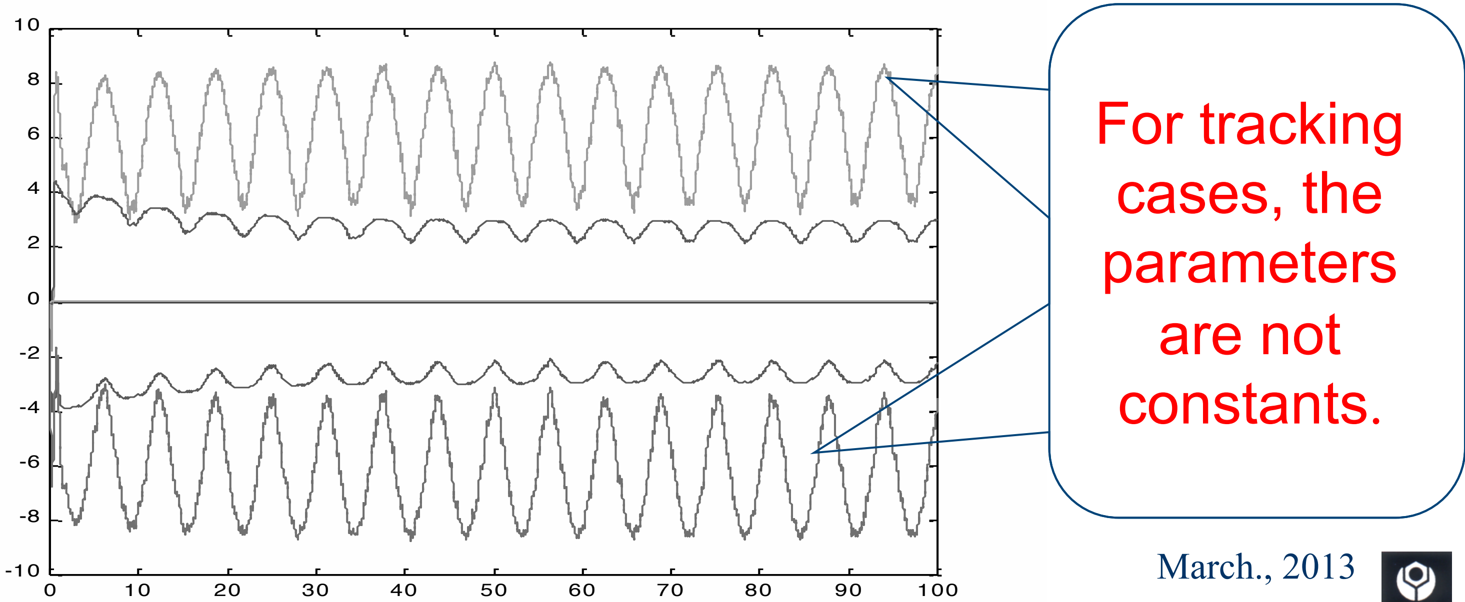
All parameters are saturated after 30 sec.

Projection– Tracking control





$\underline{\theta}$



Parameter Drifting

In above situations, it can be found that the parameters in the learned control are never constants. This violates the basic assumption in the derivation of the update rules.

Besides, it becomes an adaptive controller because the learned controller may not work well when the system stops learning.

Note that such a controller still works well, but the adaptive mechanism cannot be stopped.

Parameter Drifting

Another approach is to consider the dead-zone modification. The idea is simple. It is to stop learning under certain conditions. It is similar to the early stopping approach in neural network learning to avoid overfitting.

The problem is when to stop learning? Can the learned controller can work fairly without adaptation?

Parameter Drifting

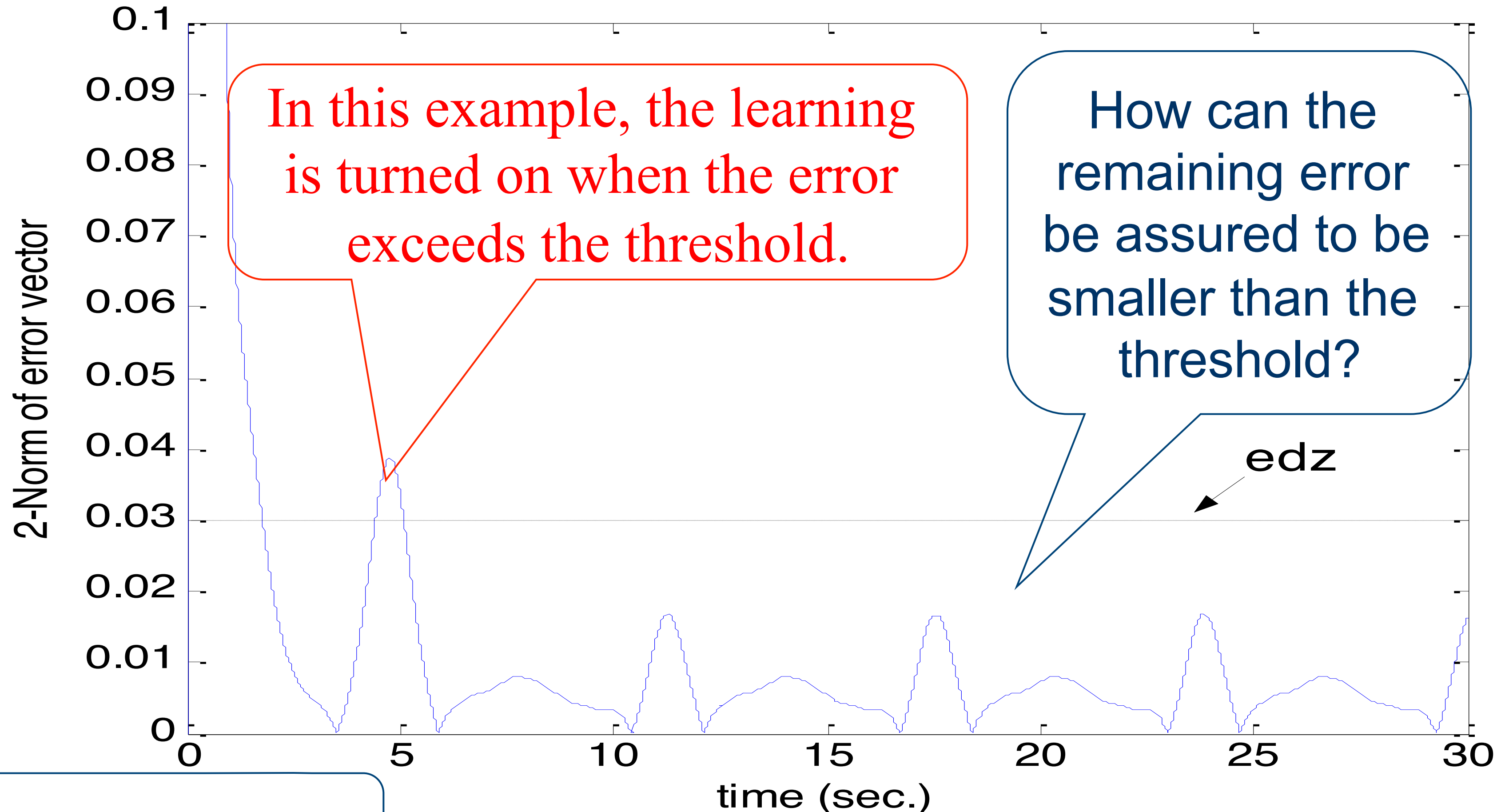
The dead-zone approach is to modify the adaptive rule as

$$\dot{\boldsymbol{\theta}}_D = \begin{cases} \alpha_m \operatorname{sgn}(g)\mathbf{e}^T \mathbf{P}\mathbf{B}_1 \boldsymbol{\omega}_D & , \quad \text{if } \|\mathbf{e}\|_2 \geq e_{dz} \\ 0 & , \quad \text{if } \|\mathbf{e}\|_2 < e_{dz} \end{cases}$$

How to select e_{dz} ?

It is desired that the error will not become larger than e_{dz} when the learning is stopped.

Parameter Drifting



An ideal case

Parameter Drifting

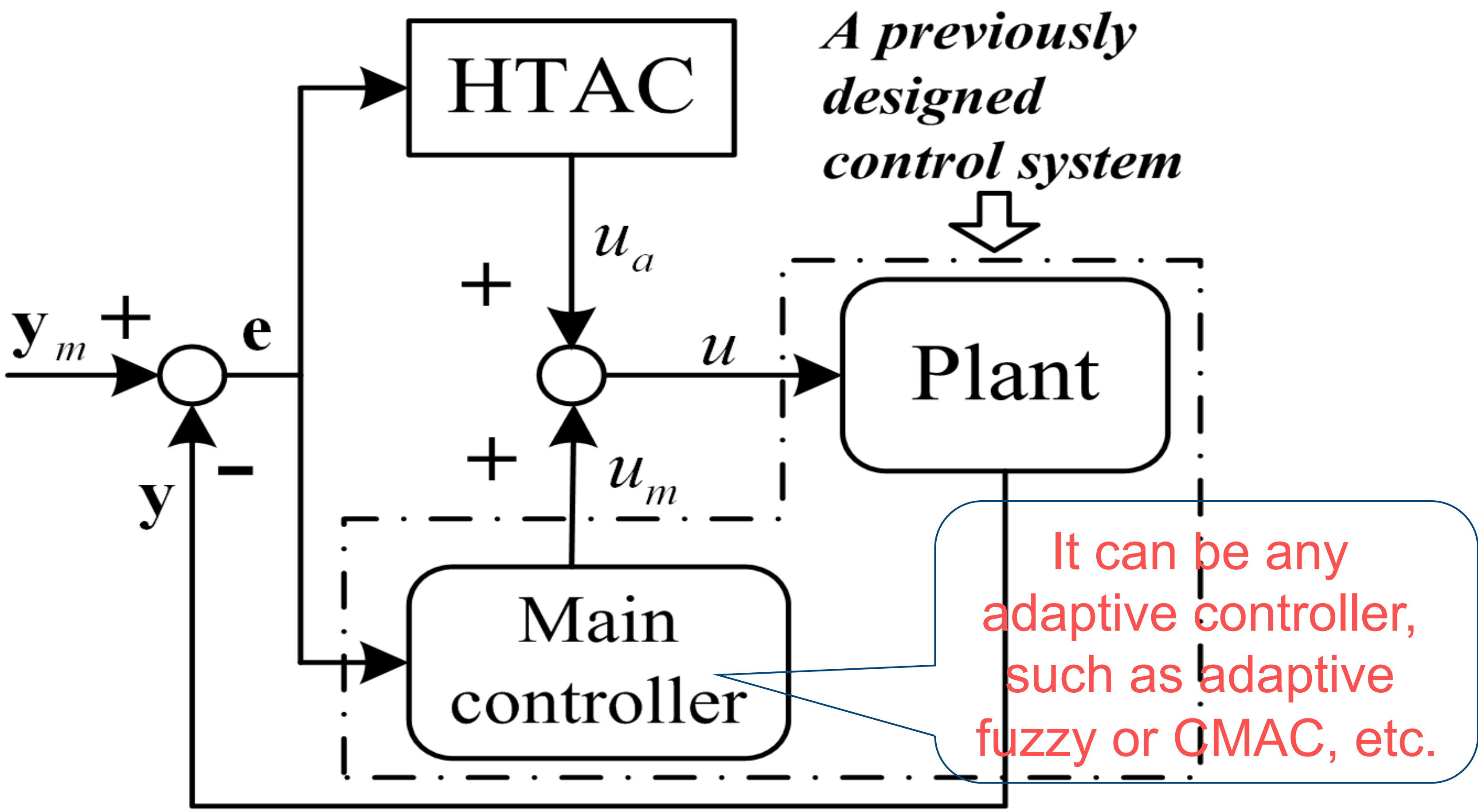
If such a learning control is desired, a robust mechanism must be employed to ensure that the error bound be restrained in the control process.

We have employed the dissipative control (HTAC) in designing the supervisory controller as:

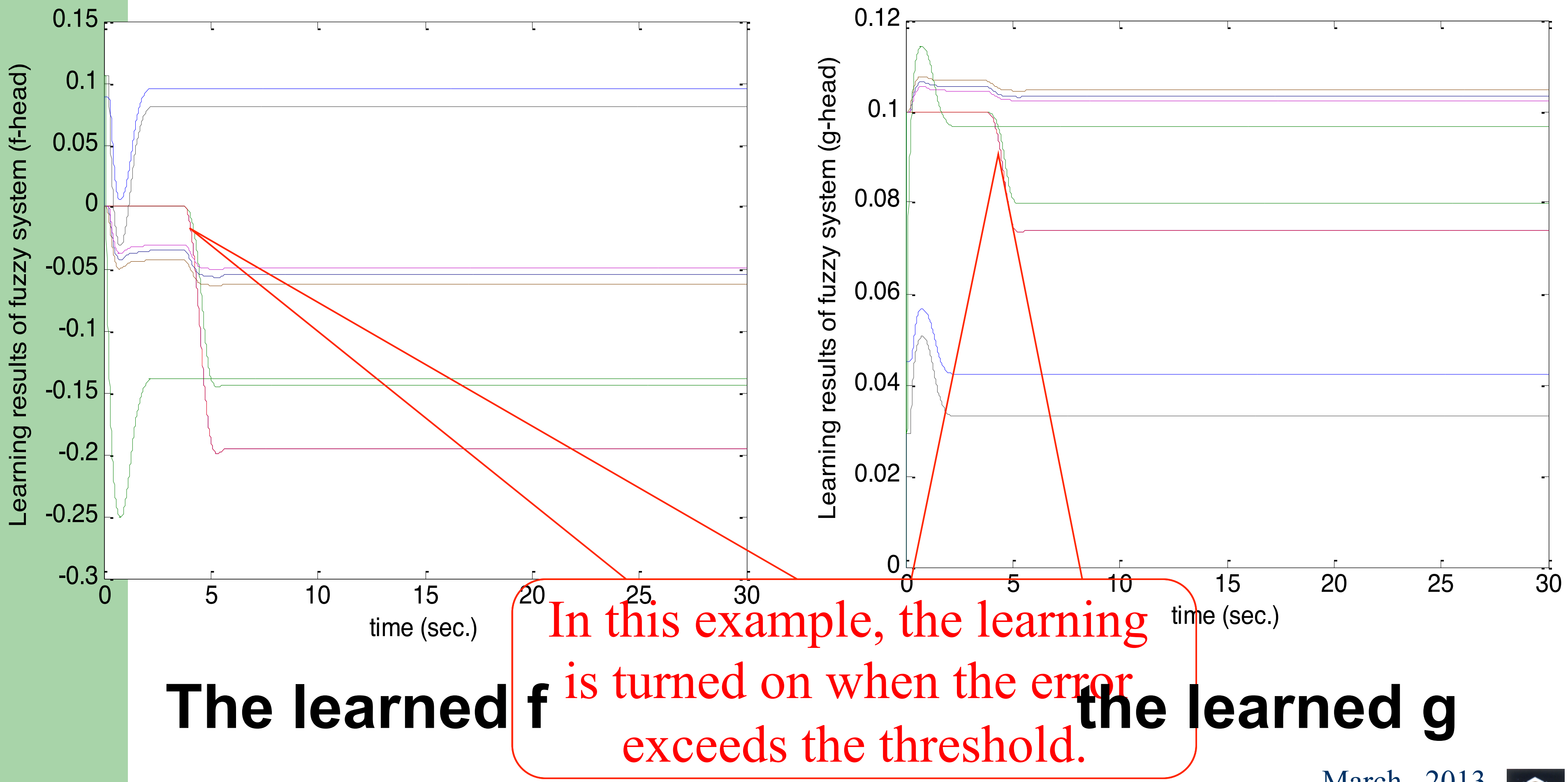
$u_a = \frac{\text{sgn}(g)}{8\sigma^2}\mathbf{e}^T\mathbf{P}\mathbf{B}_1$ with the H-infinity tracking performance having an attenuation level as

$$\delta = \left| \frac{2\sigma}{\sqrt{g_{low}}} \right| .$$

Parameter Drifting



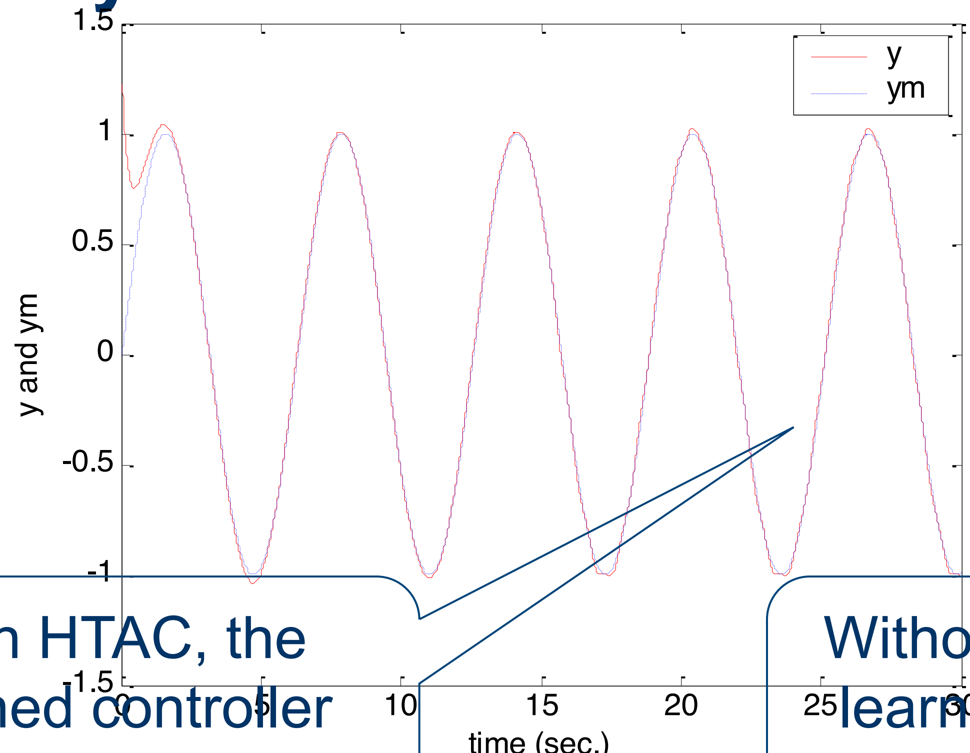
Parameter Drifting

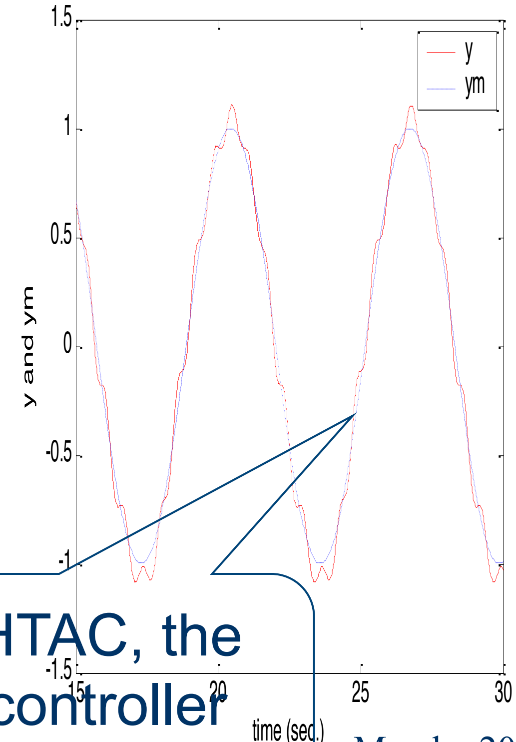


The learned f **the learned g**

Parameter Drifting

The learning is stopped after 15 seconds and an external disturbance is added into the system at the same time.





With HTAC, the learned controller can work well.

Without HTAC, the learned controller does not work well.

Conclusions

- Adaptive fuzzy control can be viewed as one learning control mechanism.
- The idea is simple and can be extended to various learning mechanisms.
- In fact, such an idea can also be employed in various learning control schemes.
- Some deficits of such an approach are discussed. If you want to use such kind of approaches, those issues must be considered in your research.

Epilogue

Those ideas are from different papers. Thus, I do not try to combine all approaches together.

Sometimes, some approaches may have similar or conflict roles. If you are interested, you may try them by yourself.

In fact, some approaches may not be complete. In other words, you may find more problems and more suitable approaches in you study .

Papers published

Thank you for your attention!

Any Questions ?!

Shun-Feng Su,

Chair Professor of Department of Electrical Engineering,

National Taiwan University of Science and Technology

E-mail: sfsu@mail.ntust.edu.tw,