Structure and Function of MIMO Benchmarking Software

Andrzej Ordys, Damien Uduehi
and Hao Xia

The Measurements of Success

The following excerpt from Ahmad & Benson (1999)[1] describes what is a world-class process plant:

> ***A world-class process manufacturing plant delivers outstanding custom service from reliable assets exhibiting operational excellence. It is operated by highly-motivated people and always maintains its licence to operate by satisfying the high safety and environmental standards of the process industries.***

The process measurements can be summarised into the following categories:

- **Custom service**: On time full delivery (OTIF), customer complaints, due date reliability, stock turn
- **Motivated People:** Absenteeism, training days, staff turnover.
- **Safety, health and environment**
- **Reliable assets:** Production rate, quality rate, availability,
- **Operational excellence**: Statistical process control, Manufacturing velocity

[1]. Ahmad, M & Benson , R (1999) *Benchmarking in the process industries. IChemE, Rugby. U.K.*

The Technical Benchmarking Process

The need for technical process benchmarking is determined by the business benchmarking. It usually consists of the following steps:

1. Find a technically and physically meaningful performance metric. It should be translated from economic metrics.

2. Prioritise the sub-process which is critical to the metric.

3. Find the best controller which can achieve the best score for this metric.

4. Analyse the causes for the gap in performance, *i.e.* find out the bottlenecks.

5. Provide solution/advice for improvements.

A Typical Plant-Wide Control Structure

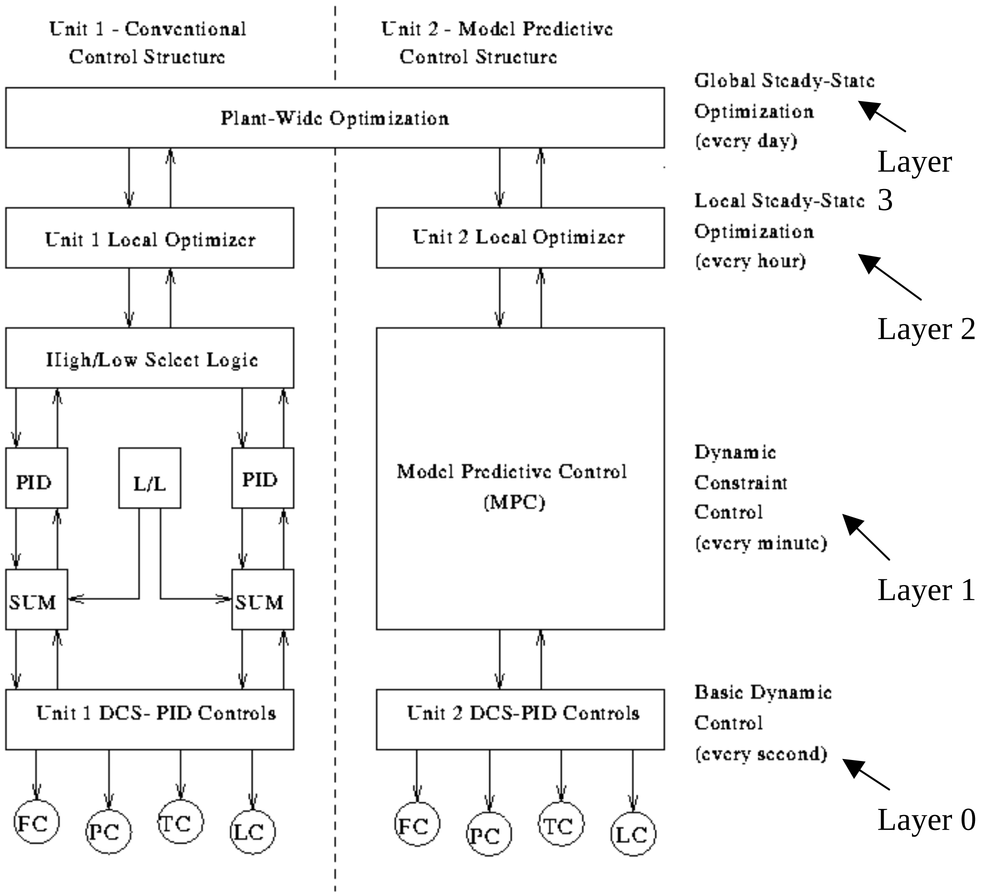


Fig.1 A typical layered plant-wide control structure.

Observations:

1. The plant is controlled in a hierarchical manner.
2. The commands flow from the top to the bottom.
3. The upper layer provides setpoints for the lower layer.
4. The steady state optimisation is performed to optimise profits and the whole operation should follow this objective.
5. The information from the lower layer is used as constraints in the upper level optimisation.

Some Thoughts on Static Optimisation

1. Problem formulation:

$$\min_{y_s,u} J^*(y_s,u)$$
$$s.t. \quad y = f_s(u), u_{min} \le u \le u_{max}$$
$$y_{min} \le y_s \le y_{max}, \quad g(y_s,u) \le 0$$

2. What is involved in the optimisation?

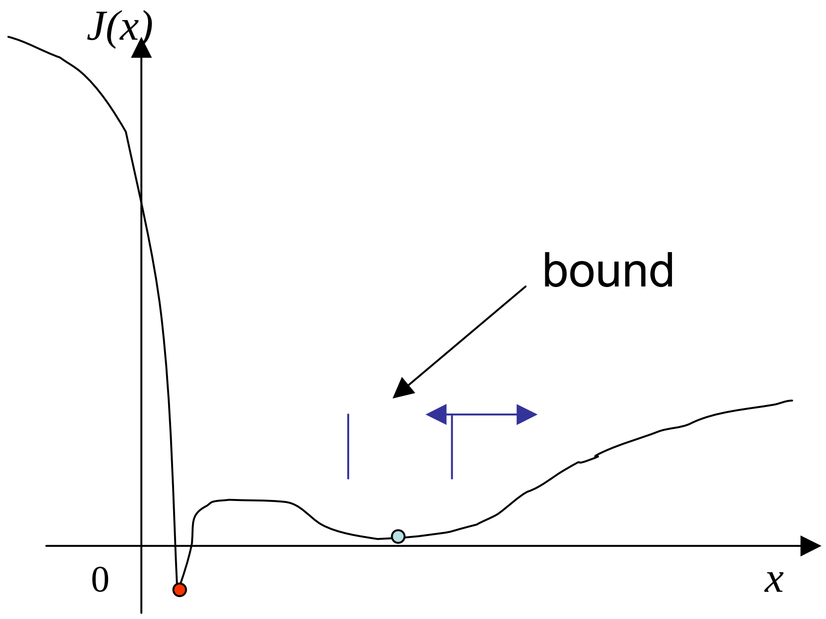
1. Economic cost function .
2. The cause/effect relationship between input/output
3. The constraints.

3.The central piece of the above optimisation problem is the static plant model:

1. It is time-varying, nonlinear.
2. It only includes the most important factors.
3. It represents the static relationships between input/output, *i.e.* many dynamical properties are omitted during the modelling.
4. Identifying the most important factors requires a sound understanding of the process.
5. The quality of this model decides the economic performance of the plant.

The Desirable Output From Static Optimisation

Ideally, the optimisation should not only decide the value of set-points but also decide the tolerable bounds around the set-points such that the cost function will not change much within the bounds.



An Alternative Control Structure

Model Predictive Control Structure.

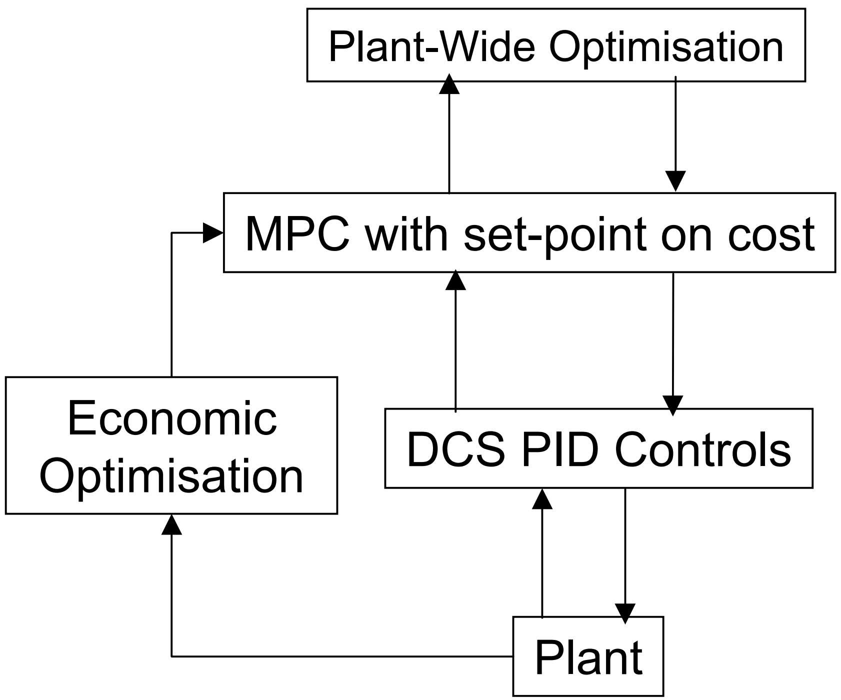


Fig 2. MPC with dynamic optimisation.

Optimisation formulation:

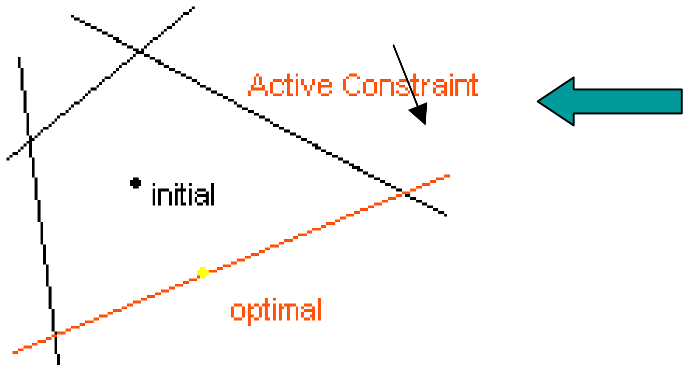
$$
\begin{aligned}
&\min_{y,u} J^*(y,u) \\
&subject \;\; to: \\
&J^*(y,u) := f_r(y,u) + \lambda f_e(y,u) \\
&y(k+1) = f(y(k), y(k-1), \ldots, u(k), u(k-1), \ldots) \\
&u_{min} \le u \le u_{max}, y_{min} \le y \le y_{max}, \\
&g(y,u) \le 0
\end{aligned}
$$

Remarks:

Since economic benefit is an integrated part of the criterion, this function may not be in the quadratic form, it may even include non-linear or discrete terms.

Soft/Hard Constraints

The optimisation problems formulated before are **constrained** optimisation problems.



In many cases, the optimal solutions are obtained with some constraints active.

Fig 3. Static constrained optimisation.

Two types of constraints:

- Hard constraints: the one can not be changed by re-tuning controller.
- Soft constraints: the one can be changed by controller tuning.

Question 1: Can we push these constraints further?

By controller benchmarking, this question can be answered.

Controller Benchmarking for Profit Optimisation

Question 2: Do we need to retune the controller?

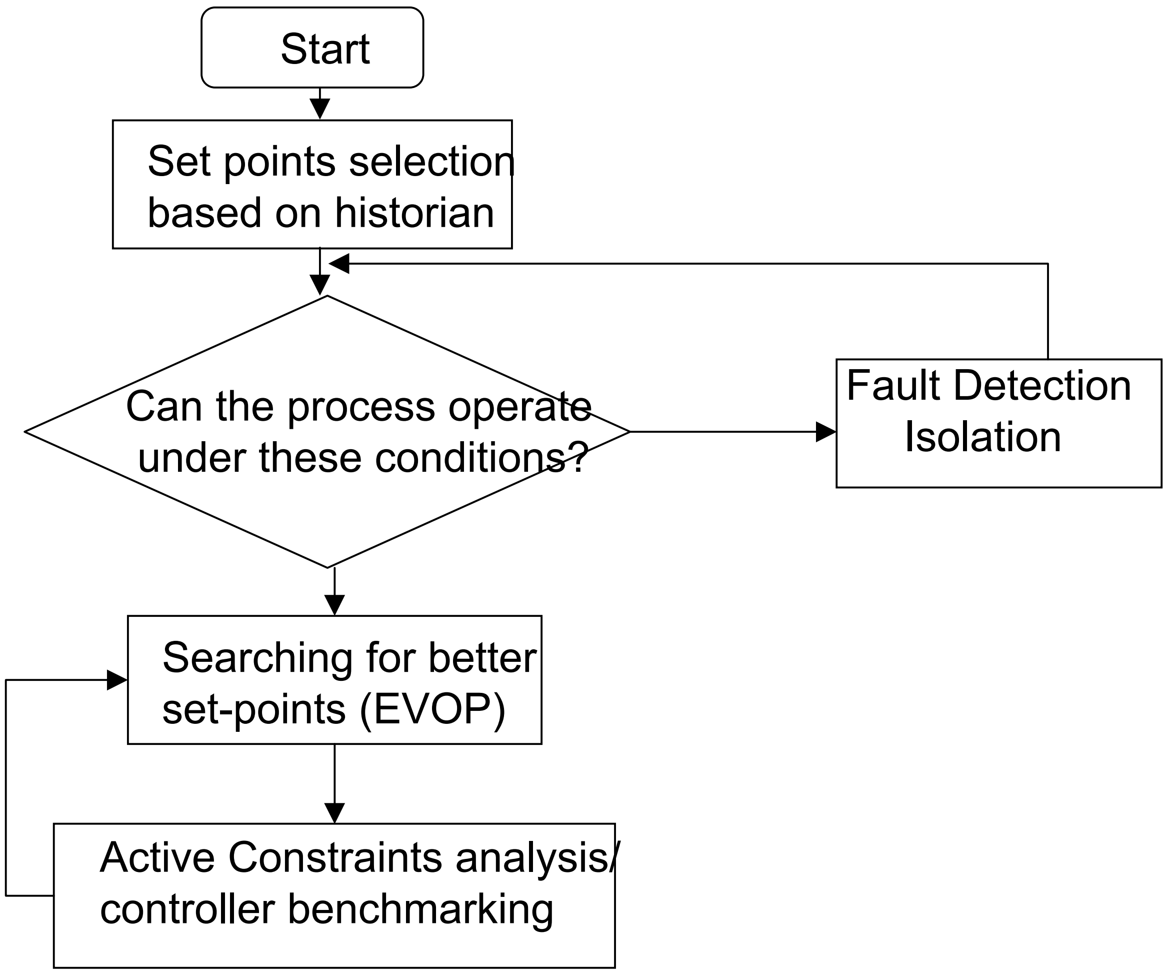
The prioritisation of re-tuning control loop(s) can be formulated as another optimisation problem:

$$
\begin{aligned}
&\max_C \Delta J^*(y,u) - J_c(y) \\
&s.t.\ \ \Delta J^*(y,u) - J_c(y) \geq 0 \\
&y(k+1) = f(y(k), y(k-1), \ldots, u(k), u(k-1), \ldots) \\
&u_{min} \leq u \leq u_{max}, y'_{min} \leq y \leq y'_{max}, \\
&g(y,u) \leq 0
\end{aligned}
$$

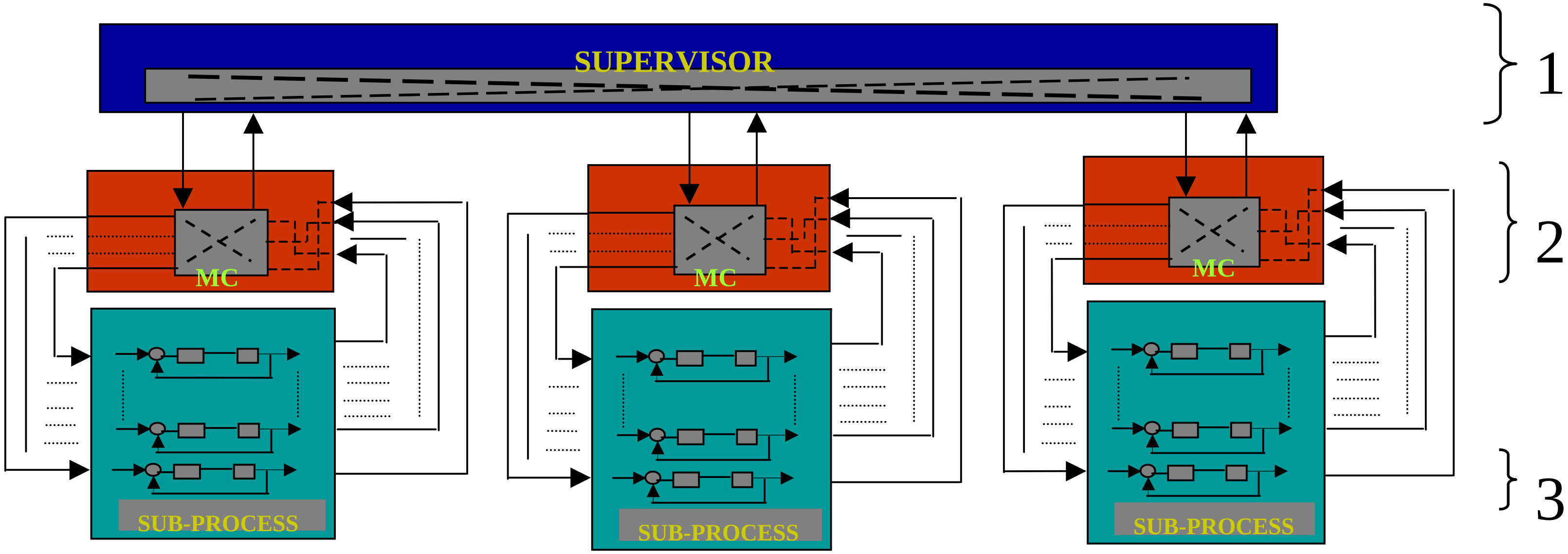
Remarks:

1. $J_c(y)$ should be defined by discussing with the industrial partners.
2. By focusing on the active constraints, we can identify the critical control loops or subsystems which have the biggest impacts on the plants' economic performance.
3. We only need to benchmark the subsystem related with the active constraints. The benchmarking problem becomes manageable.

Proposed Procedure of Optimisation



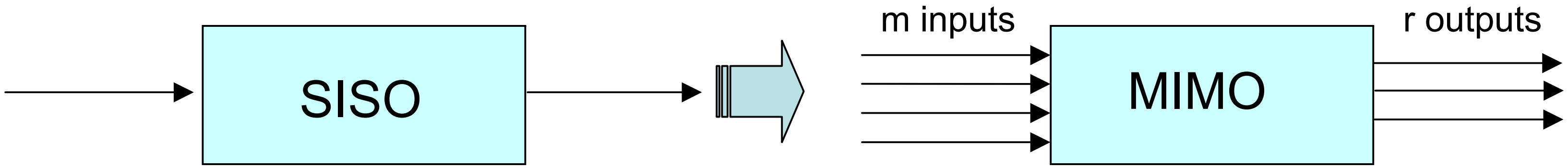
Why a MIMO Benchmark



MIMO systems contain loop interactions and recycles

- *Optimising each loop , might lead to system instability*
- *SISO benchmarking indices cannot be extended to the MIMO case*
- *MIMO benchmark for overall sub-process required*

Benchmarking of Multivariable Processes



$$W = z^{-k}\frac{B}{A}$$

$$\underset{r\times m}{W} = \underset{r\times r}{A^{-1}}\,\underset{r\times m}{B} = \underset{r\times r}{A^{-1}}\,\underset{r\times r}{D^{-1}}\,\underset{r\times m}{\tilde{B}} = \underset{r\times r}{D_r^{-1}}\,\underset{r\times r}{A_r^{-1}}\,\underset{r\times m}{\tilde{B}}$$

matrix fraction

inverse interactor matrix

Extension to multivariable systems is generally nontrivial. Possible difficulties are a result of:

- interactions between loops
- loops need to be prioritised to obtain desired objective
- performance is also dependent on control structure

Benchmarking Multivariable Processes

Optimal Benchmark Function

- Require an index that captures both transient and steady state performance
- For steady state variance is a good KPI
- For step changes, the KPI’s directly influence the integral square error (ISE) of the process
- The Optimal benchmarking function should measure the (ISE) and some weighted combination of variances
- An LQGPC cost function would provide the criteria required to benchmark the process

Multivariable Predictive Benchmarks

- The LQGPC Cost Function can be expressed as

$$J = \mathrm{E}\left\{ \lim_{T\to\infty} \frac{1}{T+1} \sum_{t=0}^{T} J_t \right\} \quad \text{where} \quad J_t = \mathrm{E}\left\{ \sum_{j=0}^{N} \left[\left(y_{t+j+1} - r_{t+j+1}\right)^T \breve{Q}_e \left(y_{t+j+1} - r_{t+j+1}\right) + \Delta u_{t+j}{}^T \breve{Q}_u \Delta u_{t+j} \right] \right\}$$

$$\textit{with}, \ \breve{Q}_e > 0 \ \text{ and } \ \breve{Q}_u \geq 0$$

■ Properties of LQGPC Optimisation

- The objective of the GPC function (J_t) is to minimise the sum of the squares of predicted system outputs and inputs.
- The function (J_t) is quadratic, so where a solution exist, it is unique.
- Minimising (J_t) optimise the predicted system trajectory from time (t+1) to (t+N), for a given initial state at time t.
- Minimisation is done with the assumption that Δu_{t+j} for j > N+1 is zero, i.e system reaches steady state.

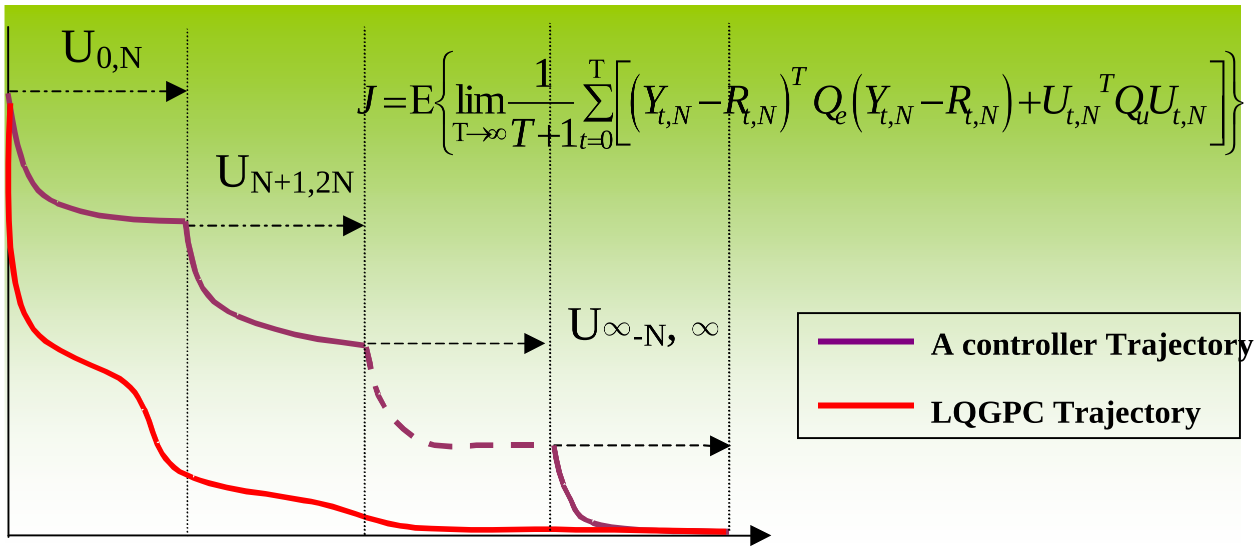
Assumption may not be valid for some choices of N

Multivariable Predictive Benchmarks

- Because the optimisation of (J_t) is finite (t+1 to t+N, N < ∞), there is no guarantee that the system trajectory from t = 0 to t = ∞, is optimal for some $U_{t,N}$

- The LQGPC function (J) takes infinite sum of indices J_t ensuring that for any time instance t, the sequence $U_{t,N}$ is optimised to ensure that the path from
- t = 0 to t = ∞, is optimal
- Notice that, because of the integrator (Δu_t), the system terminal state (i.e. t=∞) x_{t+N+1} (in the absence of noise) or its mean value (in the presence of noise) at (t+N) must be the same for any controller with integral action

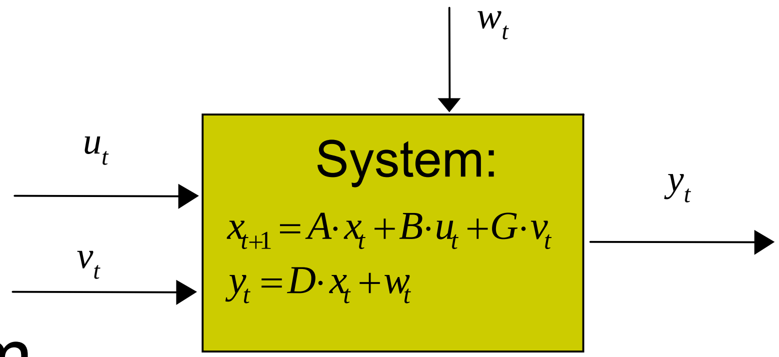
These properties are very useful in creating a predictive benchmark



A Feasible Predictive Benchmark

- Consider the system

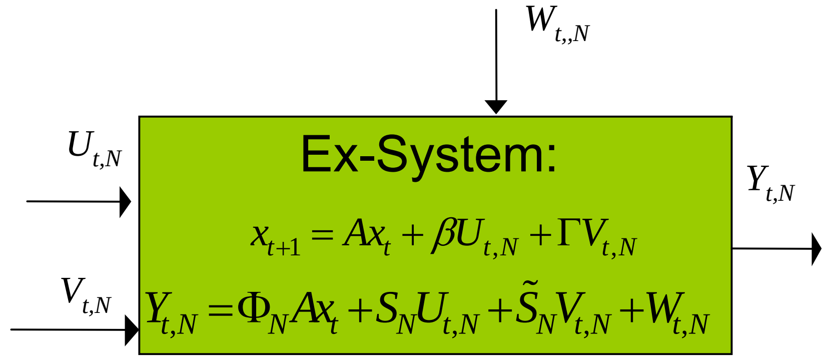
$$x_{t+1} = A \cdot x_t + B \cdot u_t + G \cdot v_t$$
$$y_t = D \cdot x_t + w_t$$



- Create benchmark system

$$x_{t+1} = Ax_t + \beta U_{t,N} + \Gamma V_{t,N}$$
$$Y_{t,N} = \Phi_N A x_t + S_N U_{t,N} + \tilde{S}_N V_{t,N} + W_{t,N}$$

$$U_{t,N} = \begin{bmatrix} u_t \\ u_{t+1} \\ \vdots \\ u_{t+N} \end{bmatrix} \qquad \beta = [B \quad O \quad \cdots \quad O]$$



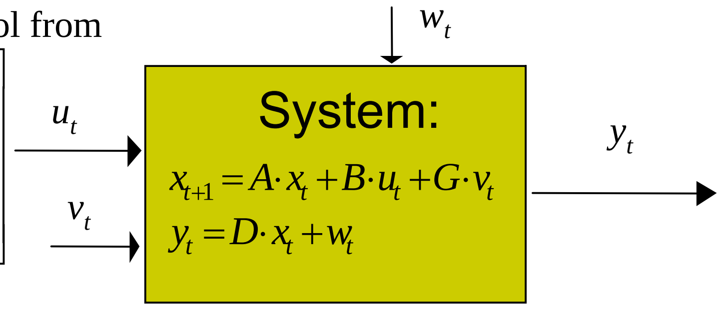
A Feasible Predictive Benchmark

- Optimal Controller

For t =i to i+N, take control from sequence $U_{t,N}$

$$U_{t,N} = \begin{bmatrix} u_t \\ u_{t+1} \\ \vdots \\ u_{t+N} \end{bmatrix}$$

Open loop

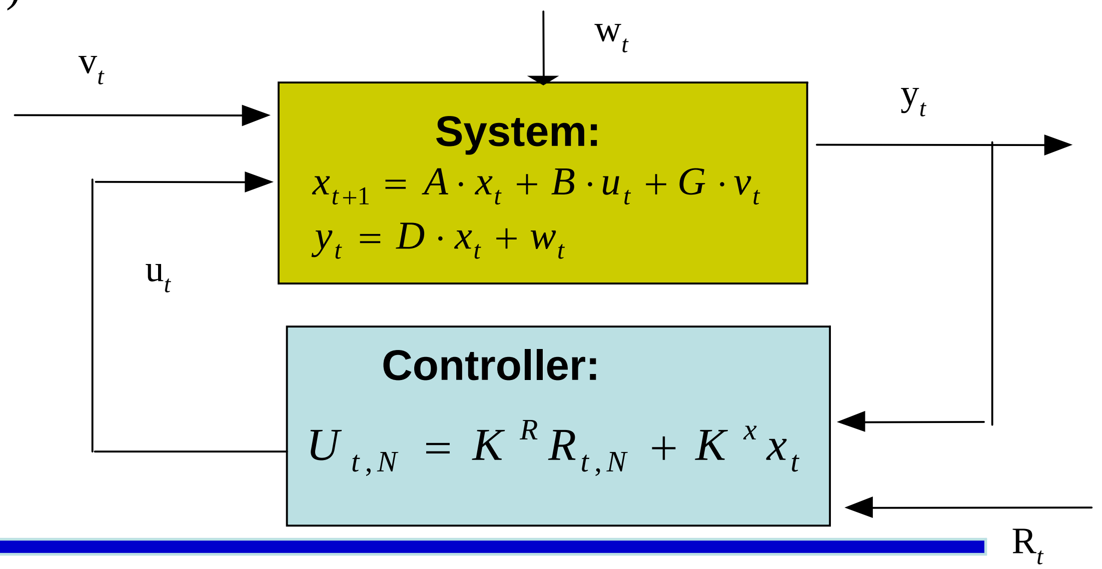


$$U_{t,N} = K^R_{LQGPC} R_{t,N} + K^x_{LQGPC} x_t$$

$$where\ K^R_{LQGPC}\ \&\ K^x_{LQGPC} x_t = f(\Phi_{Nt}, S_N)$$

$$S_N = \begin{bmatrix} DB & O & \cdots & O \\ DAB & DB & & \vdots \\ \vdots & \vdots & \ddots & O \\ DA^N B & DA^{N-1} B & \cdots & DB \end{bmatrix}, \quad \Phi_N = \begin{bmatrix} D \\ DA \\ \vdots \\ DA^N \end{bmatrix}$$

Closed loop



A Feasible Predictive Benchmark

- System Under Optimal Control:-*Then at each time instance t,*
  - The process follows an optimal state trajectory, that drives the output to the desired level
  - Trajectory is guaranteed to provide smallest value of cost and use the least amount of energy
  - once on this trajectory the state require minimum power to remain / follow the trajectory
- System Under Non-optimal Control:-*Then at each time instance t,*
  - State trajectory is sub-optimal, however the system may still arrive at the desired output level
  - Energy is wasted in arriving at the desired output level
  - Ultimately it cost more to follow this trajectory than the optimal state path

A Feasible Predictive Benchmark

- Ideally predictive benchmarking measures the energy differential between the predicted trajectory of the sub-optimal control and the predicted optimal trajectory.
- In both cases the prediction is over a finite horizon of N steps.
- The benchmark is therefore calculated for each time instance
  - We do not have the predicted trajectory of the sub-optimal controller, but it's actual trajectory.
- Instead we can measure the amount of work required by the ideal predictive controller at each time instance (t) to go from some state of the process under sub-optimal control to the desired terminal state in the interval (t+1,t+N).
  - The more sub-optimal the controller, the greater the amount of work required.
  - This benchmark is a measure of the predicted amount of wasted energy in the system in the interval (t+1,t+N).

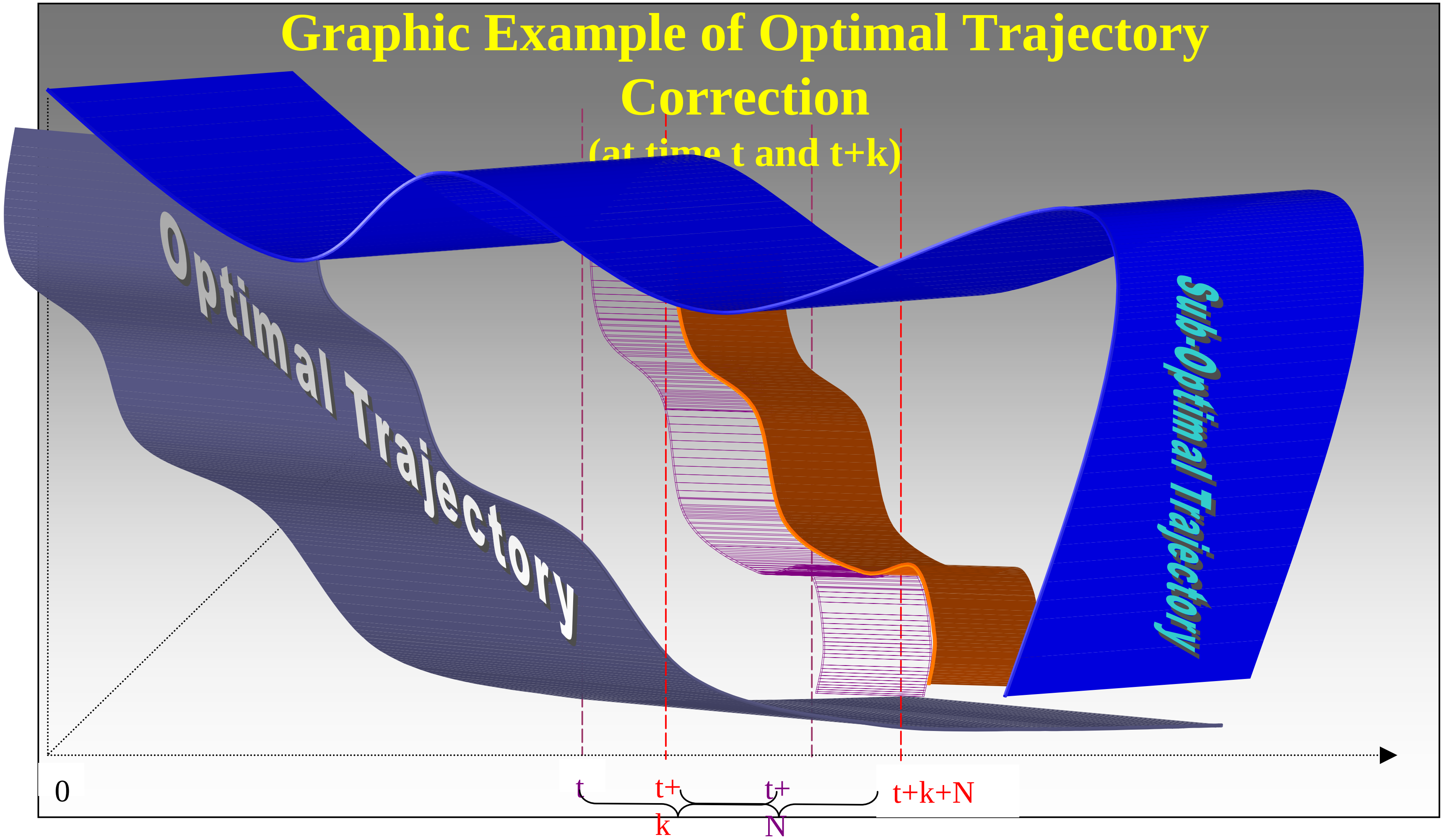
Practically, to do it, assume that at each time instance t, the process is controlled by a controller which may/may-not be optimal. But at time t+1, the process is switched onto optimal control

A Feasible Predictive Benchmark

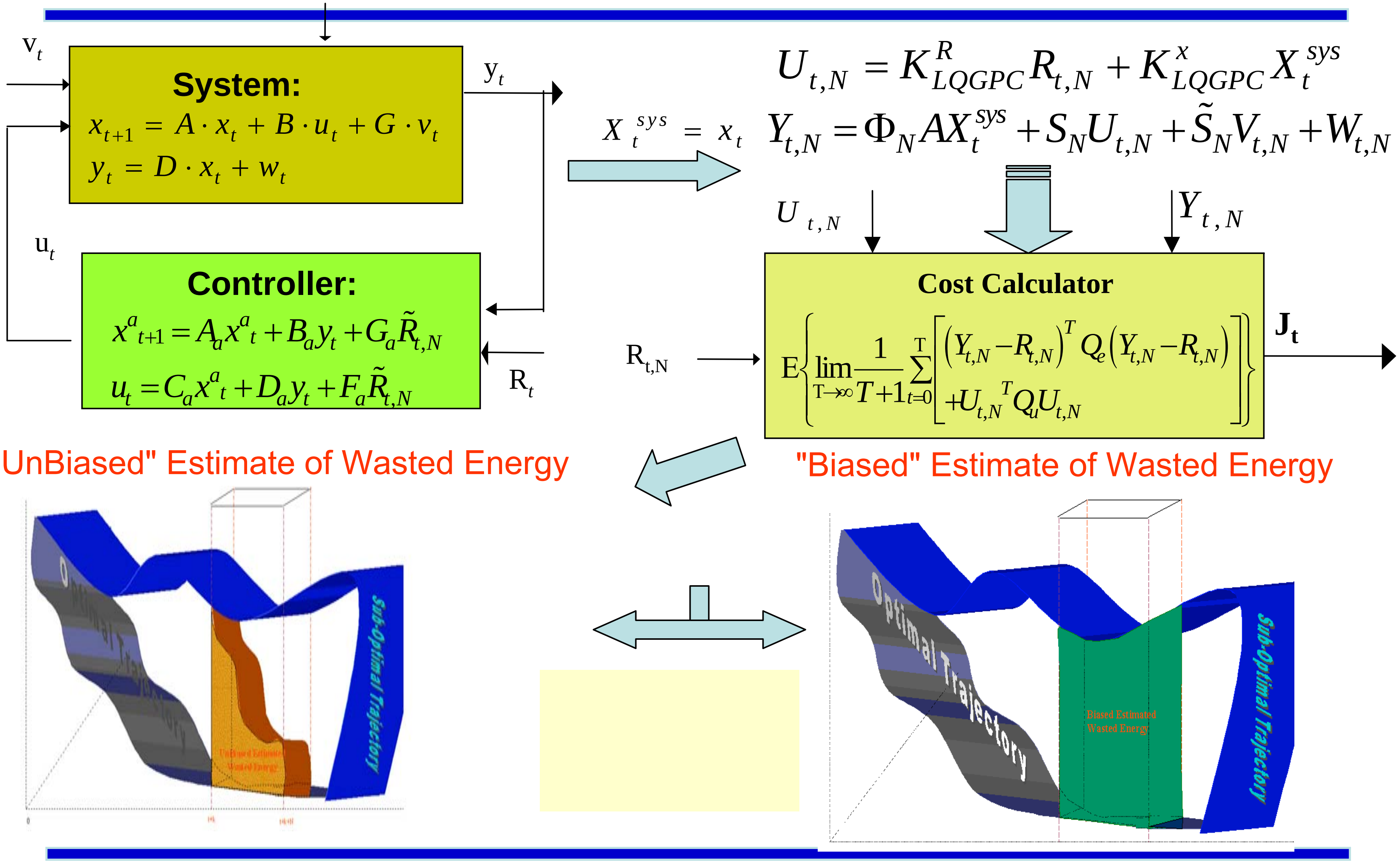
Consequently, this benchmark assumes, at the time instance t+1 (predicted), switching between the controllers

- Switching between Two Optimal Controllers
  - The controller in control of the process requires no additional power to keep the system on the optimal state trajectory
  - The total energy used (between the two controllers) is equal to the minimum required to drive the output to the desired level
  - The value of the cost is unchanged
- Switching between Non-optimal and Optimal Controller
  - The optimal controller will require additional power to bring the system state into an optimal state trajectory
  - Energy is wasted in correcting the state path to follow the optimal trajectory
  - The total energy used between the two controllers is greater than the minimum required to arrive at the desired output

A Feasible Predictive Benchmark



A Feasible Predictive Benchmark



Feasible Predictive Benchmark

Benchmark Index

$$\eta_t = \frac{J_t(LQGPC)}{J_t(Nominal)} \Longrightarrow \eta_t = \frac{Total\ Energy\ Used\ (LQGPC)}{Total\ Energy\ Used\ (Nominal)}$$

Efficiency of nominal controller

or

$$\Im_t = 1 - \eta_t = \frac{J_t(Nominal) - J_t(LQGPC)}{J_t(Nominal)} \Longrightarrow \Im_t = \frac{Wasted\ Energy}{Total\ Energy\ Used\ (Nominal)}$$

Inefficiency of nominal controller

Multivariable Predictive Benchmarking

Example

Pilot-scale binary distillation column model used for methanol-water separation

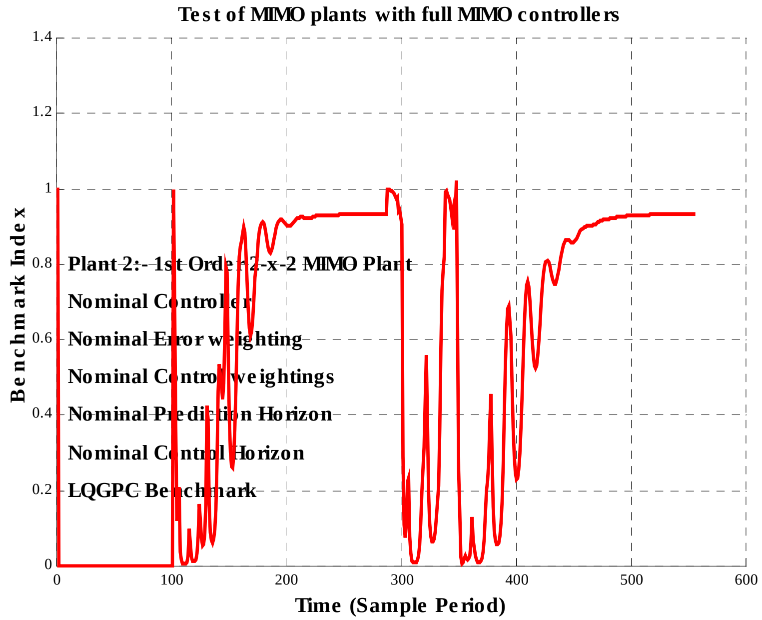
$$\begin{bmatrix} y_1(s) \\ y_2(s) \end{bmatrix} = \begin{bmatrix} \dfrac{12.8e^{-s}}{16.7s+1} & \dfrac{-18.9e^{-3s}}{21s+1} \\ \dfrac{6.6e^{-7s}}{10.9s+1} & \dfrac{-19.4e^{-3s}}{14.4s+1} \end{bmatrix} \begin{bmatrix} u_1(s) \\ u_2(s) \end{bmatrix} + \begin{bmatrix} \dfrac{3.7e^{-8.1s}}{14.9s+1} \\ \dfrac{4.9e^{-3.4s}}{13.2s+1} \end{bmatrix} d(s)$$

- Controller 1: Multi-loop PID controller designed using modified Ziegler-Nichols
- Controller 2: De-tuned multi-loop controller

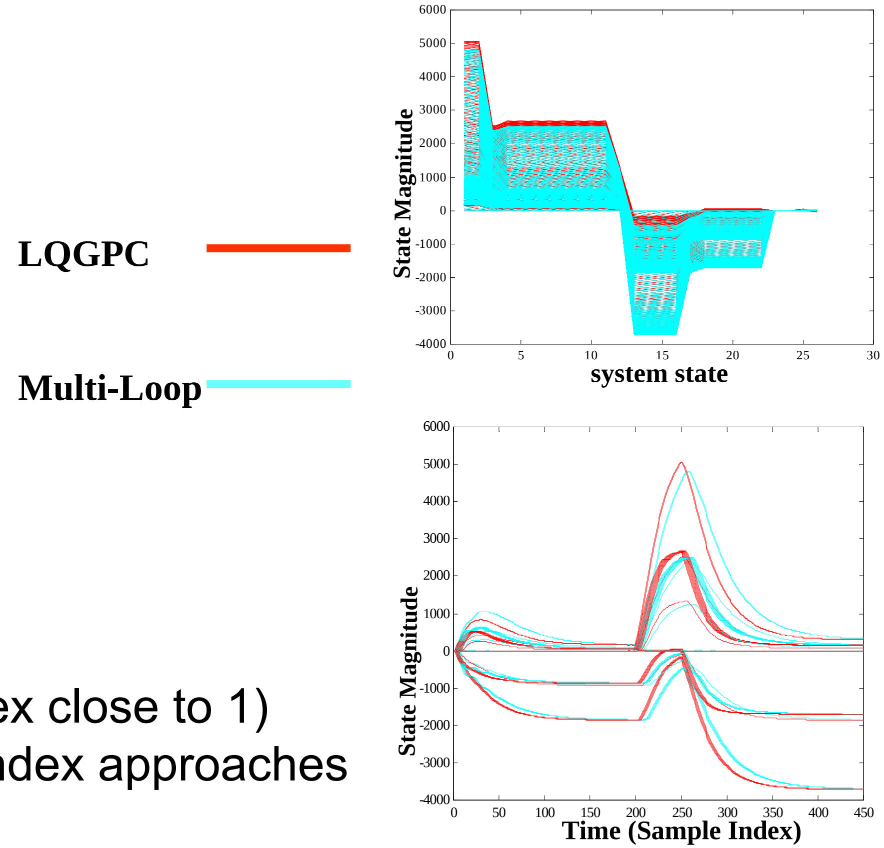
Multivariable Predictive Benchmarking

Example

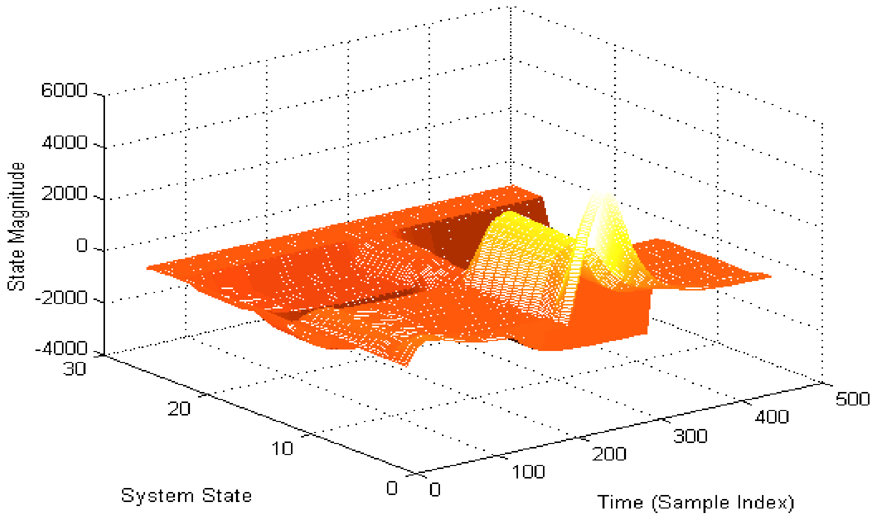
- Steady State and Transient Benchmark of Controller 1



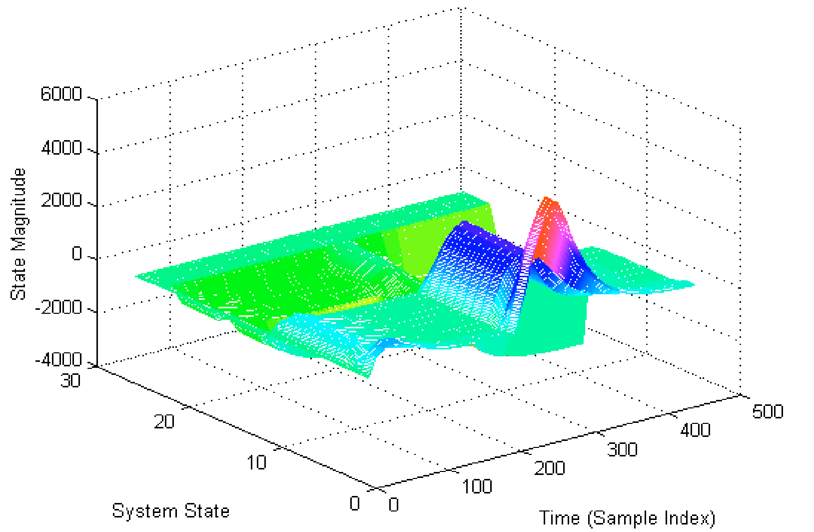
- System Trajectories (Rear and Side views)



- Trajectory patterns are similar
- Track closely at steady state (index close to 1)
- Marked differences in transient (index approaches 0)

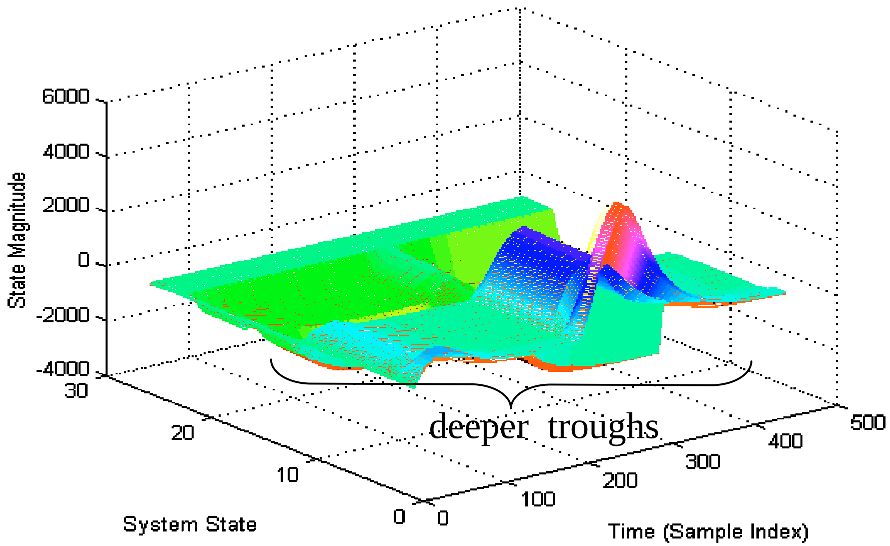


- LQGPC System Trajectories (3-D views)

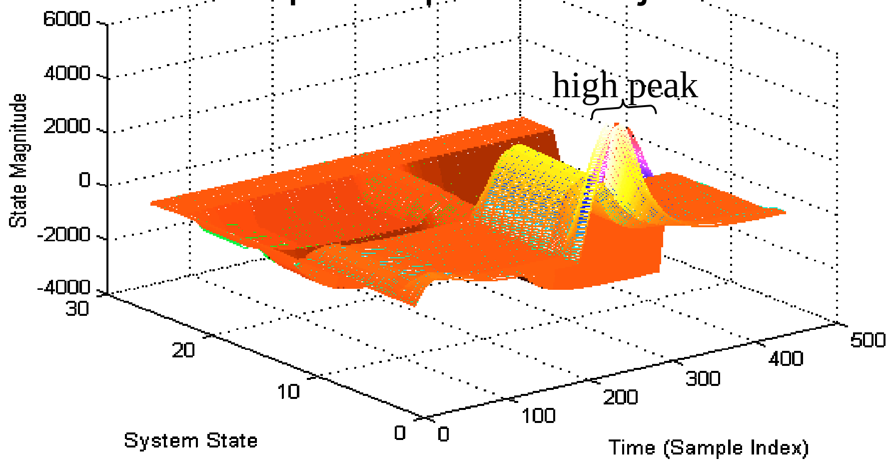


- Mulit-Loop System Trajectories (3-D views)

- Super Imposed Trajectories



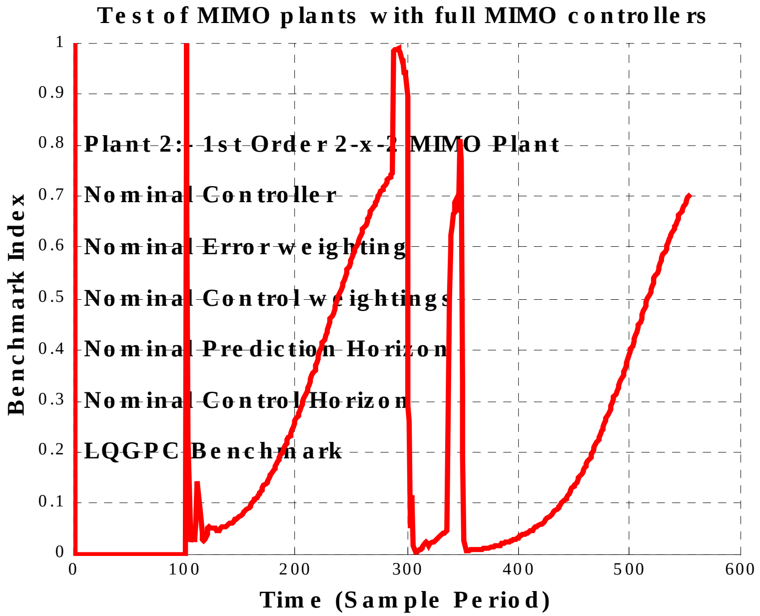
- Super Imposed Trajectories



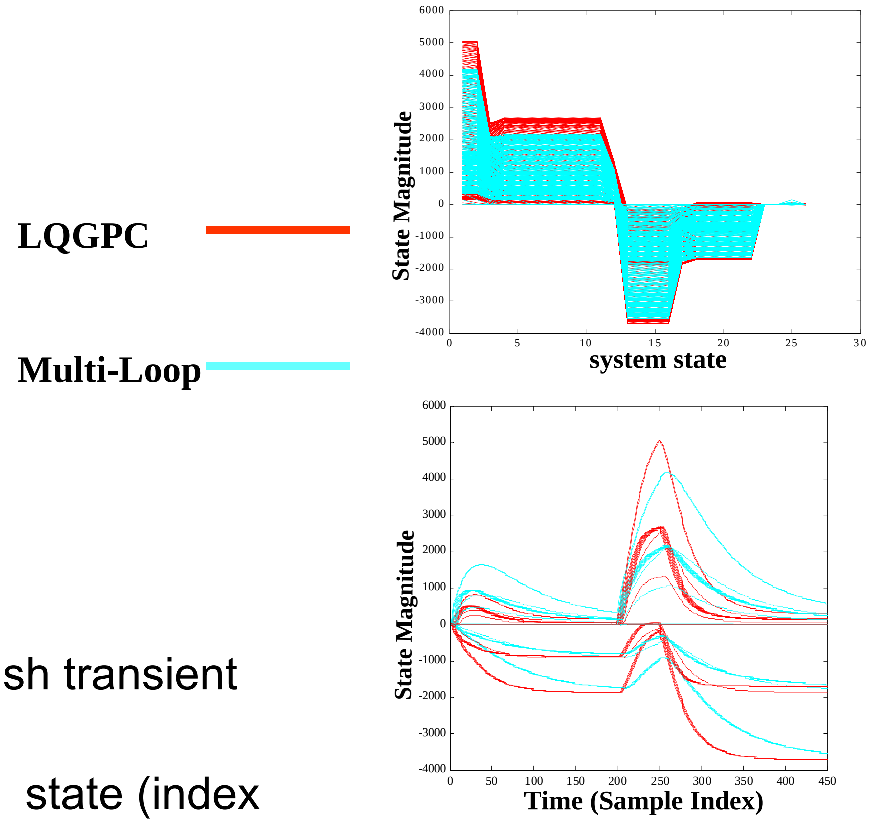
Multivariable Predictive Benchmarking

Example

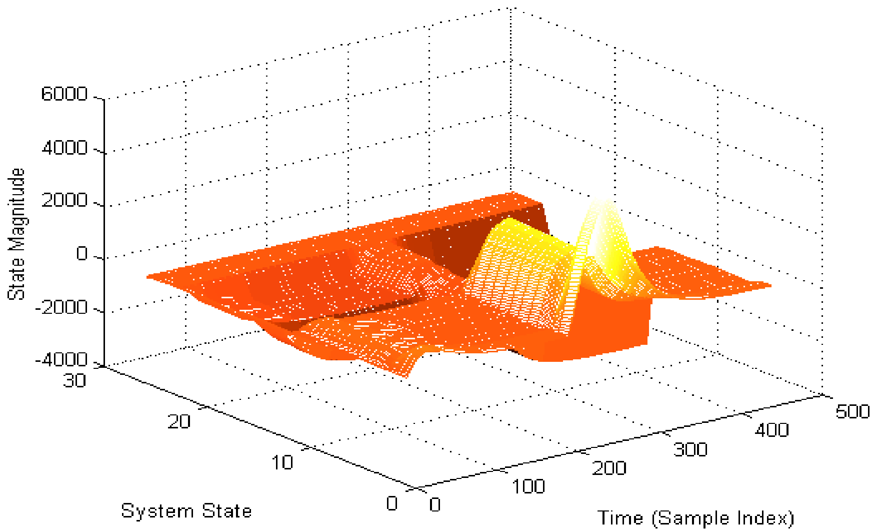
- Steady State and Transient Benchmark of Controller 2



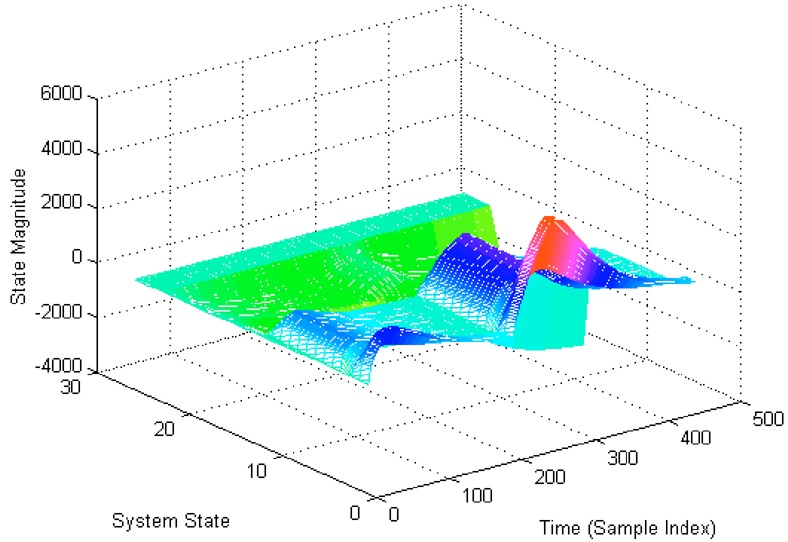
- System Trajectories (Rear and Side views)



- Trajectory patterns are dissimilar
- Marked differences in transient, sluggish transient (index approaches 0)
- Performance improves towards steady state (index approaches 1)

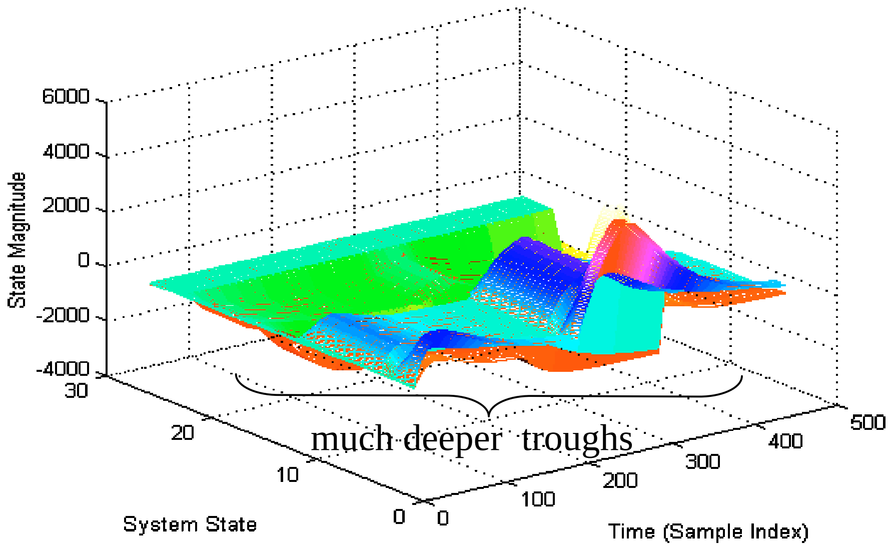


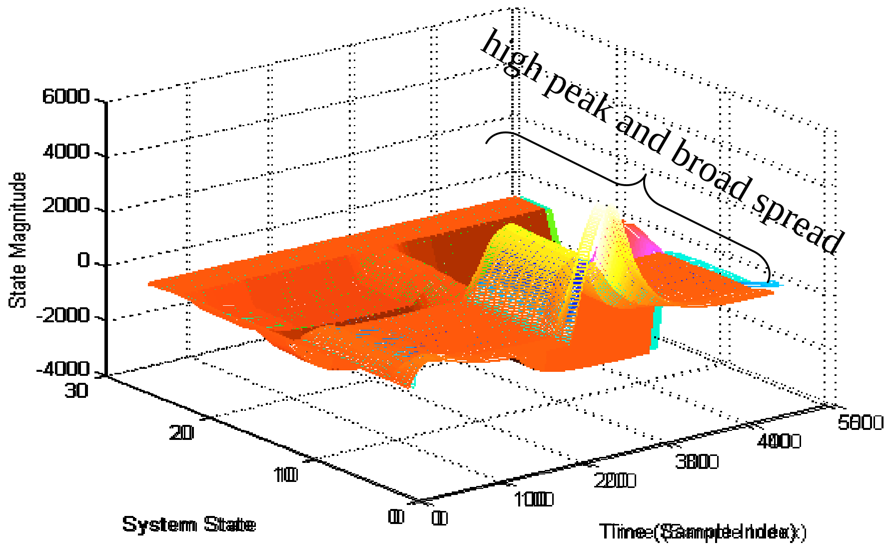
- LQGPC System Trajectories (3-D views)



- Mulit-Loop System Trajectories (3-D views)

- Super Imposed Trajectories





Conclusions

Hierarchical structure of industrial control systems require hierarchical benchmarks

The identification of active constraints can help to prioritise the control loops

MIMO benchmarking is critical for proper assessment of interacting control loops

LQGPC performance has been used to consider dynamic performance measures of MIMO systems