The European Commission
Community Research
ist information society technologies

Fifth Framework Programme

P A M

Performance Assessment and Benchmarking in Applications

Dr. Andrzej Ordys & Dr. Hao Xia

Topics Covered

- Overview
- Divided wall column example
  - SISO benchmarking procedure
  - MIMO benchmarking procedure
- Steam turbine example
- Discussions

Overview

In this presentation we are going to illustrate the benchmarking exercises on two real-world industrial simulation case studies:

- The first model is a divided column distillation plant. This plant model is quite characteristic of the chemical industry.

- The second one is a steam turbine of a coal fired power plant. The power plant system is representative of the industries present in the power and servo industrial sector.

- The control system sets-up and the performances are representative of the real process.

Overview

Characteristics of distillation column:

- stochastic sources acting on the system process,
- the systems' outputs tend to remain around a given value for long periods of time.
- if process variability is known and may be reduced, then it is possible to increase the process to its safe operational limits in order to improve productivity and efficiency.
- Since maintaining a steady product concentration level near 100% is the major control objective, the variances of the controlled temperatures should be reduced as much as possible. Taking variance as measure of performance is quite applicable and really meaningful.

Applicable benchmarks:

- MV, GMV , RS-LQG benchmarks

Overview

Characteristics of the steam turbine:

- references and disturbances tend to be deterministic in nature
- frequent references changes and hence output levels.
- variance as a measure of performance are not relevant, which implies that MV and GMV are not useful

Suggested benchmark:

- if disturbances and references models are available or can be easily obtained, then it is possible to use the RS-LQG algorithm to benchmark such systems.

Comparison of SISO Benchmarks

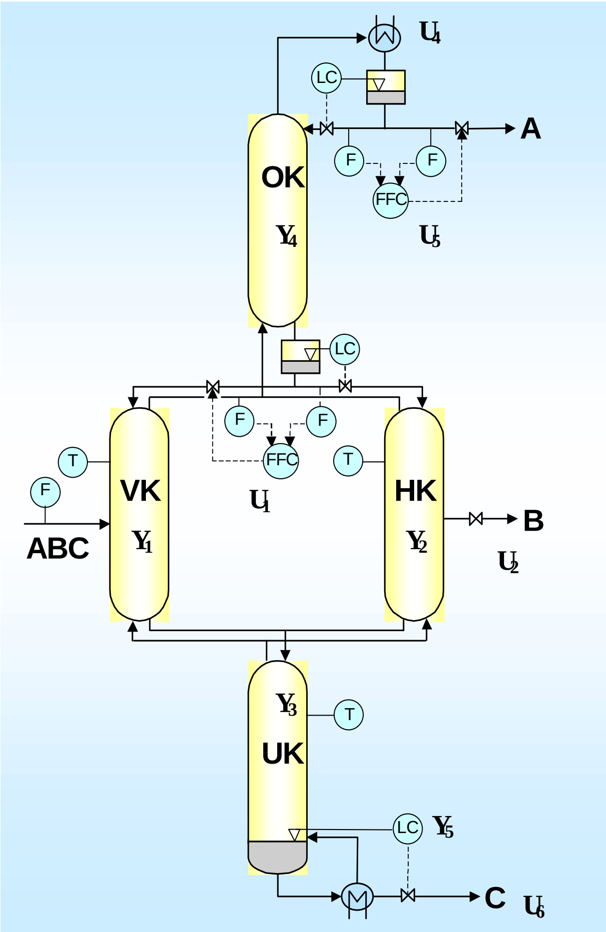
Table 1. Comparison of benchmarking methods: benefits

Benefit	MV	GMV	RS-LQG
Control performance	✓	✓	✓
Limits actuator energy	×	✓	✓
Reflects controller structure	×	×	✓
Provides optimal controller settings	×	×	✓

Table 2. Comparison of benchmarking methods: data requirements

Type of Data	MV	GMV	RS-LQG
Loop Delay	✓	✓	✓
Loop Output Error	✓	✓	-
Actuator Input	×	✓	-
Weighting Choices	×	✓	✓
System Model	×	×	✓
Controller Structure	×	×	✓

Divided Wall Column



- The ultimate control objective is the purity of all three components.
- No online measurements for purity in the plant
- Three temperatures are controlled as a substitute

Outputs	Inputs
Y1 : Temperature in VK	U1 : Split ratio between columns VK and HK
Y2 : Temperature in HK	U2 : Flow of component B
Y3 : Temperature in UK	U3 : Heating energy for component C
Y4 : Pressure in OK	U4 : Cooling energy in the condenser A
Y5 : Level in the column sump	U5 : Reflux ratio of component A
	U6 : Flow of component C

Currently applied pairing between Mv-s and Cv-s:

T_{VK} – reflux ration of A	Y1(U5)
T_{HK} – split ratio	Y2 (U1)
T_{UK} – flow of component B	Y3(U2)
P_{OK} – cooling energy in condenser of A	Y4(U4)
Luk – flow of component C,	Y5(U6).

MV Benchmark

MV: minimize the variance of the output $Var[Y(i)]$

ALGORITHM:

- Estimate the process time delay
- Estimate the minimum achievable variance
- Estimate the actual variance (or the mean square error)
- Compare these two values:

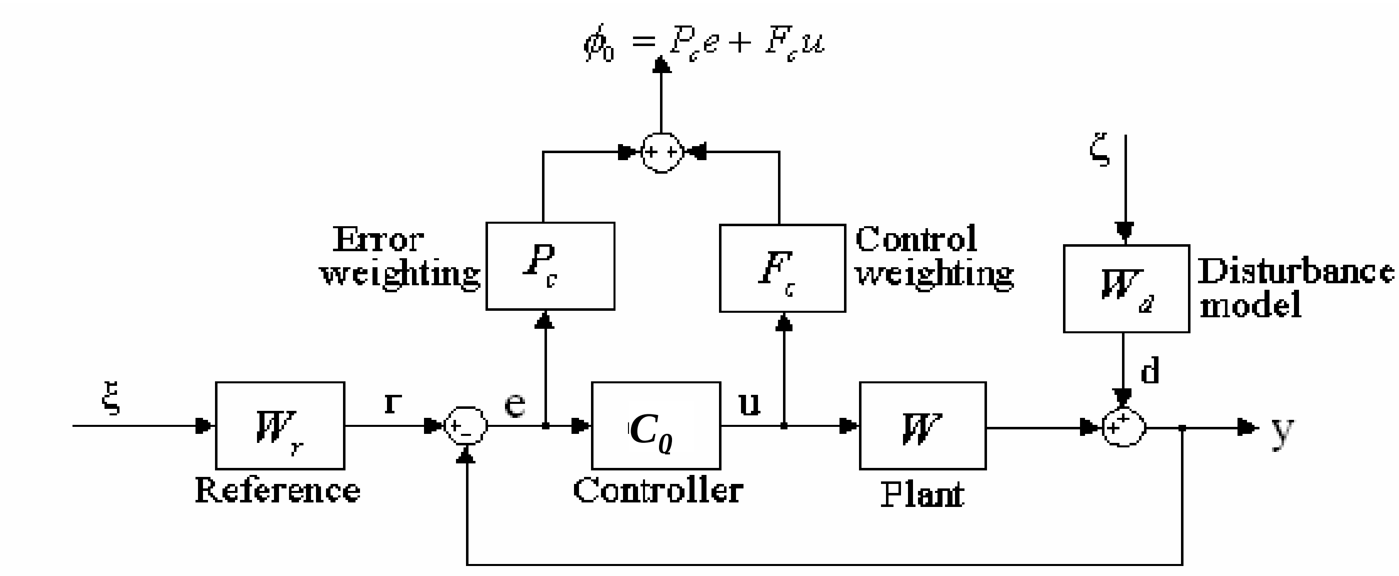
$$\hat{\eta}(k) = \frac{\hat{\sigma}_{mv}^2}{\hat{\sigma}_y^2 + \bar{y}^2} \in [0,1]$$

normalized performance index

1 MV control

0 poor control

GMV benchmark



GMV: minimize the variance of the "generalized" output f_0(t): $Var[\phi_0(t)]$

$$\boxed{\phi_0(t) = P_c e(t) + F_c u(t)}$$

MV/GMV Benchmark Computation

The MV and GMV benchmarking algorithms use only system data to compute the benchmark index. For both algorithms the user must define a data length and an autoregressive model length

- Data length and autoregressive model length are system specific
- Data length (n) influences the statistical confidence in the value of the performance index
- Autoregressive model length (m) should be such that the closed loop impulse response is fully captured with m-samples
- Generally (n) should be much greater than (m), typical values range from $n = 150 \times d$, to $n = 1500$

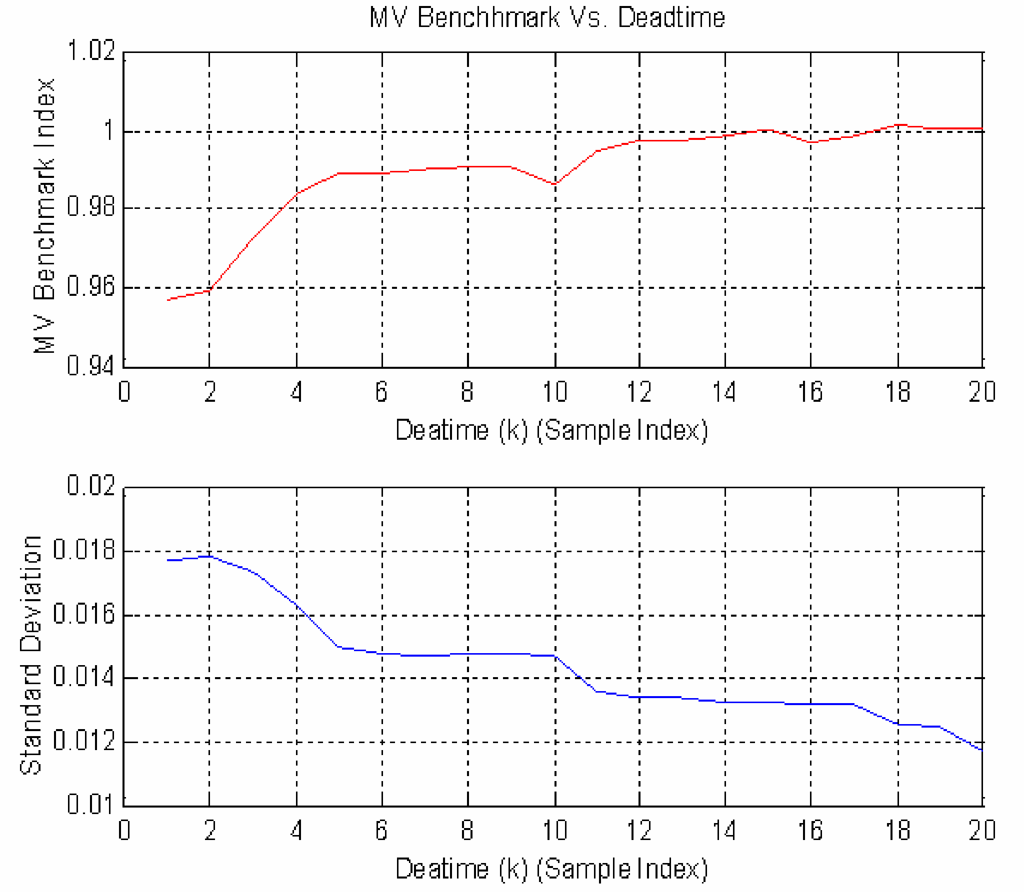
MV/GMV Benchmark Computation

The MV and GMV algorithms require that the estimate (k) of system time delay (d) be precise, i.e $k = d$

- If $k < d$, then, estimated index < true value of the of the loop performance
- By conducting a series of test using a range of time delay values, a curve can be constructed, with the true time delay and benchmark index as a point on the curve. This curve is user defined, i.e., the process controller is required to reduce the error variance to a some value in the given interval (k)

MV Benchmark Results, Loop 2

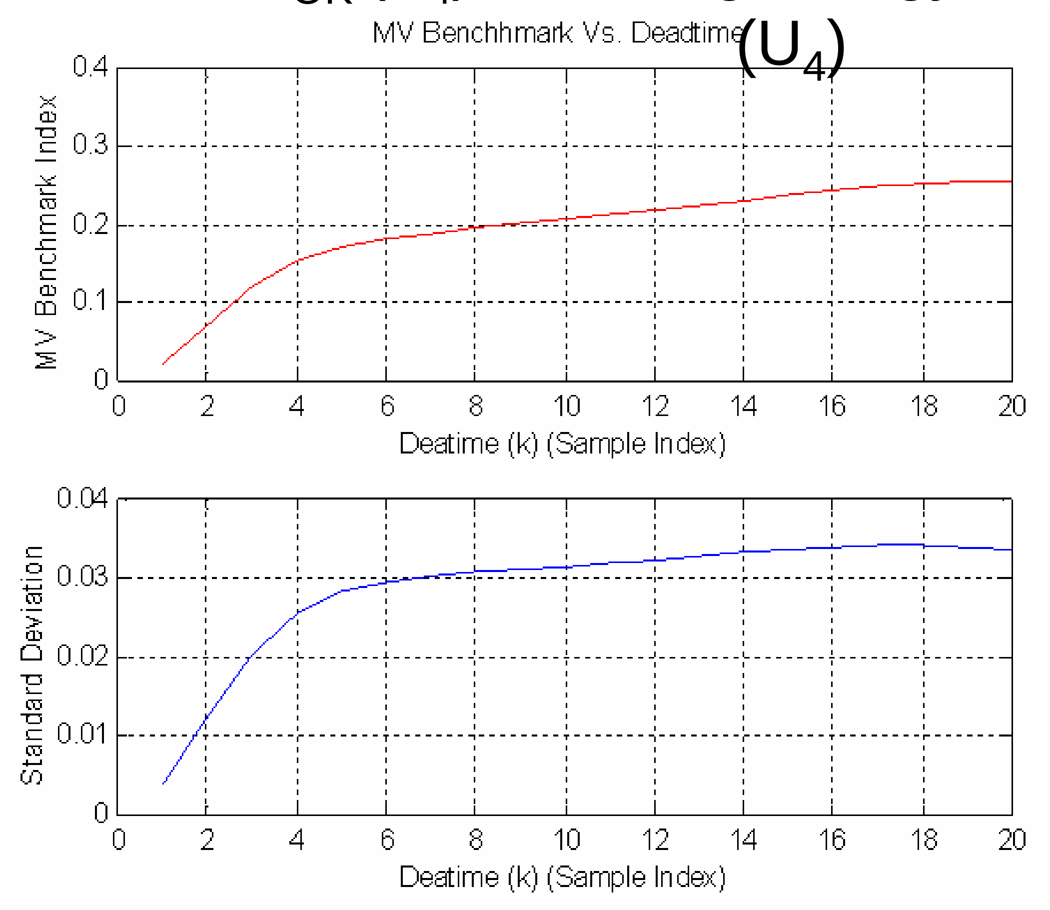
T_{HK} (Y_2) – split ratio (U_1)



- the value of the benchmark index did not change significantly as the dead time was varied.
- it is highly probable that the dead times for these loops is either 1 or 2 sample intervals.
- the existing controller is likely to be a MV controller

MV benchmark results, loop 4

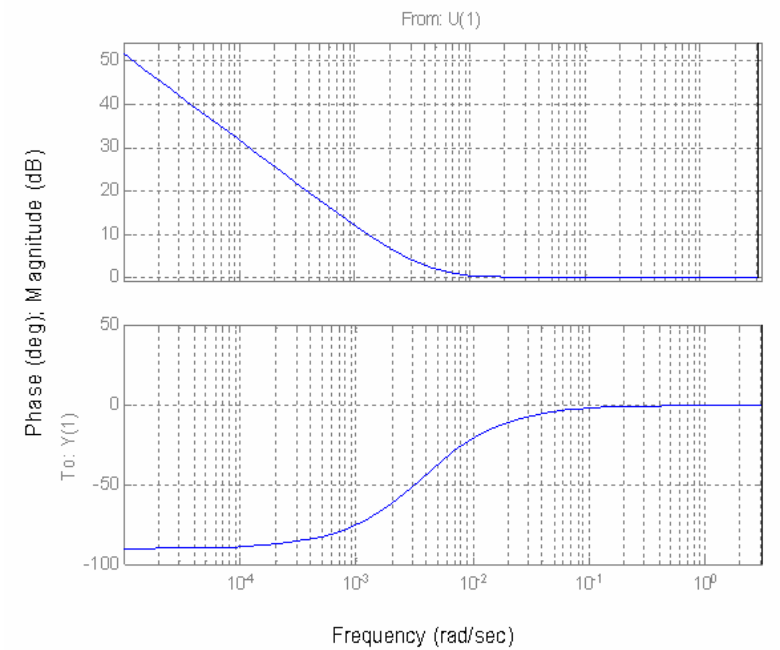
P_{OK} (Y_4) – cooling energy in reflux of A (U_4)



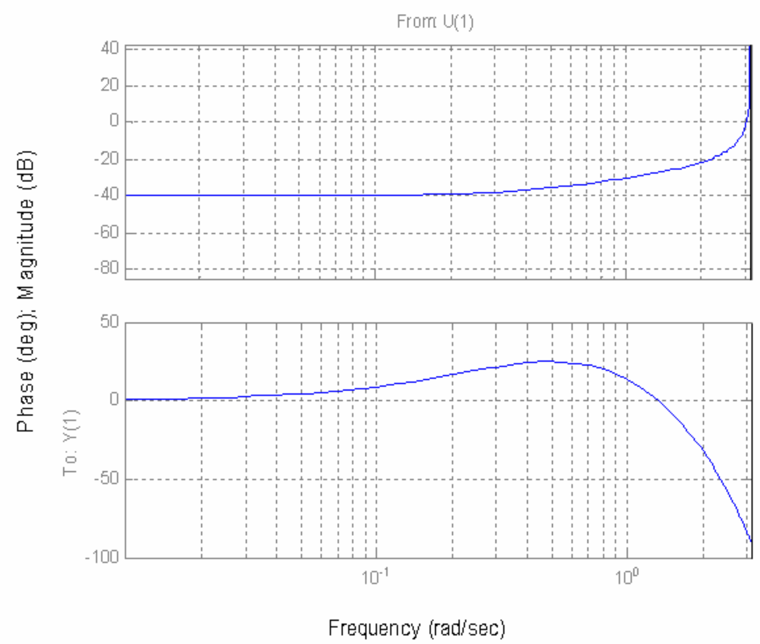
The best possible MV index under normal operating conditions is approximately 0.25, hence it can be assumed that compared to the MV controller, the controller in this loop is poorly tuned.

GMV benchmark - loop 2

The GMV benchmark algorithm needs a set of dynamic error and control weights to compute the performance index. These weights act as design parameters that specify the type of optimal controller required. the user is required to know and specify the optimal performance requirements for the control loop under assessment.



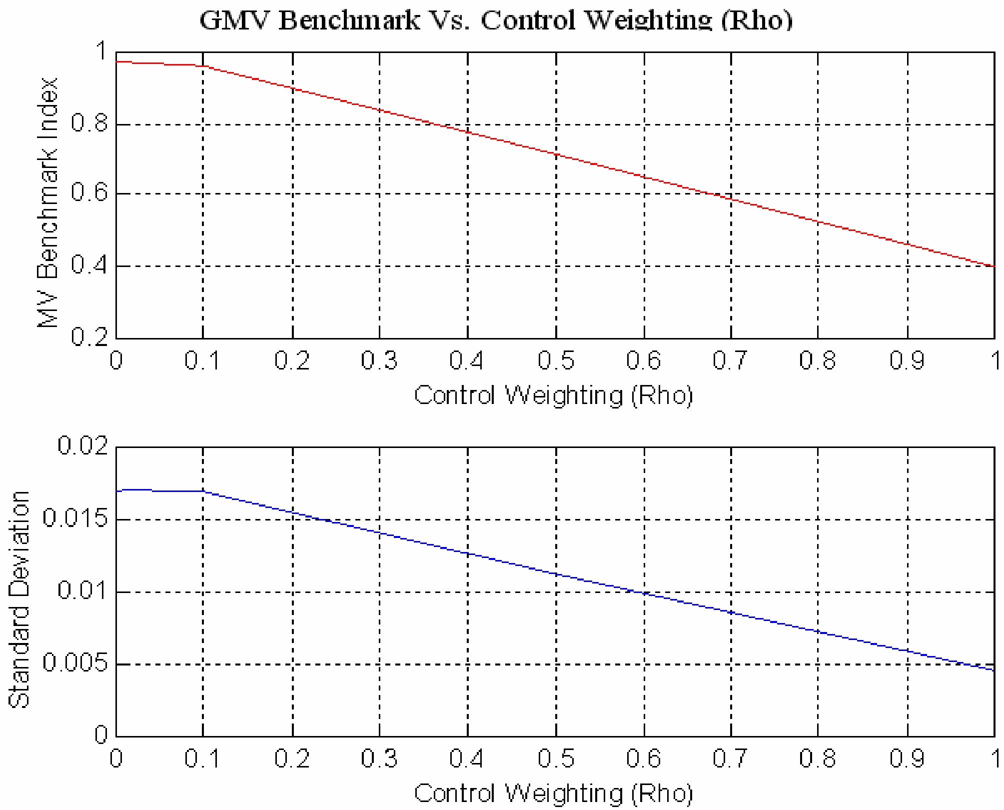
Error weighting



Control weighting

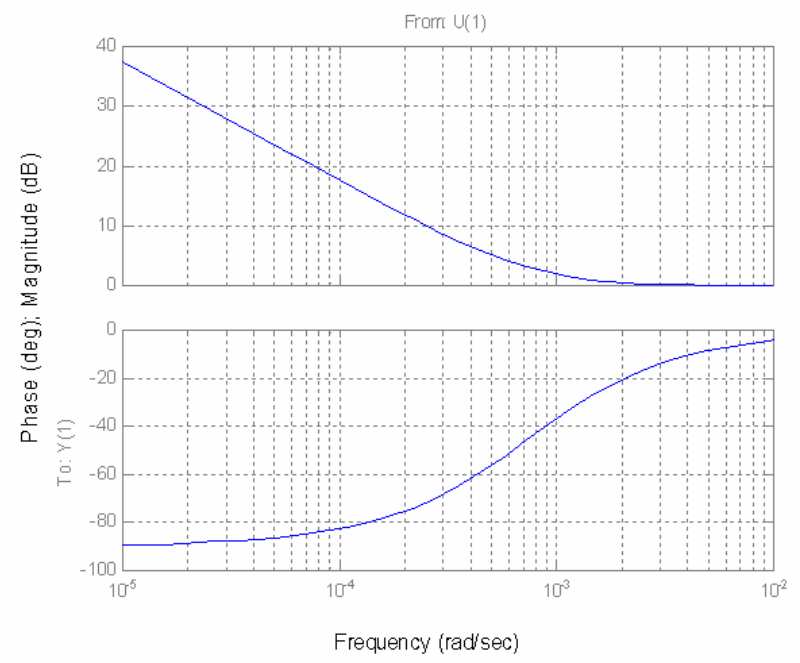
GMV benchmark : loop 2

In order to evaluate the control effort, we use the weighting shown in the previous slide and vary the relative weighting between them. As the weighting of control increases, more penalty is put on control action.

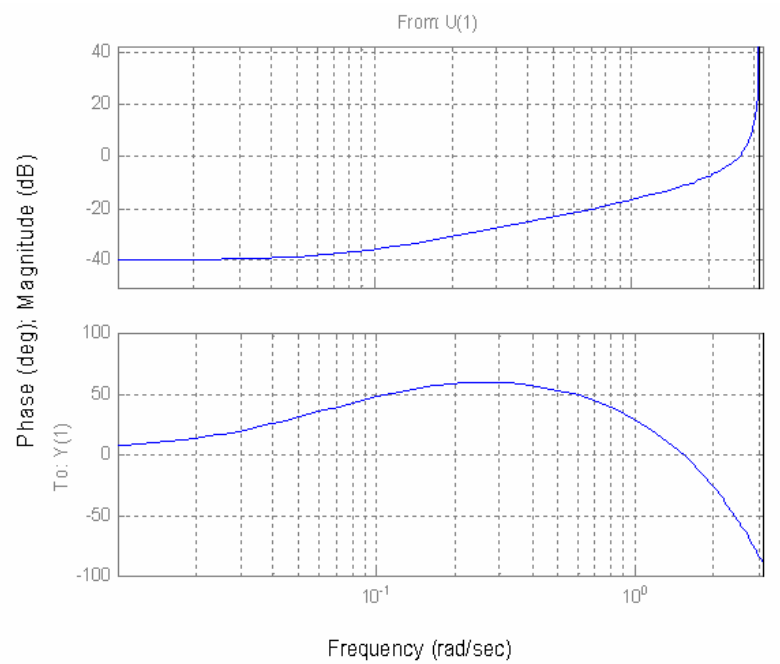


As the value of the scalar term was increased, the performance of controller can be seen to depreciate as indicated by the benchmark index. The controller is indeed a MV controller and it may be using too much control action.

GMV Benchmark - Loop 4

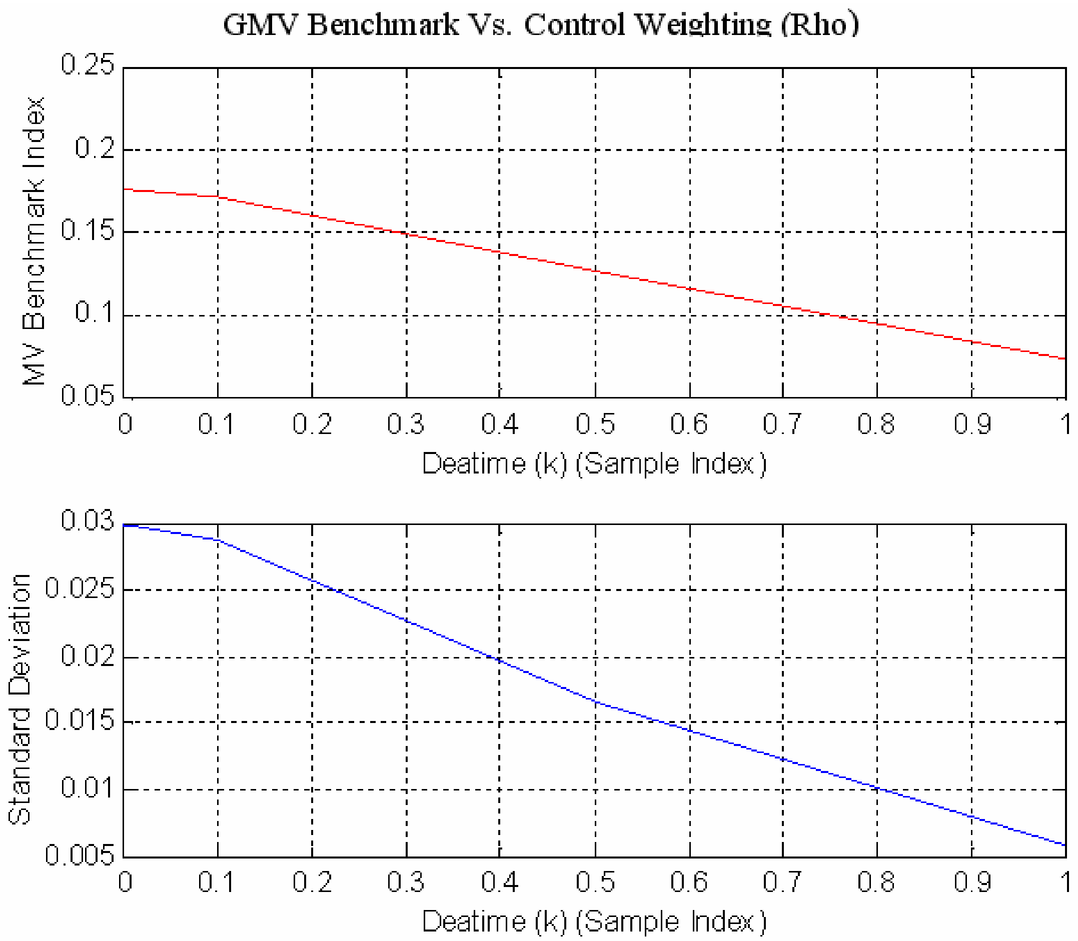


Error weighting



Control weighting

GMV Benchmark - Loop 4



In term of GMV metric, loop 4 still under-performed, it is likely that it should be re-tuned.

RS-LQG Benchmark

The RS-LQG algorithm does not use plant data to compute the benchmark index, a process model in transfer function form is required

1. Existing controller and RS controller type required (this information normally already exist).
2. Models of the system disturbance and reference in transfer function format are also required
3. The accuracy of the results returned ultimately depends on the accuracy of the model used for benchmarking

RS- LQG Benchmark

The RS-LQG algorithm requires the user to specify dynamic error and control weightings

1. Weightings determine the desired optimal controller required
2. Difficult to compare the performance results returned for the same process control loop when two different set of weightings are used
3. The choice of weightings must be consistent with the control problem, especially for RS benchmarking

In the following slides, the effect of weighting on RS-LQG will be illustrated.

RS-LQG Loop 4

Using system identification, we have:

Plant $\left(\dfrac{B_0}{A_0}\right)$ $$z^{-1}\frac{-11630+4510\,z^{-1}+5301\,z^{-2}}{1-0.6757\,z^{-1}-0.4152\,z^{-2}+0.0947\,z^{-3}+0.0009064\,z^{-4}}$$

Noise $\left(\dfrac{C_d}{A_d}\right)$ $$\frac{.005}{1-.4z^{-1}}$$

Original controller: C= -2e-6

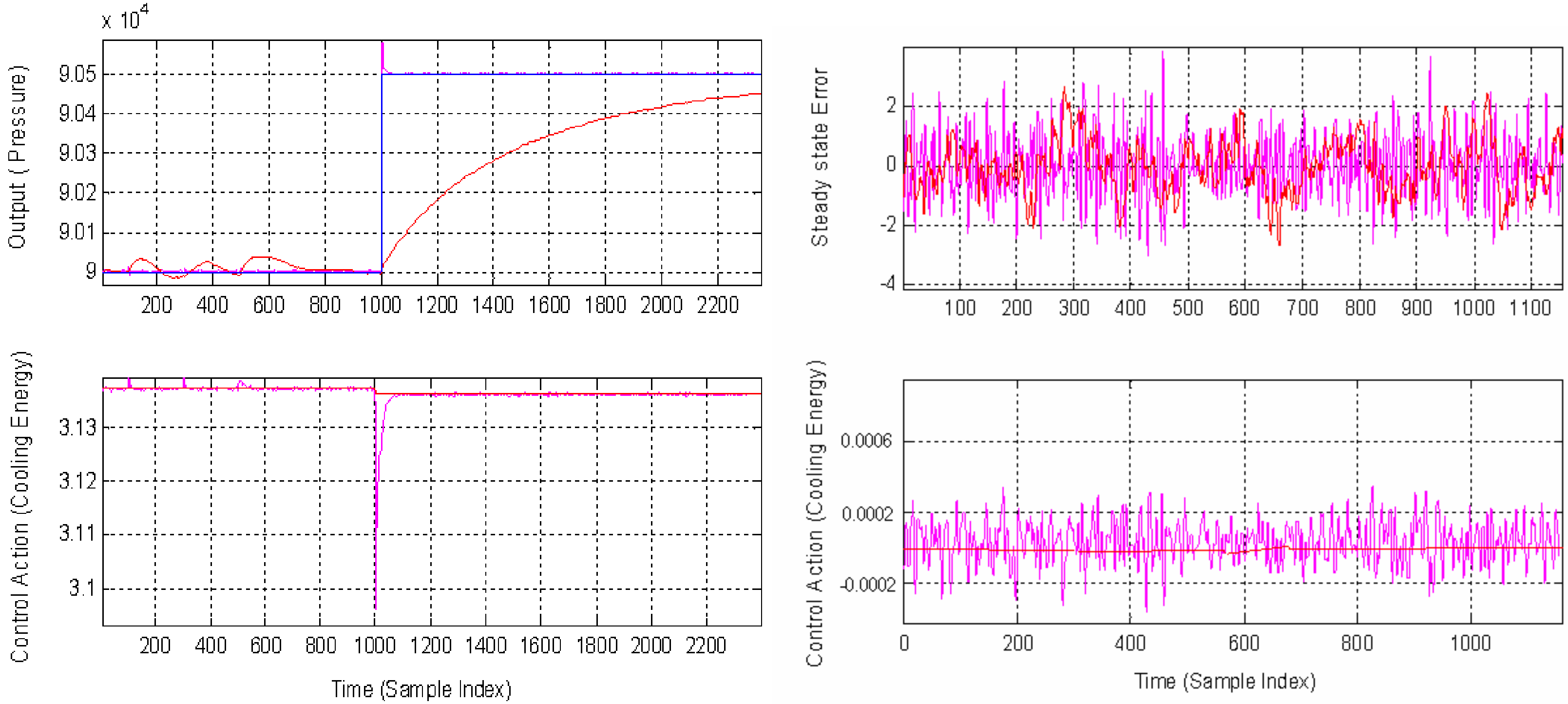
First, the error and control weighting are selected as:

Error weighting $Q_c = \dfrac{1.002-0.9985\,z^{-1}}{1-z^{-1}}$

Control weighting $R_c = 0.1$

RS-LQG Benchmark - Loop 4

Red: existing Controller – Violet: re-tuned controller



Dynamics response

Steady state response

RS-LQG Benchmark - Loop 4

From the previous slide, it can be seen that the re- tuned RS-LQG has a much faster dynamic response than the original controller.

On the other hand, in term of regulation performance, the original controller is better. The control action is much less than that of the RS-LQG controller. The variance of the output is also smaller. It should be noted that the steady state results shown is after de-trending.

Now, the we change the RS-LQG weighting as follows:

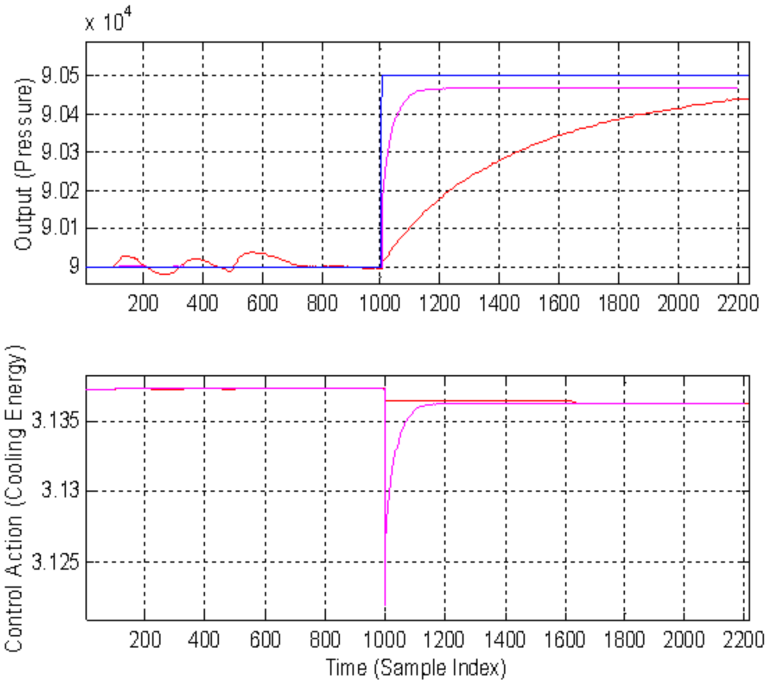
Error weighting $Q_c=1$ Control weighting $R_c= 0$

The noise is the same as before:

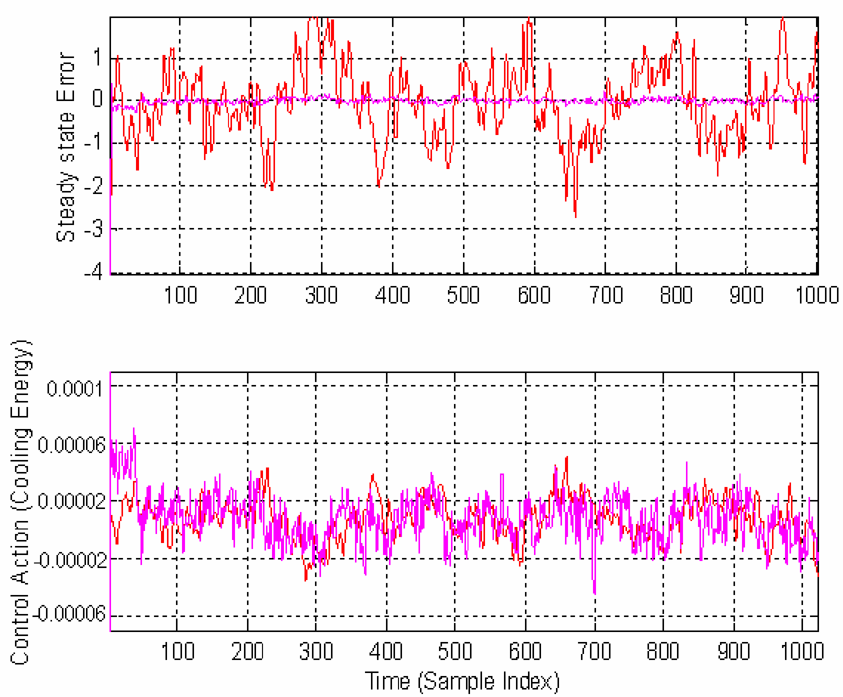
$$\text{Noise}\left(\frac{C_d}{A_d}\right) \quad \frac{.005}{1-.4z^{-1}}$$

RS-LQG Benchmark - Loop 4

Red: existing Controller – Violet: re-tuned controller



Dynamics response



Steady state response

RS-LQG Benchmark - Loop 4

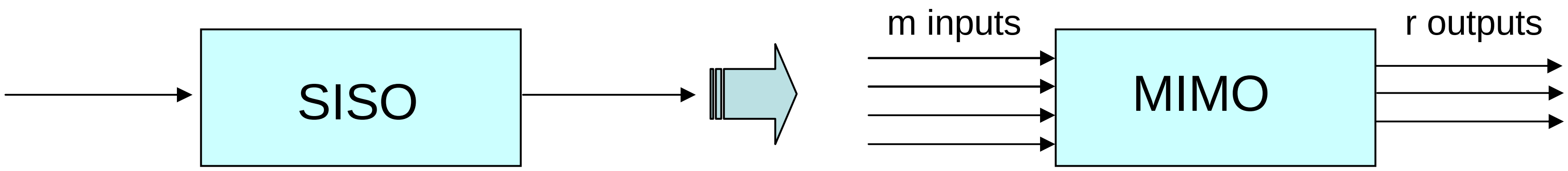
Now, the re-tuned RS-LQG controller has better performance than the original controller both in terms of dynamic response and regulation performance. In the case of the regulation performance, the new controller has greatly reduced the output variance, with a modest increase on the control effort. Since there is no integral in the error weighting, the new controller is also a proportional controller. However, the gain is greatly increased.

This clearly illustrate that the weighting selection for the RS-LQG design plays a deciding role on its success.

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:

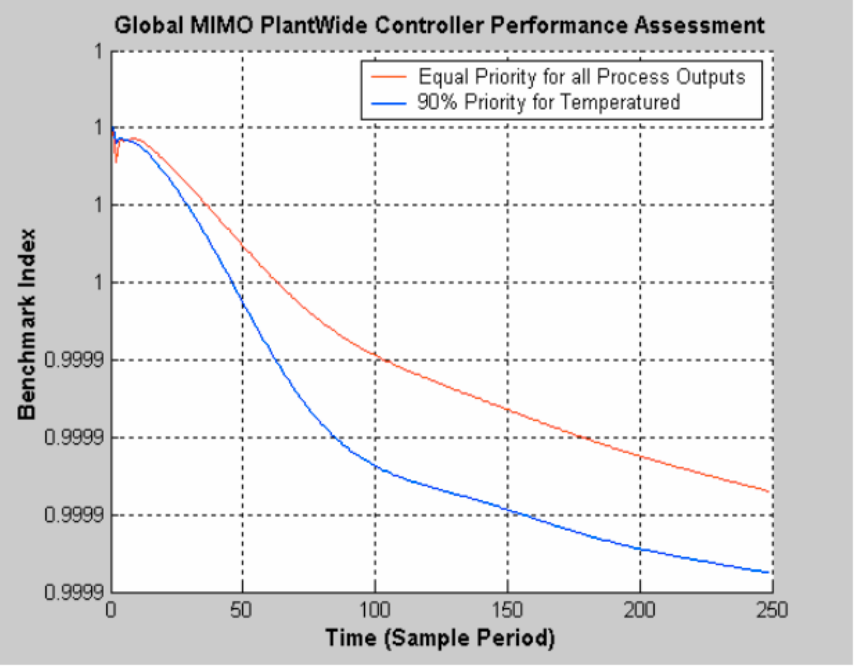
- interactions between loops
- loops need to be prioritised to obtain desired objective
- performance is also dependent on control structure

LQGPC benchmark

- The LQGPC algorithm does not use plant data to compute the benchmark index, a process model in state space form is required
- Existing controller model is required
- Models of the system disturbance are required
- Future reference trajectory is assumed to be given
- The accuracy of the results returned ultimately depends on the accuracy of the model used for benchmarking

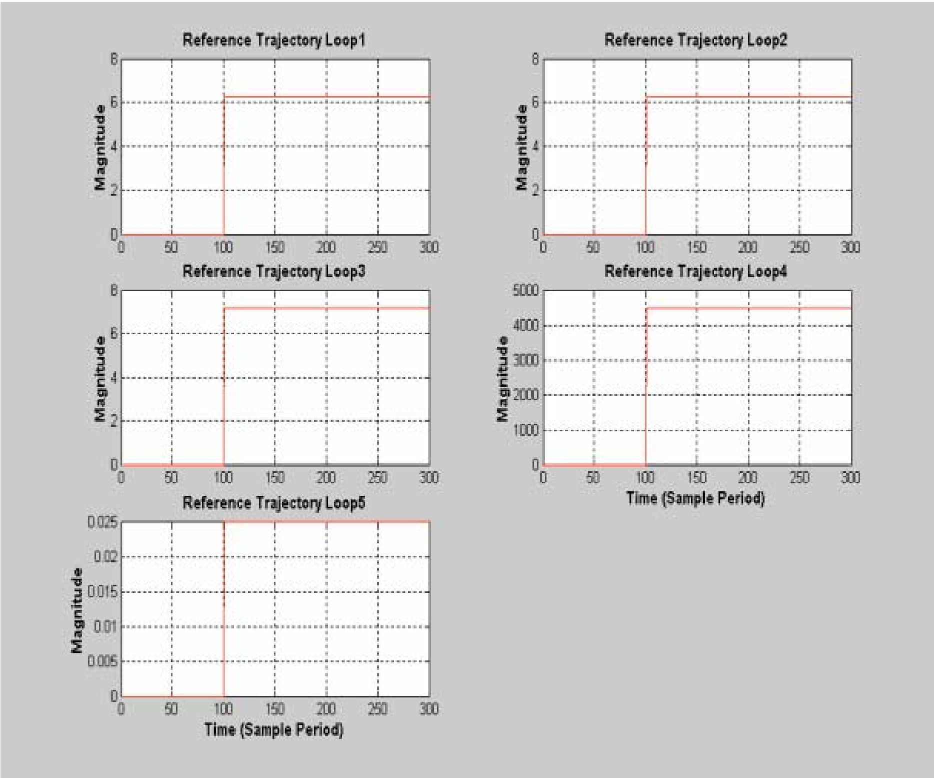
Benchmarking Results - Steady State

The control weighting is set to be zero. The LQGPC benchmark tries to minimise weighted output variance. Two error weightings are used. The result indicates that the original PID controller is operating as a MIMO MV controller



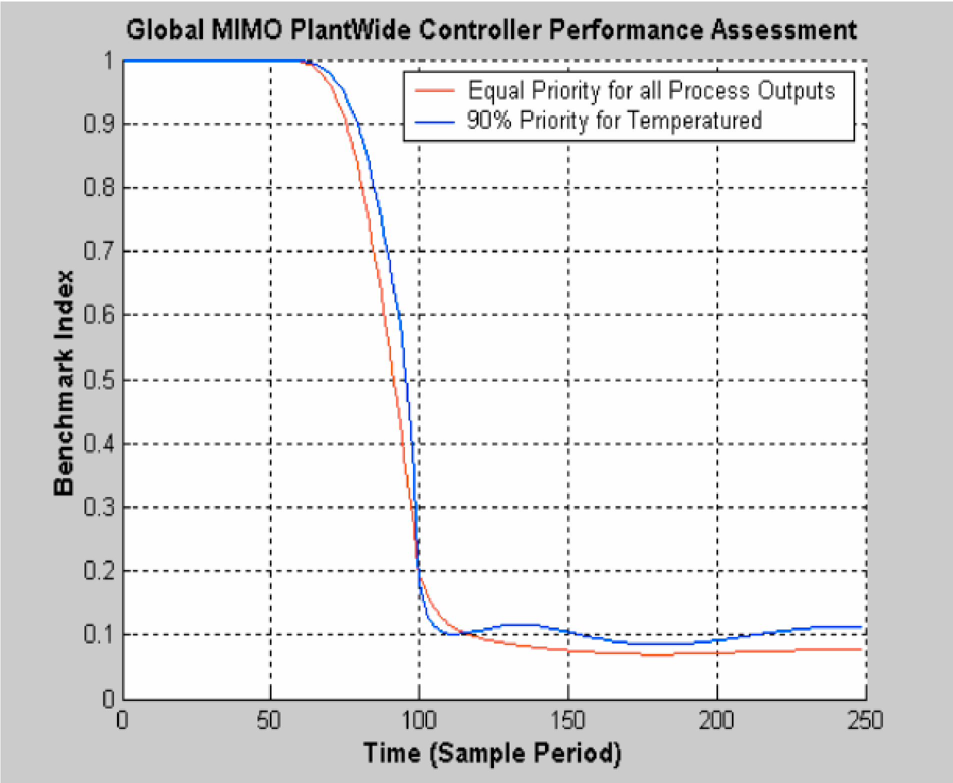
LQGPC - Transient Performance

For the following reference changes:

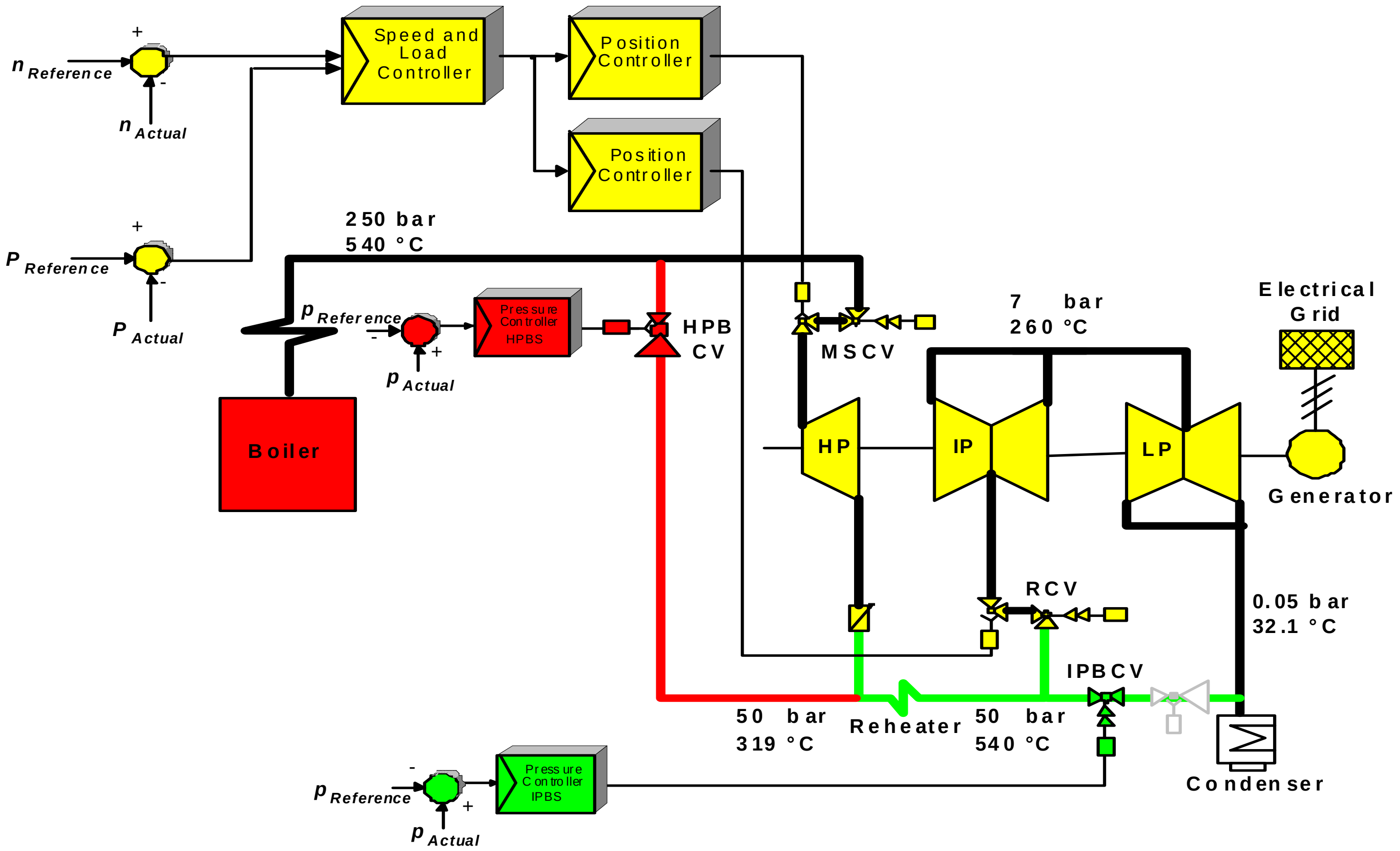


LQGPC - Transient Performance

The transient performance test indicates that the multi-loop PID controller is operating at only 10% of the GPC optimum.



Steam Turbine



RS-LQG Benchmark

As discussed before, in the case of a steam turbine, dynamic performance is the major concern.

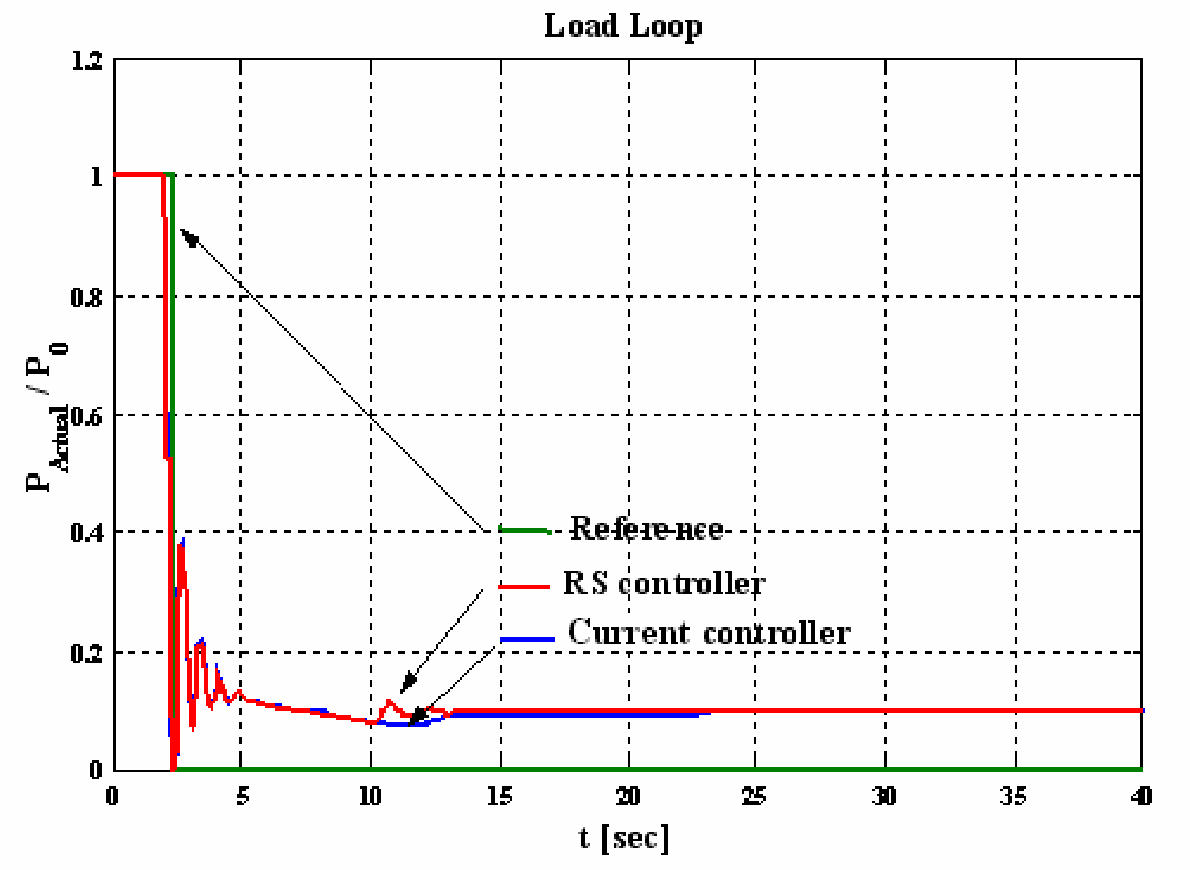
The stochastic type of benchmark is not useful in this case. Only RS-LQG and LQGPC type of benchmarking will be discussed.

Since both of them are model based, it can be considered as a controller retuning exercise.

In the following, we will only present results from tuning.

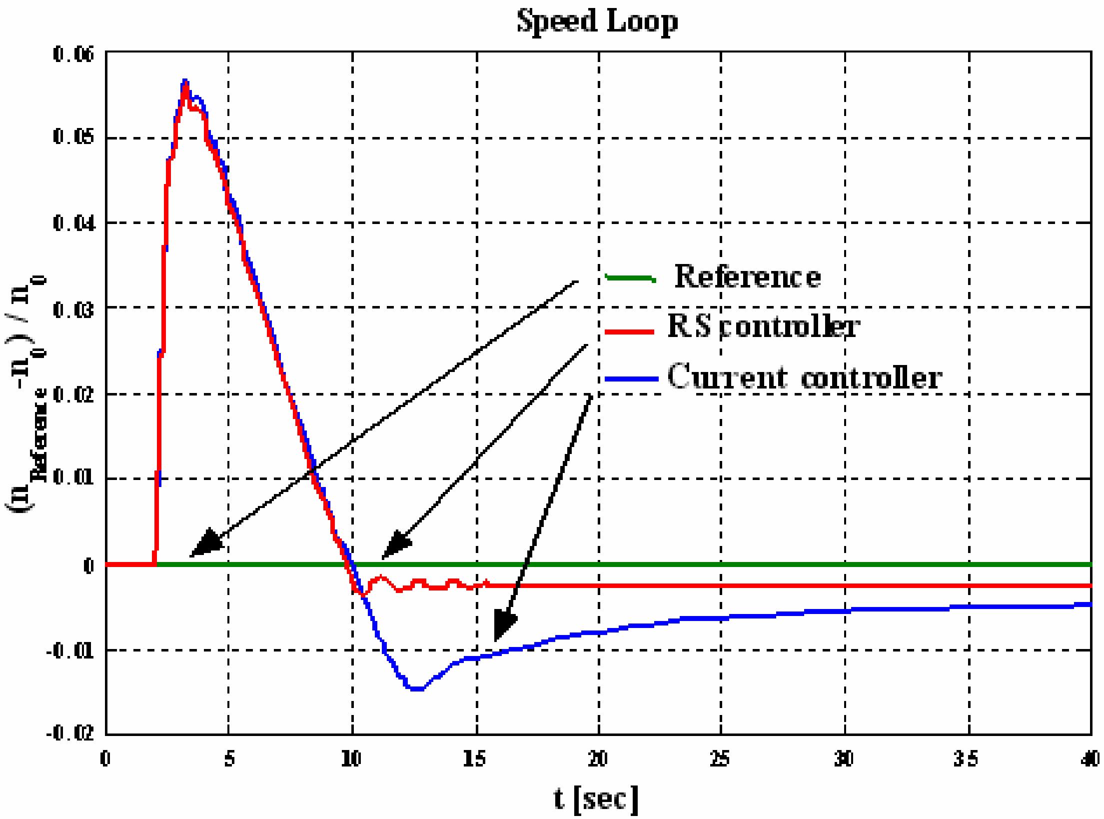
RS-LQG Result

Load Rejection 100% to 10%



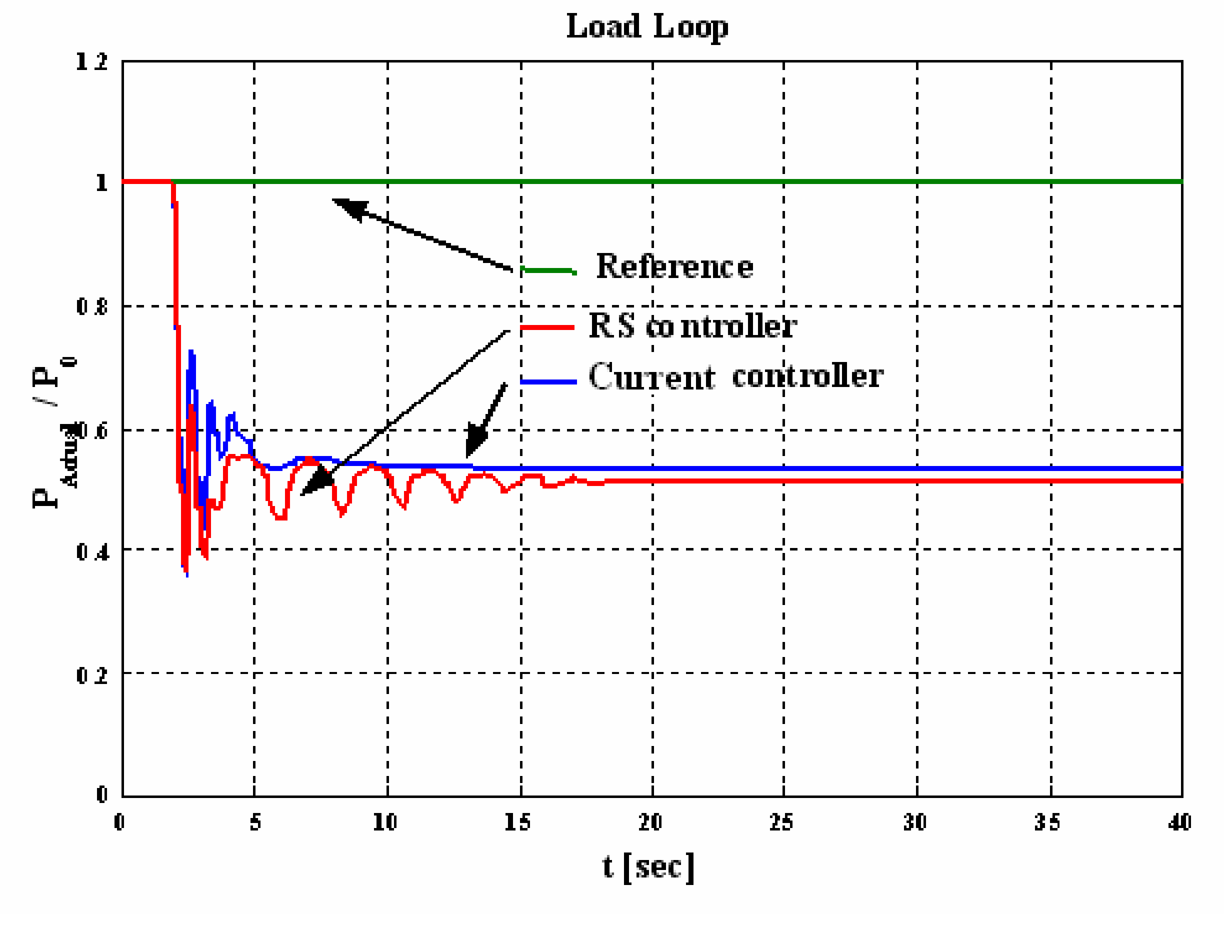
RS-LQG Result

Load Rejection 100% to 10%



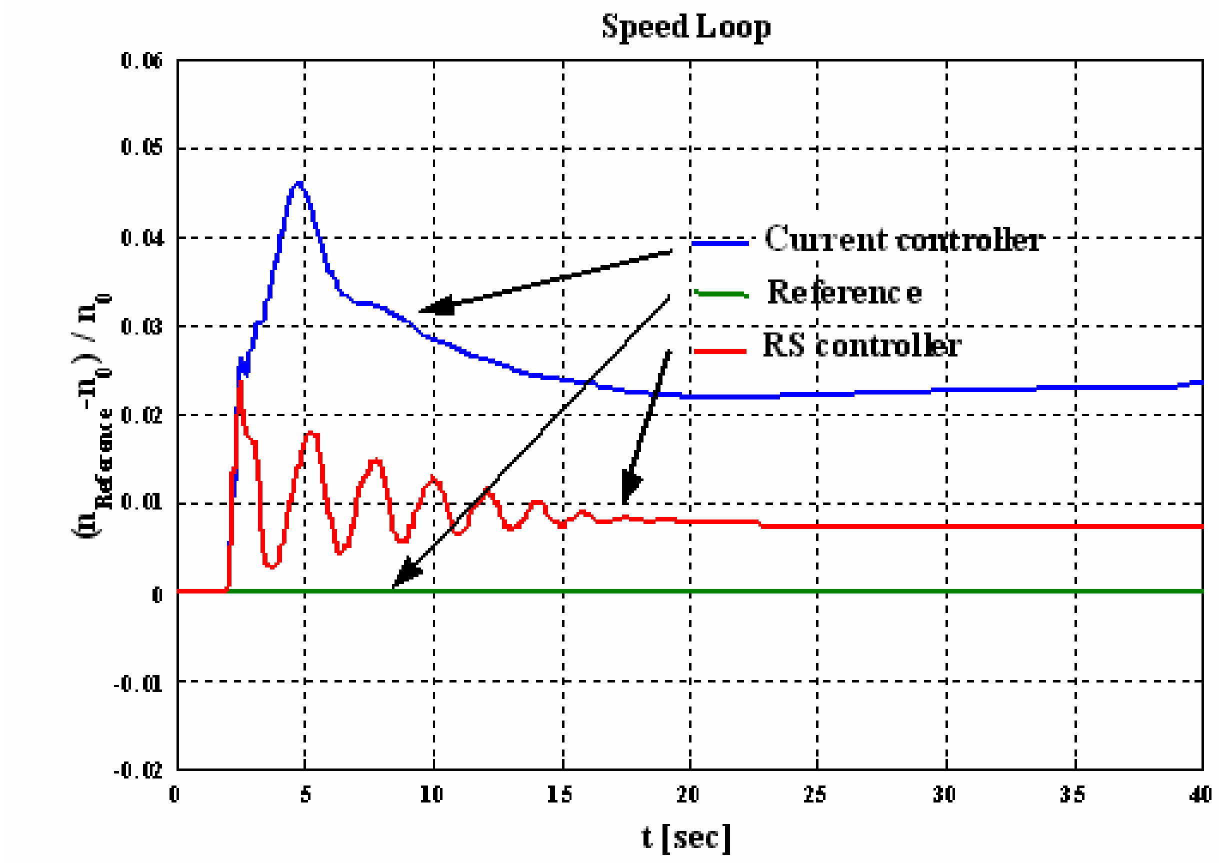
RS-LQG Result

Load Rejection 100% to 50%



RS-LQG Result

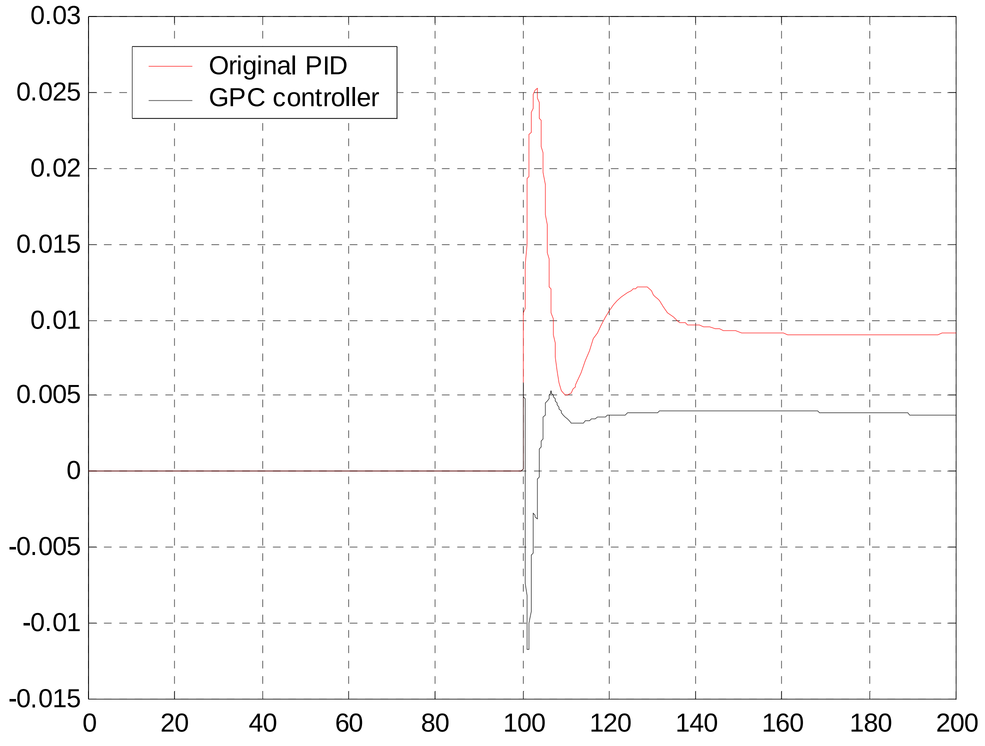
Load Rejection 100% to 50%



LQGPC Result

Load Rejection 100% to 80%

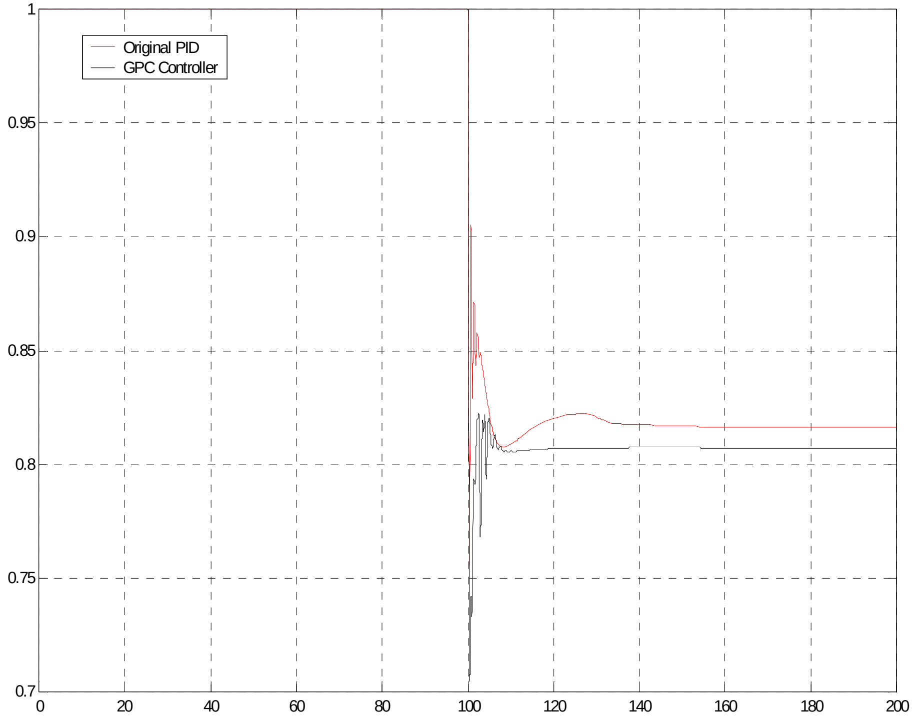
Speed



LQGPC Result

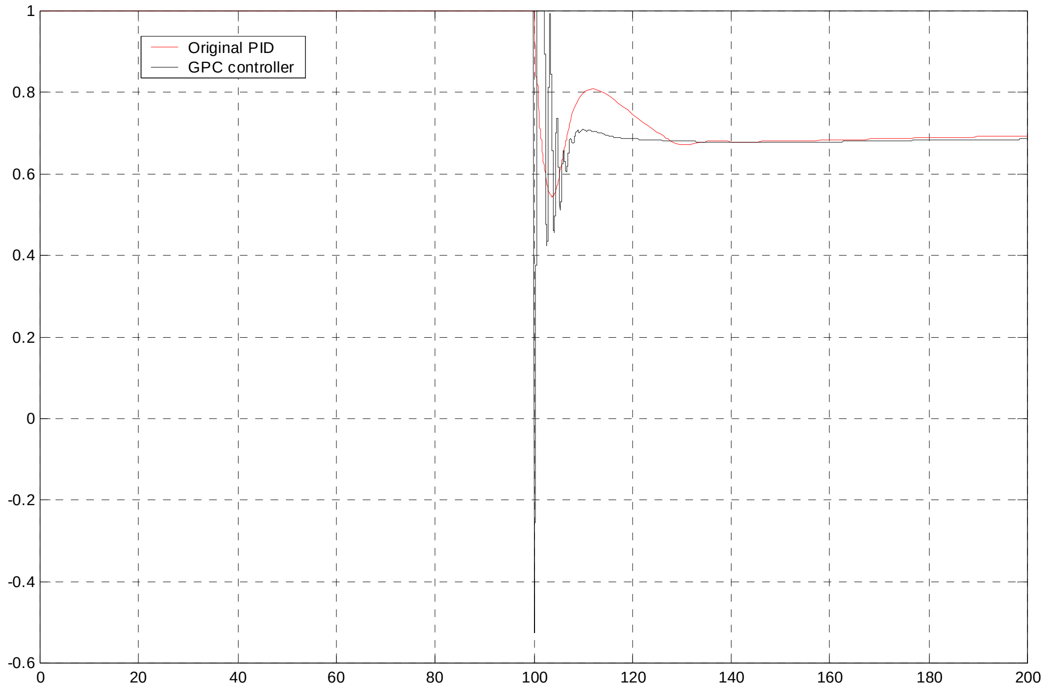
Load Rejection 100% to 80%

Load



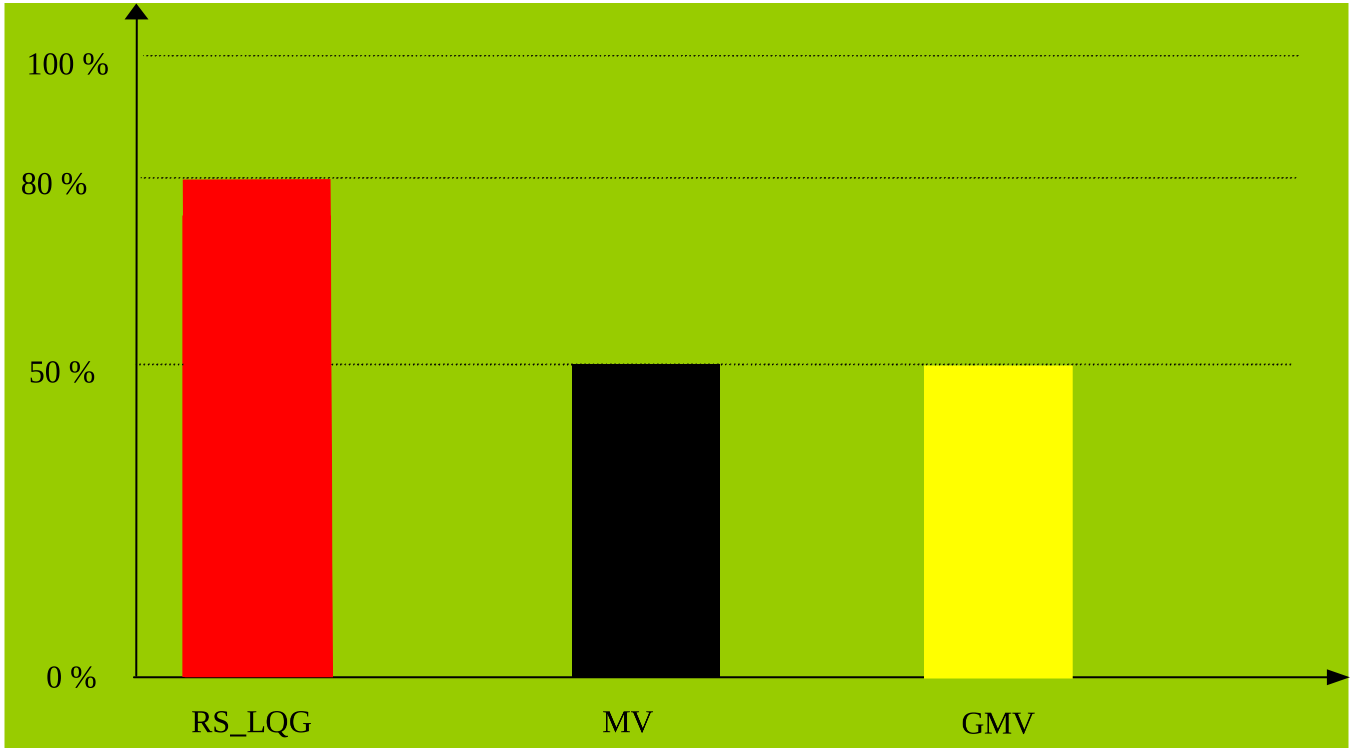
Load Rejection 100% to 80%

Control Valve



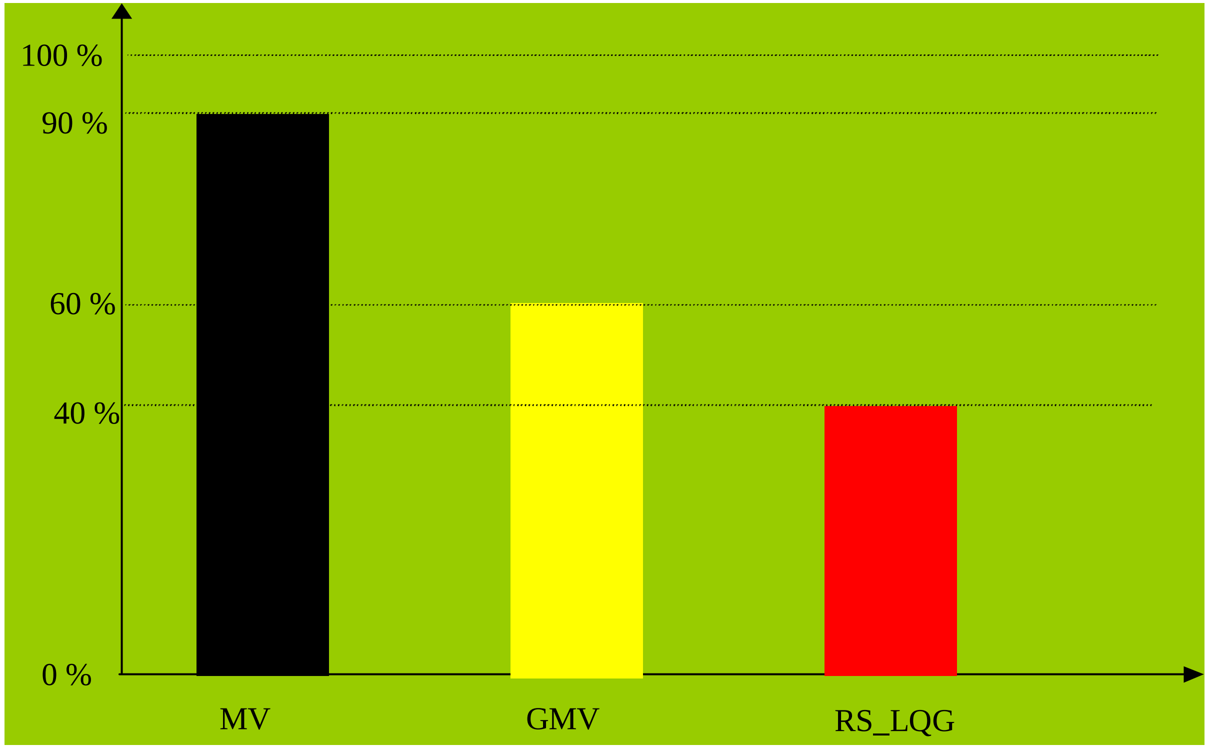
Benchmark Criteria Effectiveness

The performance of a controller in steady state conditions **might be significantly different** from it’s performance in dynamic conditions



Benchmark Computation Effectiveness

How to **translate** gain margins, phase margins, bandwidth, overshoot, rise time, e.t.c **to** dynamic error and control weightings.



Controller Design / Retuning Effectiveness

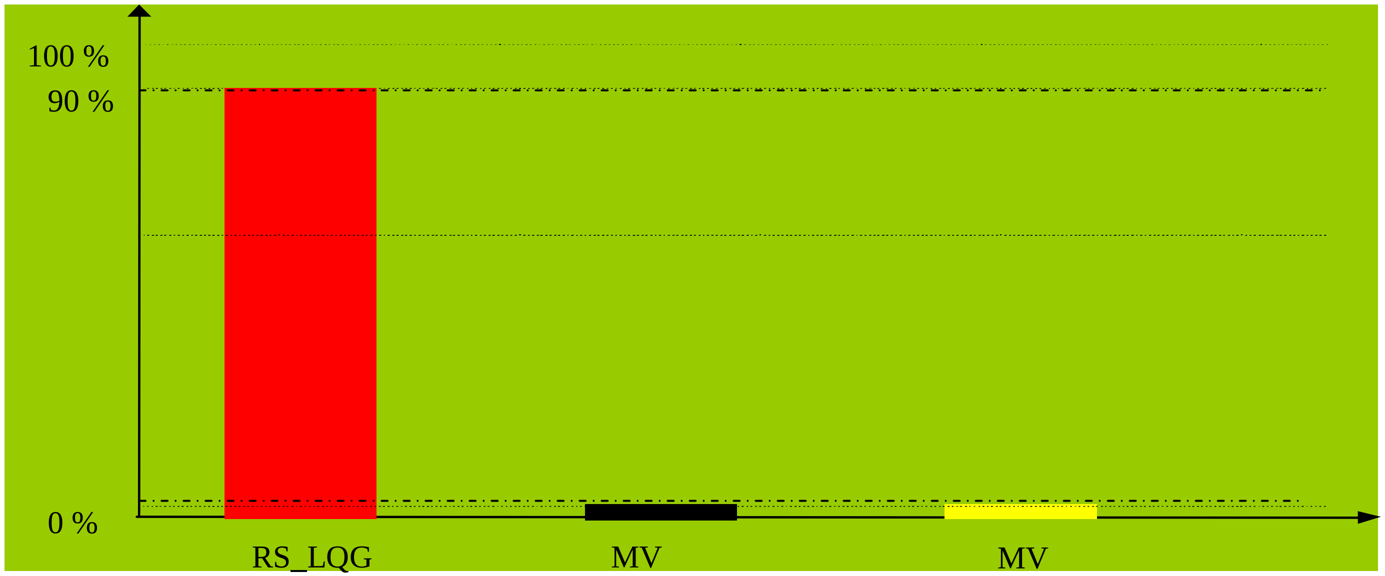
The main question is not how well the system is performing, but can system performance be improved and how?

- The MV and GMV algorithms only give an indication of how well the existing controller is performing
- No information is provided on how the controller can be re-tuned to obtain that performance
- No information is provided that could aid controller re-design
- The RS_LQG algorithm gives an indication of how well the existing controller is performing as well as design information
- The RS_LQG algorithm can be used to test different optimal design scenarios

Controller Design / Retuning Effectiveness

MV, GMV, RS_LQG algorithms are defined for SISO systems, but most industrial systems are MIMO in structure

- The performance index is thus the performance of the loop controller if all other process loops are set to manual
- The improvement in the performance in one loop could cause performance degradation in a corresponding loop



Conclusions

- SISO benchmarks provide a wealth of useful information.
- The move from SISO to MIMO algorithms will provide better optimisation targets.
- Engineering judgement is still an essential part of the benchmarking process.
- Define the control/optimisation problem, then chose the benchmark tool that best fits best.