Iterative Learning Control, Delays and Repetitive Control

David Owens and Jari Hätönen*
University of Sheffield
Department of Automatic Control and Systems Engineering
** EPSRC PDRF at Sheffield. Also with the University of Oulu Systems Engineering Laboratory, Finland*

Basis of the Presentation

The Presentation is based on the premise that

- *Delays* *(time and transport)*
- *Repetition* *(periodicity/reprocessing)* *and*
- *Iteration* *(improvement by repetition)*

have a place in applied Engineering Control and present there own challenges and problems to analysis and design.

They have much in common!

Why?

- Delays mean action at time t is based on “out of date” data I.e. stability and performance problems
- Repetition such as reprocessing a work-piece as in metal rolling has to cope with stability and performance implications of physical interactions between repetitions
- Iteration in the sense of repeating an action to improve the control performance is a special case of repetition.

Introduction – "Classical Control"

- Classical control theory considers the plant model (assumed discrete for simplicity) in R^n

$$\begin{cases} x(t+1) = Ax(t) + Bu(t) \\ y(t) = Cx(t) \end{cases}$$

with an initial condition $x(0) = x_0 \quad t \in [0, \infty)$

- The control design objective is to drive the state $x(t)$ to zero (regulation) or make the output of the system $y(t)$ to track a given reference $r(t)$ (tracking problem) with a feedback controller.

Introduction

- The exists several well-known techniques to solve the control design problem:
  1. PID Control (the practical approach)/compensator design (frequency domain)
  2. Classical state-feedback control, $u(t)=-Kx(t)$
  3. Optimal Control (Riccati-equation)
  4. Stochastic methods (Kalman filters)
  5. Robust Control (frequency domain)
  6. Adaptive Control
  7. Polynomial methods (pole placement)

Introduction

- Extensions to nonlinear systems:
  1. Geometric approach, i.e. Lie Algebras, Output Linearization etc.
  2. Sliding Mode Control
  3. Back-Stepping methods
  4. “Intelligent methods”: Neural networks, Fuzzy Control, Genetic Algorithms, …….

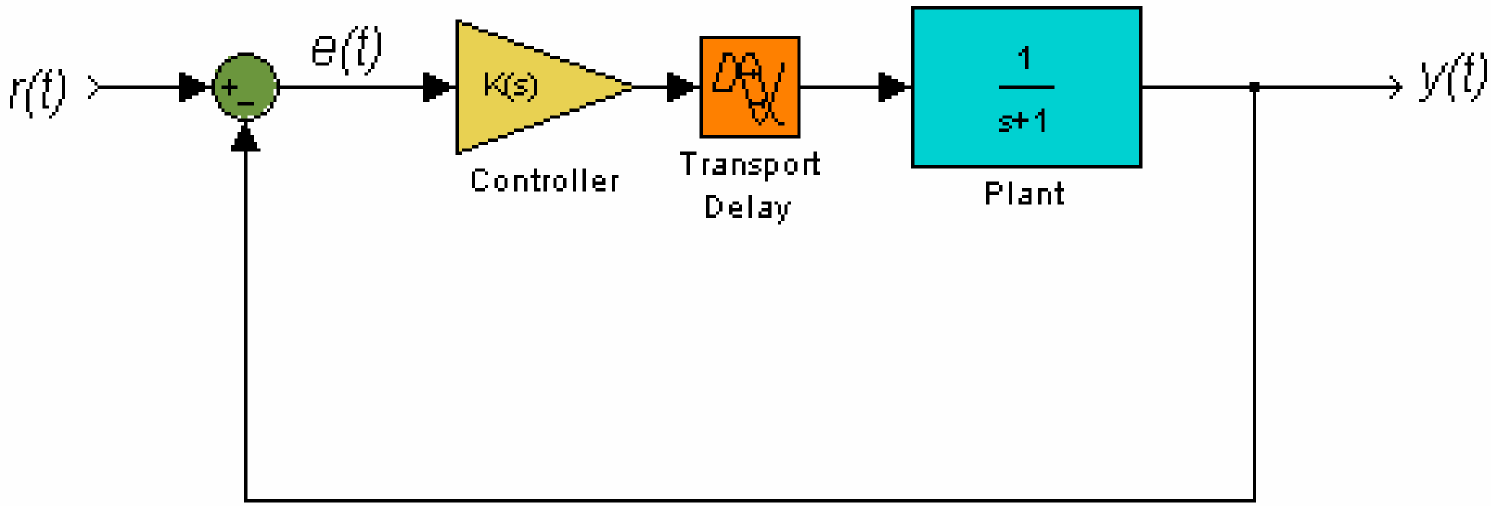
Note: The inclusion of delays makes all these theories more complex mathematically and more difficult to implement. More seriously, delays causes severe deterioration in closed loop performance.

Firstly

DELAYS ARE BAD FOR PERFORMANCE

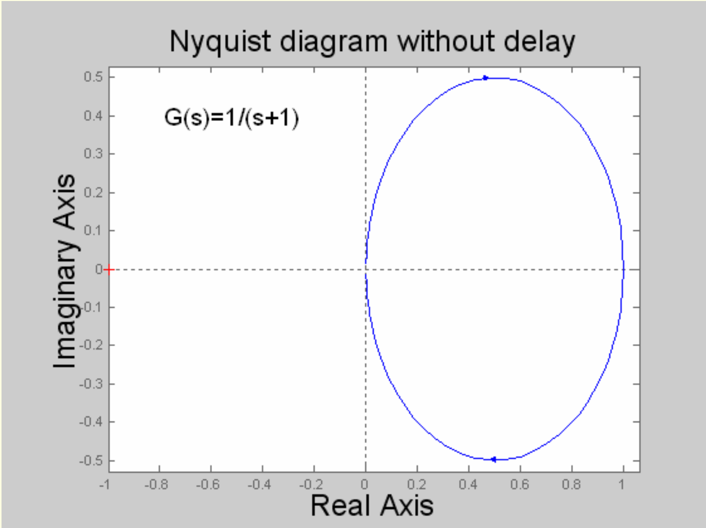
Feedback System With Delay

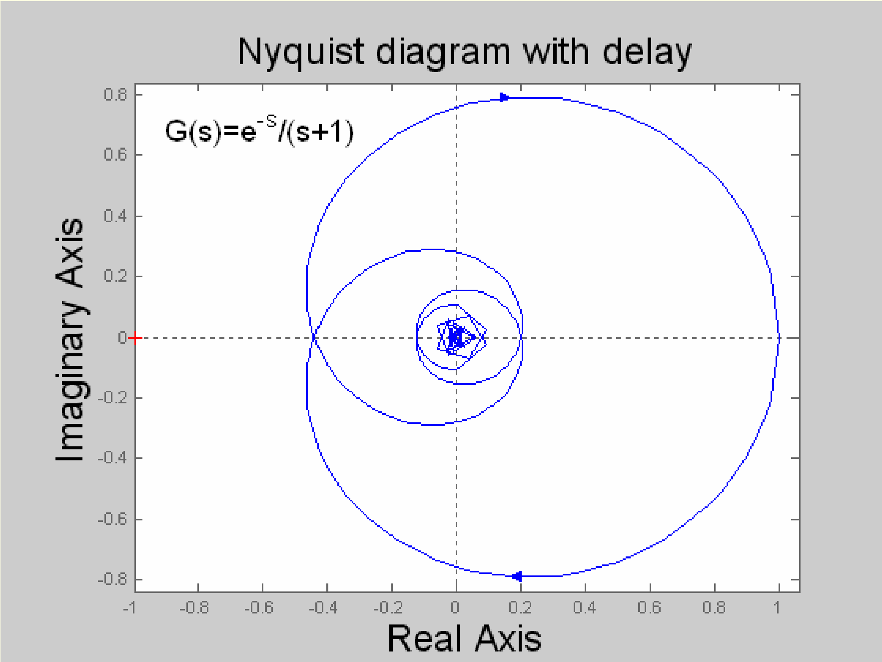
- A typical feedback system has a series delay as illustrated below



Nyquist Analysis

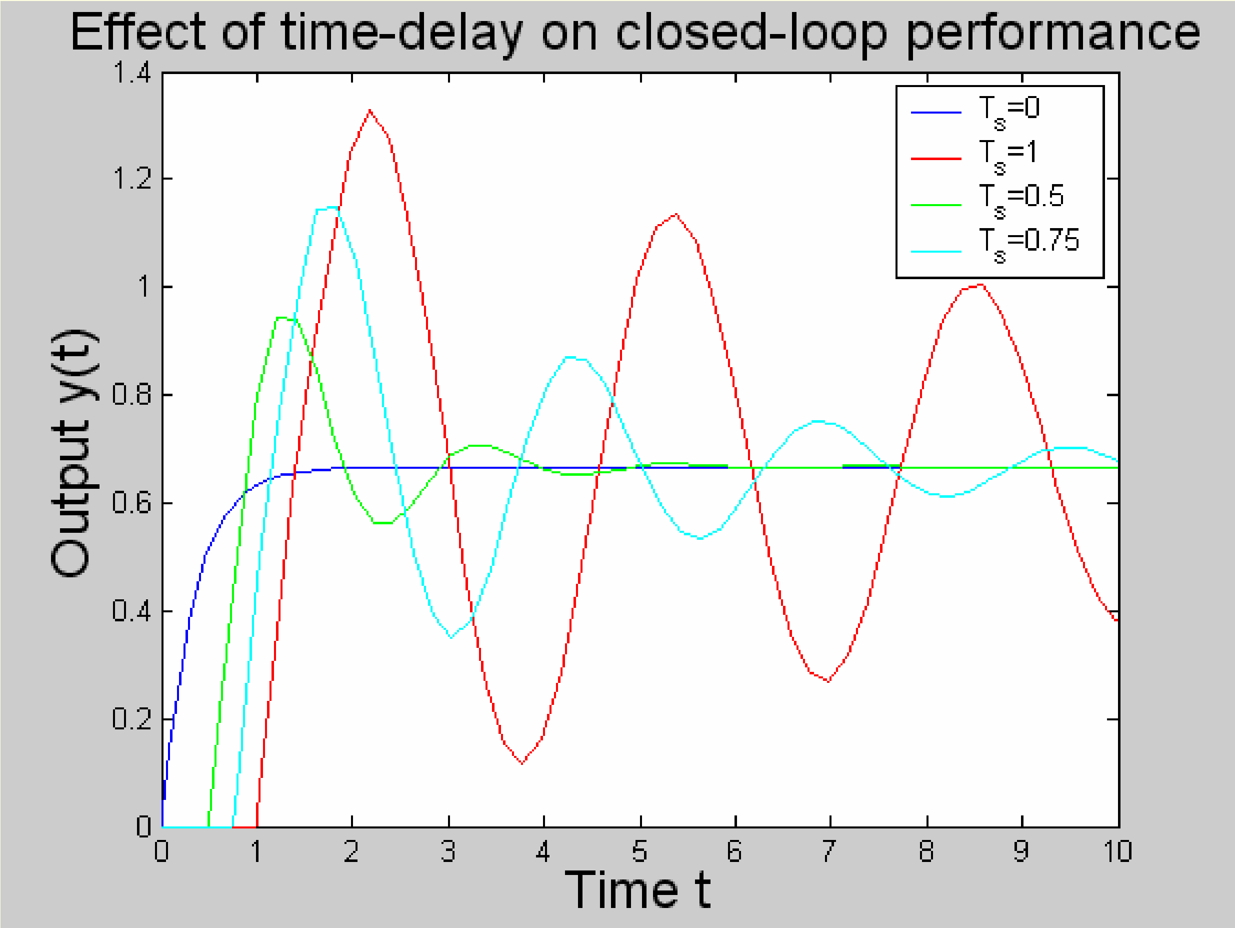
- Nyquist analysis indicates that stability is affected (dependent on both delay & gain)





Effect on Performance

Increasing the delay causes poorer performance and ultimate instability



Note that

DELAYS, REPETITION AND ITERATION ARE SIMILAR MATHEMATICALLY BUT DIFFERENT PHYSICALLY

Differences and Similarities

Delay Systems:

$$\begin{cases} dx(t)/dt = Ax(t) + Bu(t) + B_0 x(t-\tau), \quad x(0) = x_0 \\ y(t) = Cx(t) + Du(t) \end{cases}$$

Repetitive Systems:

$$\begin{cases} dx_{k+1}(t)/dt = Ax_{k+1}(t) + Bu_{k+1}(t) + B_0 x_k(t), \quad x_{k+1}(0) = f(x_k(.)) \\ y_{k+1}(t) = Cx_{k+1}(t) + Du_{k+1}(t) + D_0 x_k(t) \end{cases}$$

Iterative Systems

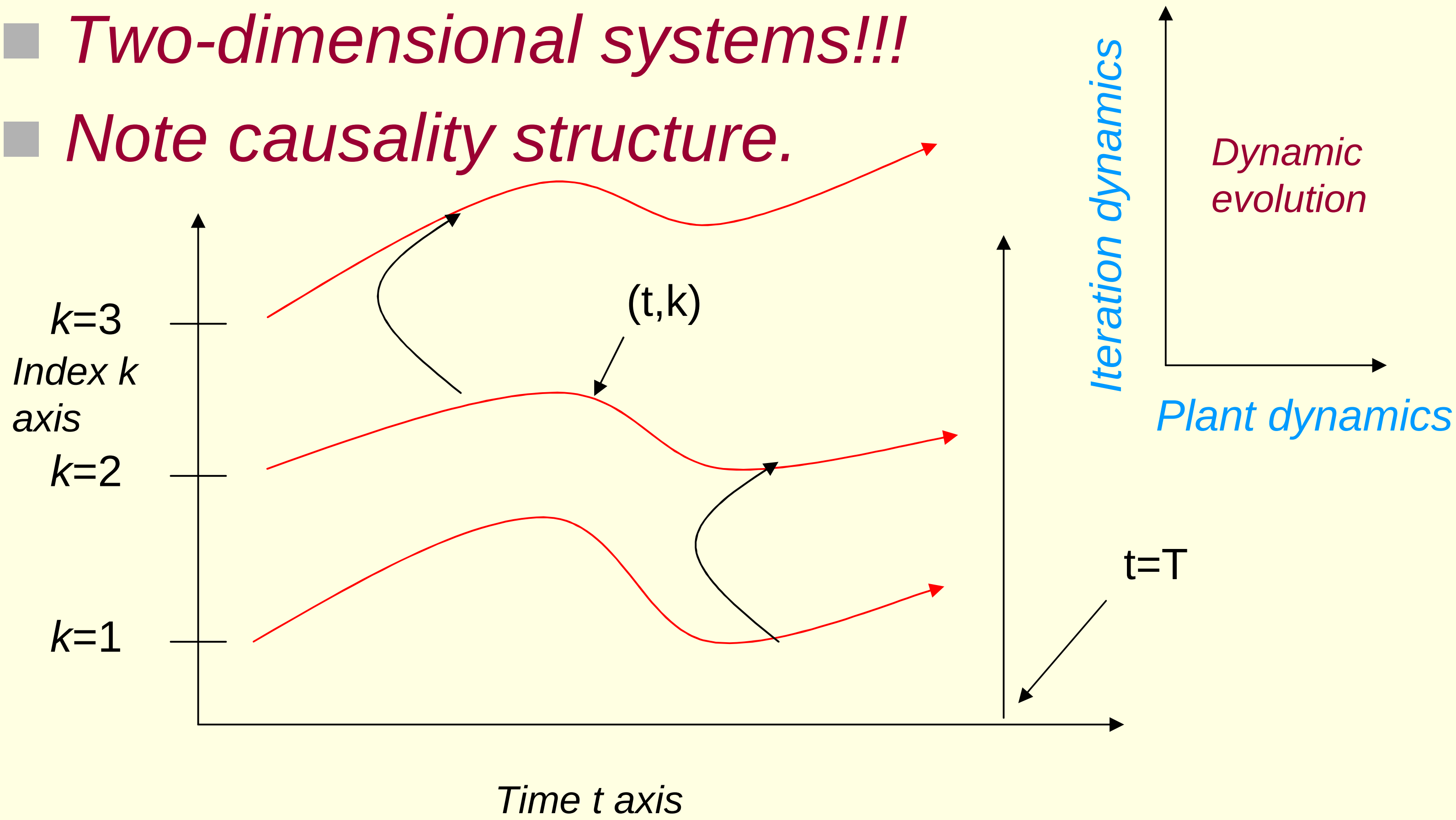
$$\begin{cases} dx_{k+1}(t)/dt = Ax_{k+1}(t) + Bu_{k+1}(t), \quad x(0) = x_0 \\ y_{k+1}(t) = Cx_{k+1}(t) + Du_{k+1}(t) \end{cases}$$

In what way are they similar?

- The delay system is repetitive with the choice of $D_0=0$ and $x_{k+1}(0)=f(x_k(.))=x_k(T)$
- The repetitive system is iterative with $B_0=0$ and $D_0=0$
- This similiarity leads to similar problems in analysis although there are some differences!

Part of the Similarity is that they are all

- *Two-dimensional systems!!!*
- *Note causality structure.*



And now

ITERATIVE LEARNING CONTROL

Iterative Learning Control – An Introduction

- Consider the following standard linear time-invariant state-space equation

$$\begin{cases} x(t+1) = Ax(t) + Bu(t), \quad x(0) = x_0 \\ y(t) = Cx(t) \end{cases}$$

defined over a *finite* time-interval $t \in [0, T]$

- The system is supposed to track a reference signal *r*(*t*) for $t \in [0, T]$ by manipulating the input variable *u*(*t*) (a classical tracking problem over a finite time-interval).

ILC-Introduction

- After the system has reached the final time point $t=T$, the state $x(T)$ is reset to x_o and the system is required to track the same reference signal $r(t)$ again.
- Real-life applications:
  1. *Robotics*
  2. *Chemical batch processing*
  3. *Start-up and shutdown of general industrial systems (for example a gas-turbine)*

ILC-Introduction

- In the past this problem was solved by picking up a fixed controller (e.g. PID-controller) and this control is applied during each repetition.

- The problem: if $u(t)$ does not give perfect tracking, then the same non-zero tracking error $e(t)$ $):=r(t)-y(t)$ is repeated during every repetition. There is no improvement!

ILC-Introduction/Simulation example

- Consider a simple illustrative plant

$$G(s) = \frac{1}{s^2 + 5s + 6}$$

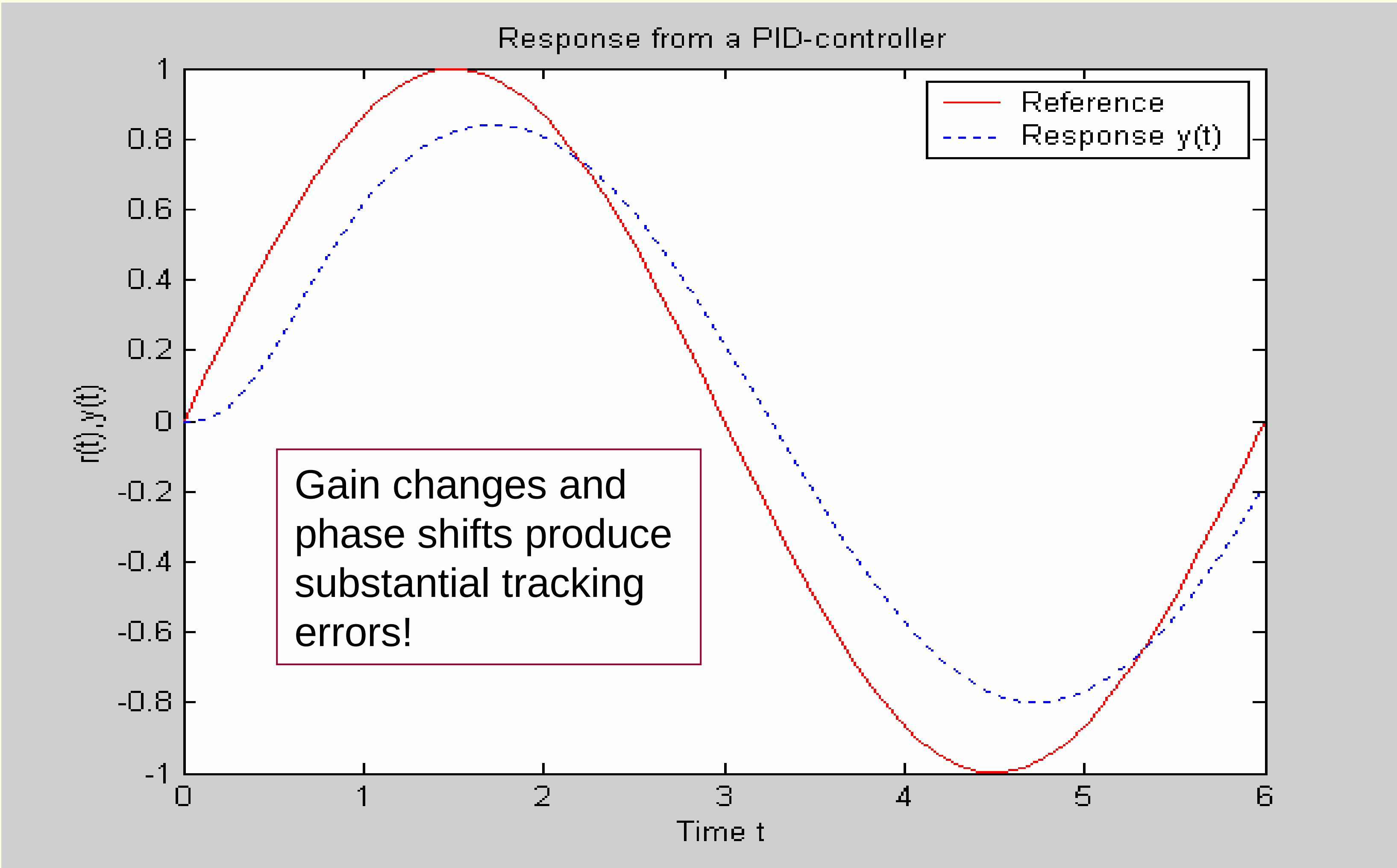
- The plant is defined over time-interval [0,6] and it is required to track a reference signal

$$r(t) = \sin(2\pi t / 6)$$

- The plant is controlled with a classical PI-controller

$$u(t) = K_p e(t) + K_I \int_0^t e(\tau) d\tau, e(t) := r(t) - y(t)$$

ILC-Introduction/Simulation example



ILC-Introduction

- In order to improve this situation, at the beginning of 1980's Japanese researchers suggested that one should use information from *previous trials* to come up with a *new input function* $u(t)$ that gives better tracking.
- Repetition makes this possible
- Repetition is the mother of learning?

Seeing the light!

ILC-General Problem definition

- Control Design is the choice of a control law

$$u_{k+1}(t) = f(e_{k+1}, e_k, \ldots, e_{k-m}, u_k, \ldots, u_{k-m})$$

so that (1) learning convergence is achieved i.e.

$$\lim_{k\to\infty} \|e_k\| \to 0 \quad \text{and} \quad \lim_{k\to\infty} \|u^* - u_k\| \to 0$$

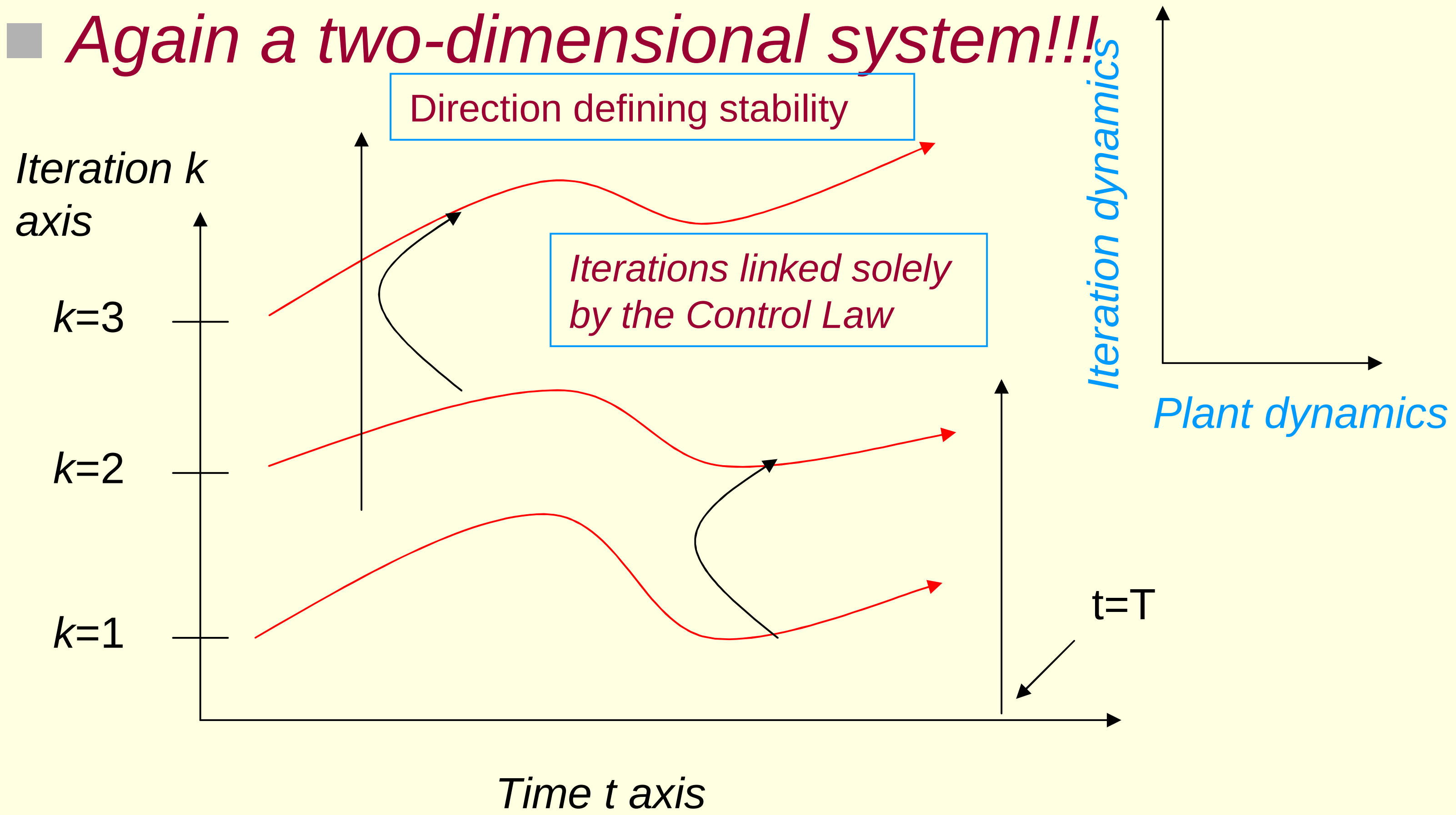
where $y_{k+1}(t) = [Gu_{k+1}](t)$, $r(t) = [Gu^*](t)$

and $e_{k+1}(t) := r(t) - y_{k+1}(t)$

(2) rate and form of convergence is acceptable!

ILC-Problem definition

- *Again a two-dimensional system!!!*



Arimoto-law: A First Attempt

- One of the first algorithms proposed for ILC was

$$u_{k+1}(t) = u_k(t) + \gamma e_k(t+1)$$

for discrete systems having a relative degree 1.

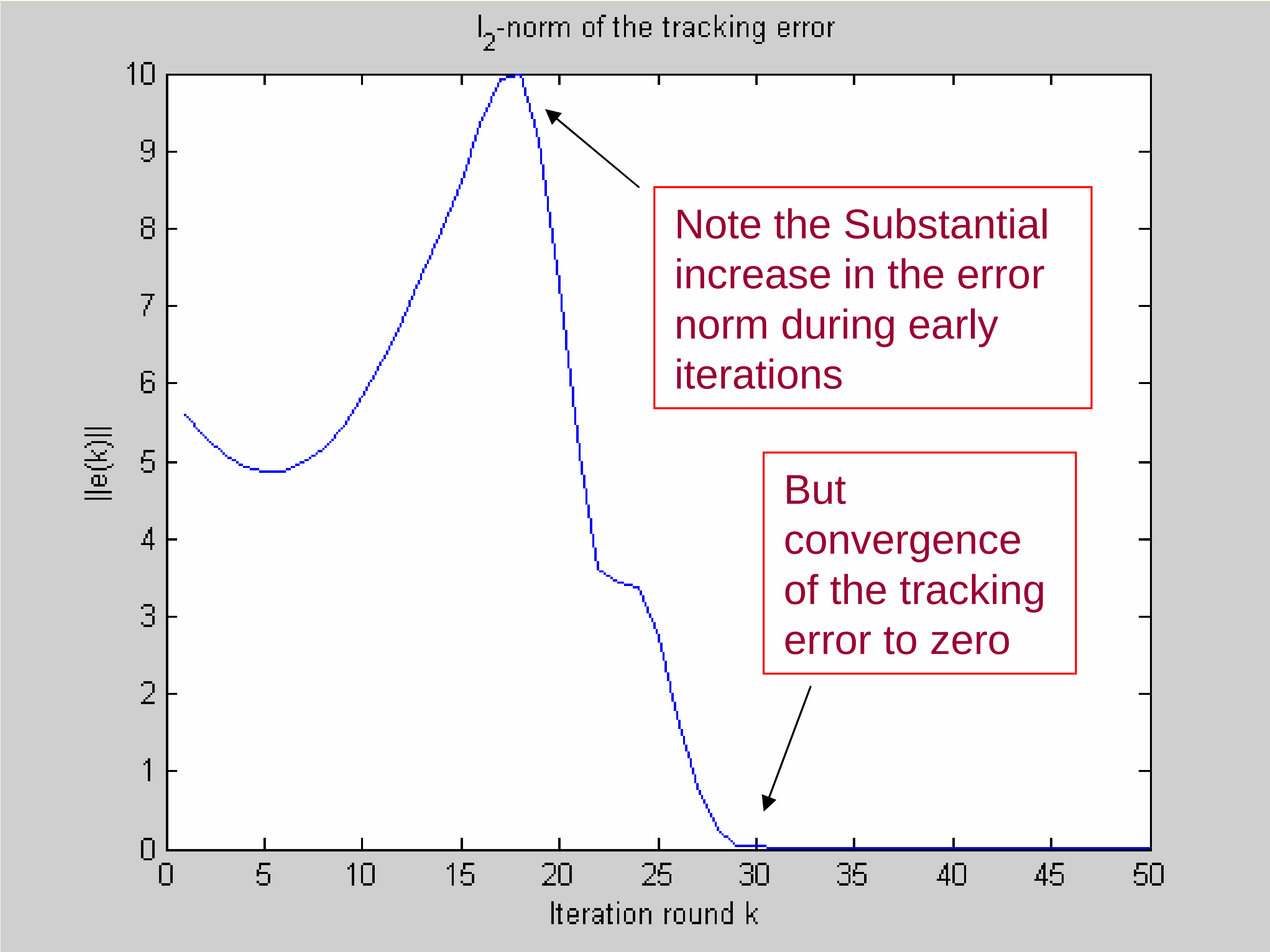
- Convergence/stability condition is

$$|1-\gamma CB| < 1 \quad \text{and} \quad r(0) = y(0)$$

- *Little information is needed about the dynamics of the plant (i.e. A matrix). Is this too good to be true? Of Course It Is!!!!!!*

Arimoto-law – Example of Poor Performance

Asymptotic convergence but poor performance!!!



Can this problem be removed?

- Is is possible to construct an algorithm with guaranteed monotonicity properties?
- That is the tracking error gets smaller each iteration …….

Norm-Optimal ILC (Amann et al)

- Idea: use quadratic optimisation in a general Hilbert-space setting by solving the Minimisation problem

$$\min_{u_{k+1}\in H_1} J(u_{k+1}) \qquad\qquad J(u_{k+1}) := \|e_{k+1}\|_{H_2}^2 + \|u_{k+1} - u_k\|_{H_1}^2$$

Subject to constraint equation defined by the model Of systems dynamics)

Note: For notational convenience, define the model via

$y_{k+1} = Gu_{k+1}$, $\quad G : H_1 \to H_2$, *G* linear and bounded

and H_1 and H_2 are suitable Hilbert-spaces

Norm-Optimal ILC (NOILC)

- Abstract solution of the optimisation problem is given by

$$u_{k+1} = u_k + G^* e_{k+1}$$

where $G^* : H_2 \rightarrow H_1$ is the *adjoint* operator of *G*

- This gives formal error evolution equation

$$e_{k+1} = (I + GG^*)^{-1} e_k$$

DOES THIS IMPLY MONOTONIC ERROR CONVERGENCE TO ZERO?

NOILC – Convergence Rates

- Suppose now that (which true in the d.t. LTI case)

$$< v, GG^* v > \geq \sigma \|v\|^2 \quad \forall v \in H_2, \sigma > 0$$

where $< \cdot, \cdot >$ is the inner product in H_2

then $$\boxed{\|e_{k+1}\| \leq \frac{1}{1+\sigma} \|e_k\| < \| e_k \|}$$

i.e. Convergence is monotonic and geometric to zero!!!!

- Note: In the continuous-time LTI case the result is *almost geometric monotonic convergence*

NOILC/Implementation Issues

- In the dynamical system context the control law

  $$u_{k+1} = u_k + G^* e_{k+1}$$

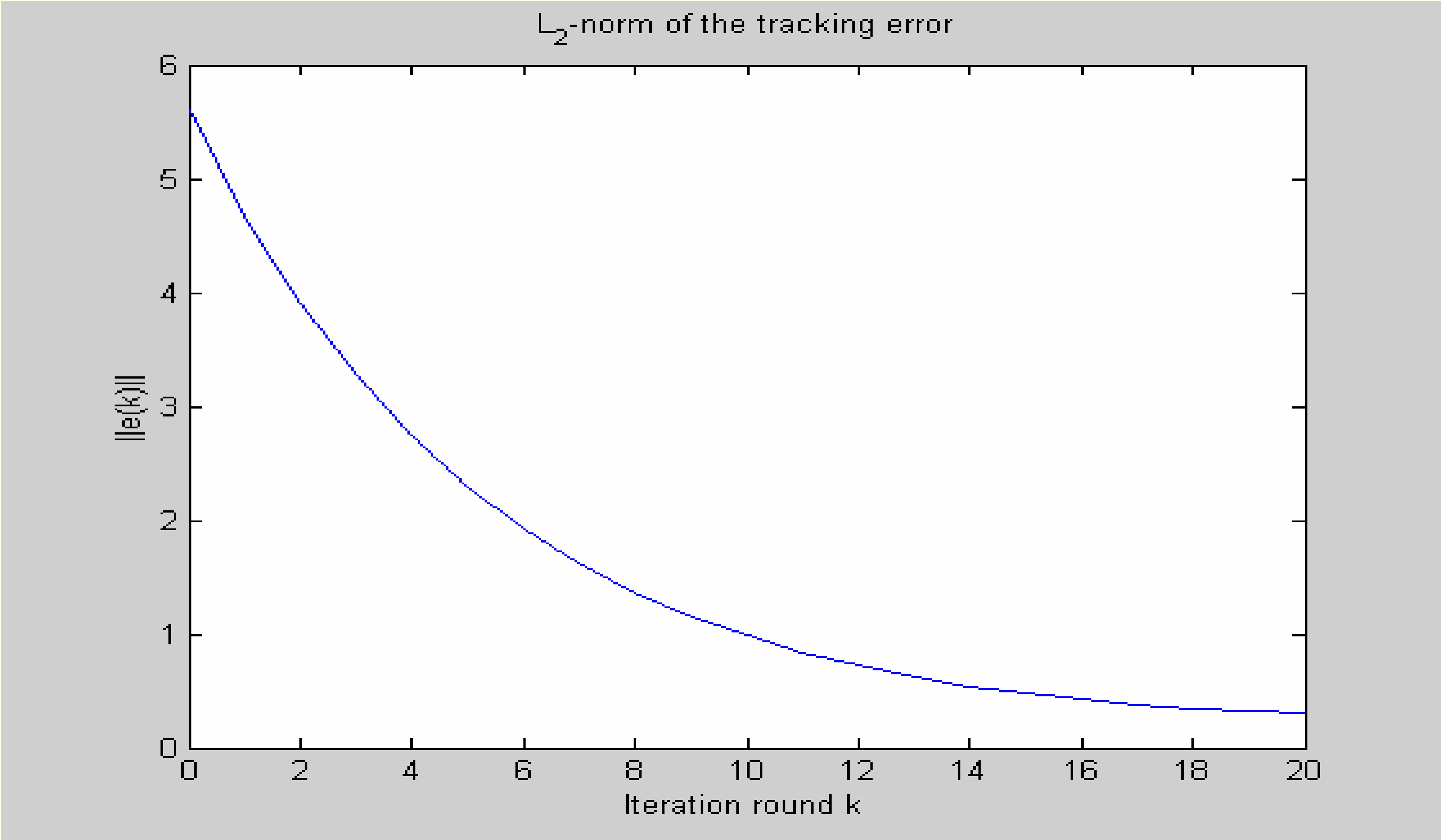
  is non-causal, and cannot be used directly in practice

- Fortunately it can be shown that an equivalent causal representation exists for LTI state space systems of the general form

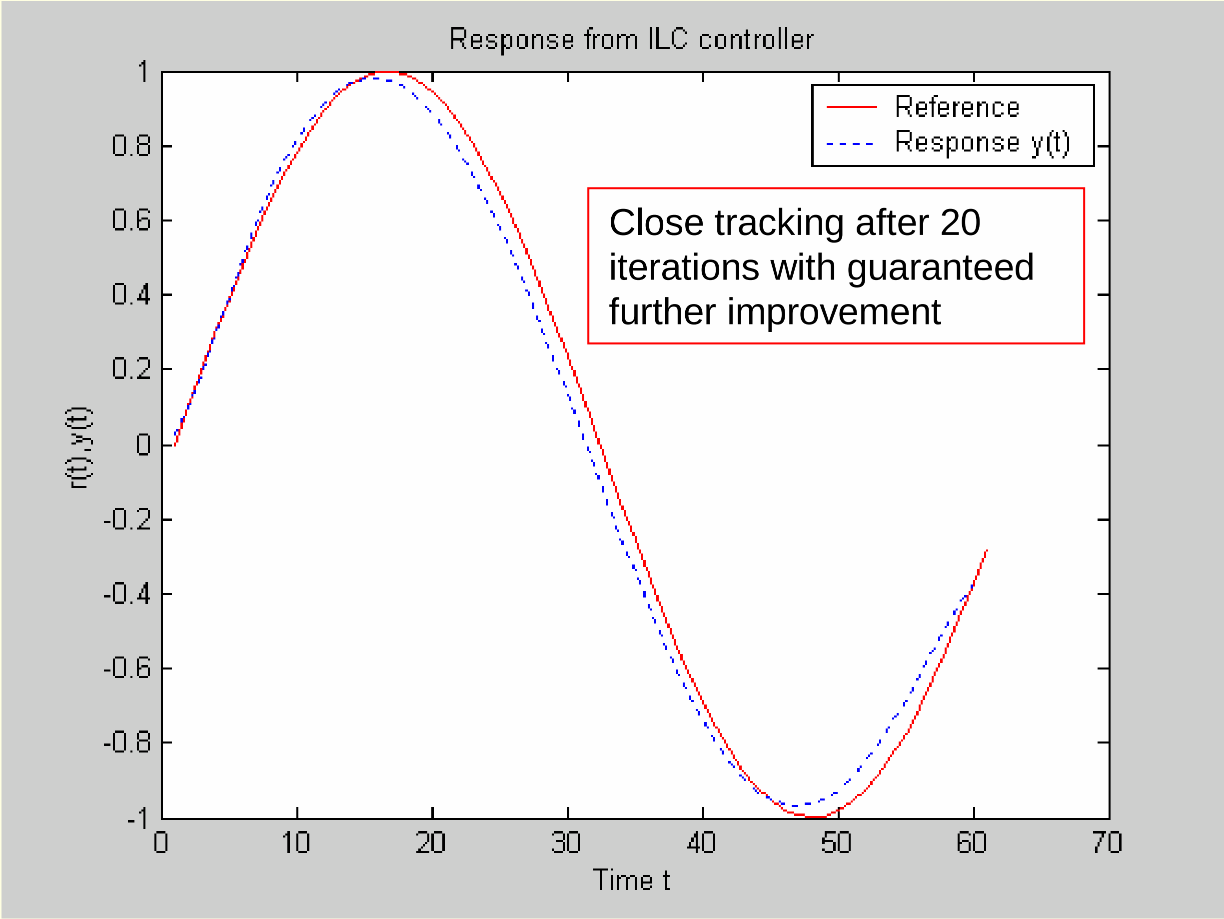
  $$u_{k+1}(t) = u_k(t) - B^T \left[K(t)(x_{k+1}(t) - x_k(t)) - \xi_{k+1}(t) \right]$$

- $K(t)$ is a solution of a Riccati equation and $\xi_{k+1}(t)$ is a “predictive term” that has to be computed between trials from past error and input data

Norm-Optimal ILC/Simulation example - revisited



Norm-Optimal ILC/Simulation example



Norm-Optimal ILC/Further results

- Improved Convergence rates can be obtained by appropriate choice of weight matrices in the performance criterion
- A predictive formulation also adds convergence benefits using the following criterion and a receding horizon principle

$$J(u_{k+1},\lambda)=\sum_{i=1}^{N}\lambda^{i-1}\left(\left\|e_{k+i}\right\|^{2}+\left\|u_{k+i}-u_{k+i-1}\right\|^{2}\right)$$

- Gives considerably faster convergence with monotonic convergence as fast as λ^{-k} if the "horizon" N is large
- However, implementation is far more complex
- The ideas do however show the potential implicit in the paradigm
- Parameter Optimization Approaches offer the potential for good performance with simpler computations ……

Parameter Optimal ILC (POILC)

- How to select *γ in the Arimoto algorithm*?

$$u_{k+1}(t) = u_k(t) + \gamma e_k(t+1)$$

- Let the gain γ vary from iteration to iteration and minimize the criterion

$$J(\gamma_{k+1}) = \|e_{k+1}\|_2^2 + w\gamma_{k+1}^2; e_{k+1} := r - G_e u_{k+1}$$

- The solution of this optimisation problem is $\gamma_{k+1} = \dfrac{e_k^T G_e e_k}{w + e_k^T G_e^T G_e e_k}$

The resultant algorithm is monotonically convergent to zero tracking error if the plant is positive-real

NOILC/Current Applications Studies

- Currently a project between Soton and Sheffield to further develop the theory of ILC but also to undertake a serious application of Norm-Optimal ILC on a conveyor-belt “robot” in food industry.
- The goal is to be able to fill in two tuna cans in one second – a massive improvement in production rate with a computational technique (low investment – massive improvement in production rate?)
- New applications are also being sought from Finnish and other European (incl. UK) companies.

And now ….

REPETITIVE CONTROL

RC - Introduction

- Consider the following standard linear time-invariant state-space equation

$$\begin{cases} x(t+1) = Ax(t) + Bu(t), \quad x(0) = x_0 \\ y(t) = Cx(t) \end{cases}$$

defined over a time-interval $t \in [0, \infty)$

- The system is supposed to track a reference signal a T-periodic reference signal $r(t)$ (i.e. $r(t)=r(t+T)$) by manipulating the input variable $u(t)$.
- Note that the only information available is periodicity, but the actual shape of $r(t)$ is arbitrary.

RC-Internal Model Principle

- Let the control law be

$$[Mu](t) = [Ne](t)$$

- The internal model principle says that the operator *M* has to include a model *P* of the reference signal where [*P*r](t)=0
- Because the reference signal is T-periodic, the internal model is P=1-z^{-T}, where z^{-1} is the standard backward-shift operator, i.e.

$$[(1 - z^{-T})r](t) = r(t) - r(t - T) = r(t) - r(t) = 0$$

RC/A polynomial approach

- Let the process model be

$$A(z^{-1})y(t) = B(z^{-1})u(t)$$

- Using the internal model $1\text{-}z^{-T}$ the process model can be written as

$$\widetilde{A}(z^{-1})e(t) = B(z^{-1})\Delta u(t)$$

where

$$\begin{cases} \widetilde{A}(z^{-1}) := -(1 - z^{-T})A(z^{-1}) \\ \Delta u(k) = u(t) - u(t-T) \end{cases}$$

RC/A polynomial approach

- The model has an equivalent state-space representation

$$\begin{cases} x_m(t+1) = Ax_m(t) + Bu(t), \quad x(0) = x_0 \\ e(t) = C_m x(t) \end{cases}$$

- This a standard tracking problem, can be solved by a lot of different techniques (pole-placement, adaptive control robust control)
- Receding Horizon Optimal control, solve

$$\begin{cases} \min\limits_{\Delta u \in l_2} J(\Delta u(k), x_m(t)) \\ J(\Delta u(k), x_m(t)) = \sum\limits_{i=t}^{\infty} [e(i)^T Q e(i) + \Delta u(i)^T R \Delta u(i)] \end{cases}$$

RC/A polynomial approach

- The optimal control law is Riccati feedback

$$u(t) = u(t-T) + Kx_m(t)$$

Compare with NOILC

$$u_{k+1} = u_k + G^* e_{k+1}$$

- Furthermore, the state $x_m(t)$ can be estimated with a state-observer (Kalman-filtering approach -> increases robustness
- Note also that the dimension is $n+T$, where n is the dimension of the plant and T is the number of time-points inside a period = high-order control.
- The approach works also for T-periodic load disturbances and multi-periodic signals.

RC/Simulation example

- Consider again the process model

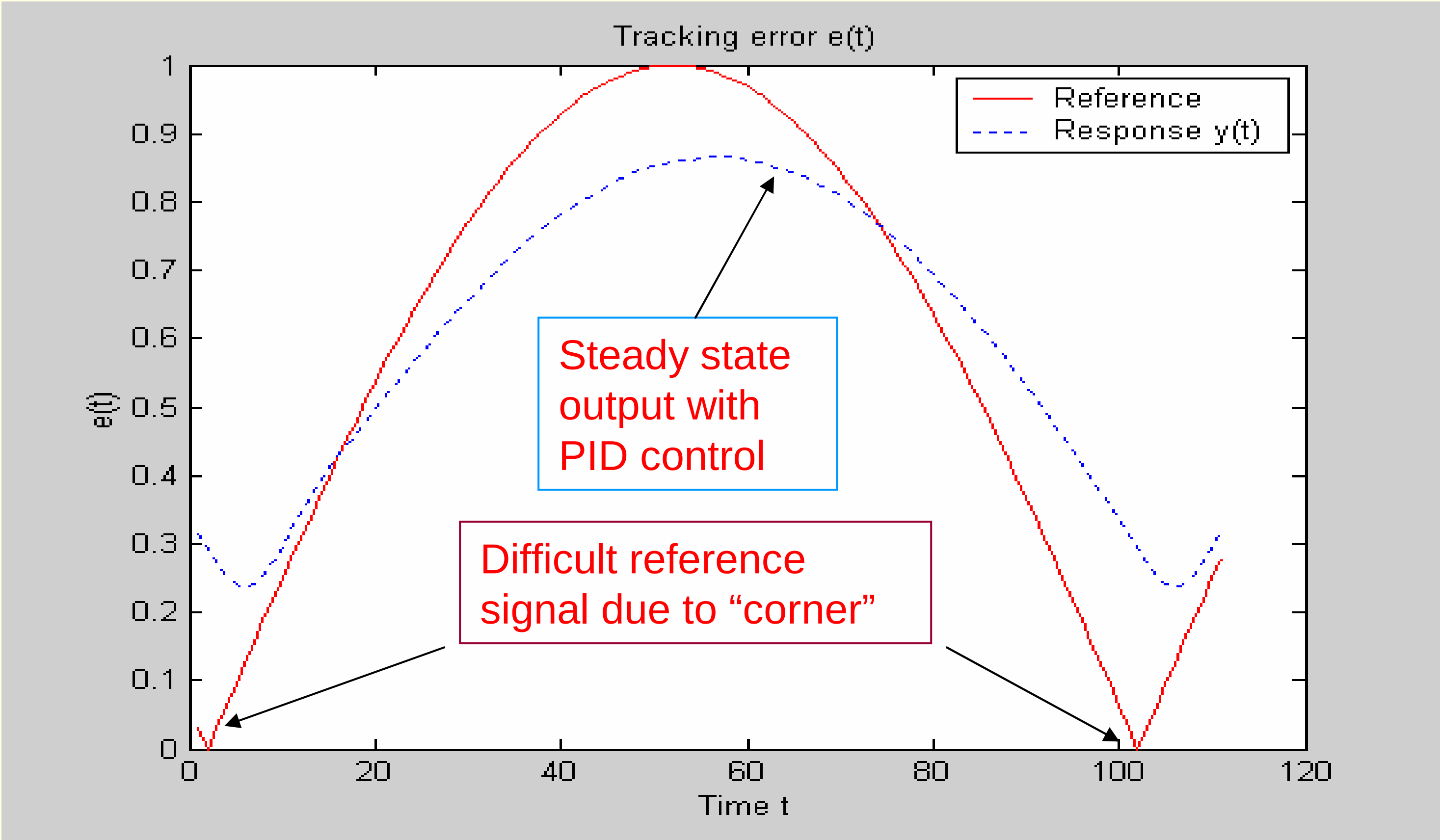
$$G(s) = \frac{1}{s^2 + 5s + 6}$$

and the plant is supposed to track a reference signal $r(t)$ where $r(\mathrm{t})=r(t+10)$.

- The process is sampled with sampling time T_s=0.1 seconds.

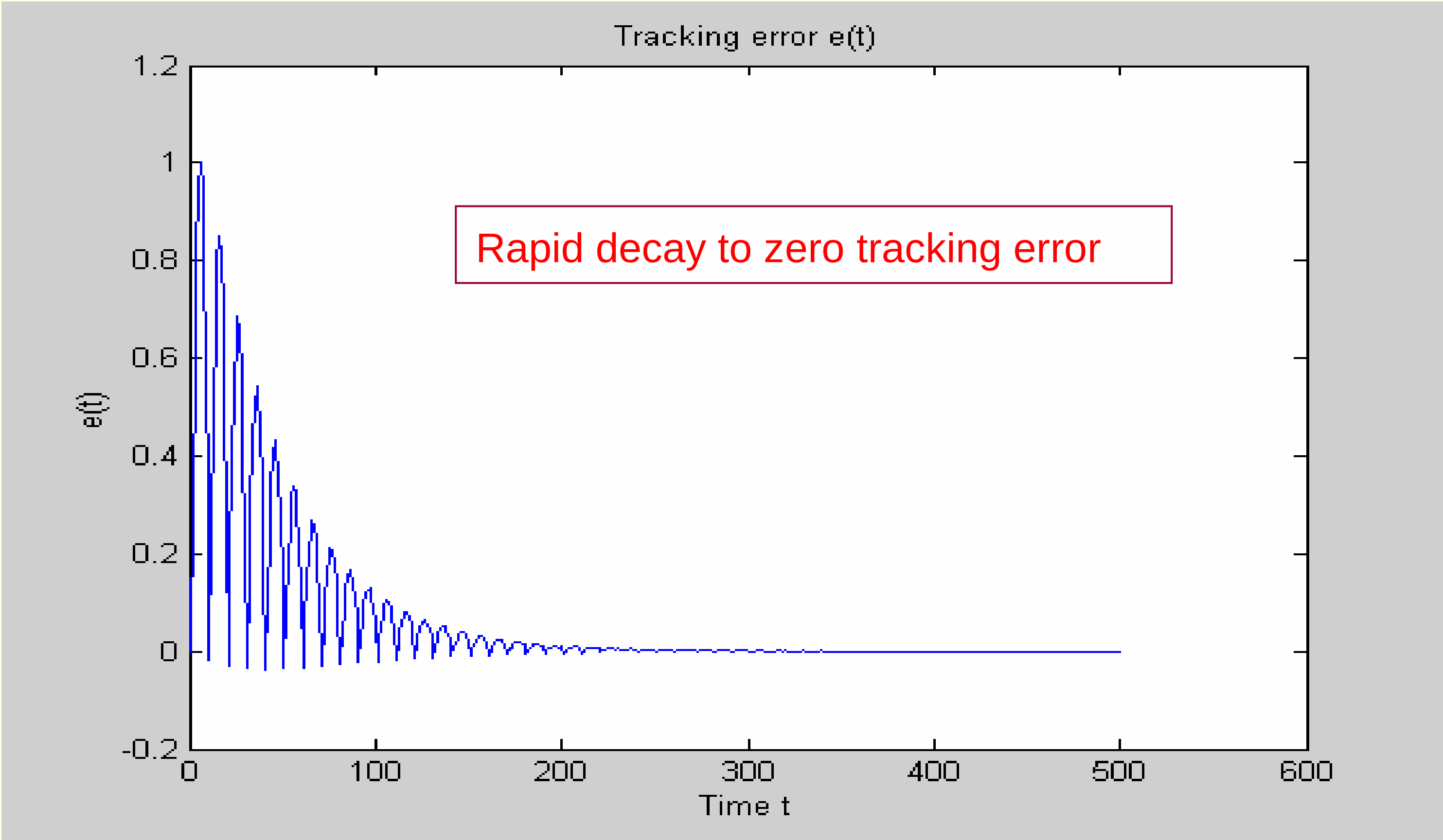
RC/Simulation Example

(Steady State) Results with a PID-controller



RC/Simulation Example

Results with a RC controller



RC/Applications

- Reported applications include:
  1. Control of rotating machines
  2. Control of PWM-inverters
  3. Casting
  4. Rolling processes
- Several international patents especially in the metal industry

Generalisations

- ILC/RC systems are a special case of more general 2-D systems

$$\begin{cases} x_{k+1}(t+1) = Ax_{k+1}(t) + B_1 u_{k+1}(t) + B_2 y_k(t),\ x_{k+1}(0) = d_{k+1} \\ y_{k+1}(t) = Cx_{k+1}(t) + Dy_k(t) \end{cases}$$

where $t \in [0, T]$

- This classes of processing can be found for example in mining and sanding.
- The stability properties of this system depends only on the *D* matrix!!!

Conclusions

- Delays, Repetitive Control and Iterative Learning Control are aspects of similar theories
- All are subject to the “delay effect” of destabilization or performance deterioration
- Effective optimal control laws produces monotonic convergence using similar control structure
- There are great possibilities for the use of these (more sophisticated) control laws
- Investment need not be great and could lead to increased production rate scenarios
- The price you pay is the need to think in a non-classical way during control design