Wannabe-MPC for Large Systems Based on Multiple Iterative PI Controllers

Pasi Airikka, Mats Friman

Metso Corp., Finland

17th Nordic Process Control Workshop

Jan 26-27 2012 DTU Denmark

Content

Motivation

Multiple PI (MIPI) controllers

Practicalities

MIPI controller design

Simulation example

Conclusion

Motivation

- PID controllers have a profound position in process control applications
  - Relatively easy to understand them without proper education
  - Available basically in every DCS (Distributed Control System) or PLC (Programmable LogiC) as a function block
- MPC based controllers are obviously more seldom used
  - Requires a specific function block for applicability and usage
  - Still rather tedious to design (modelling) and to tune properly
- Could PI(D) controllers be organized to work like a multivariable MPC? Could they act as wannabe-MPC's without requiring
  - Cumbersome tuning based on cost function with weights
  - Optimizer for solving minimization

metso

Content

Motivation

Multiple PI (MIPI) controllers

Practicalities

MIPI controller design

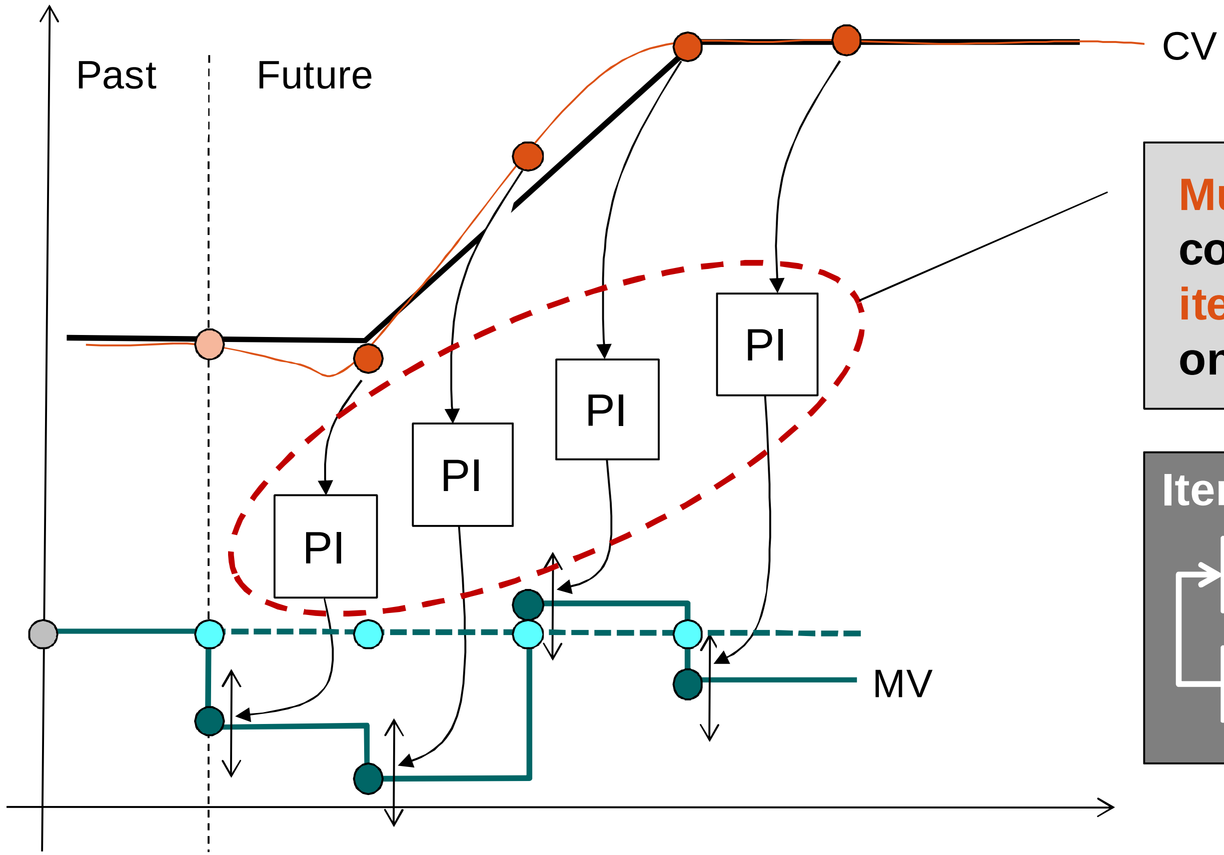
Simulation example

Conclusion

metso

Multiple PI (MIPI) controllers

Principle for one CV



Multiple PI controllers are iterated during one control cycle

Iterations

Simulate CV

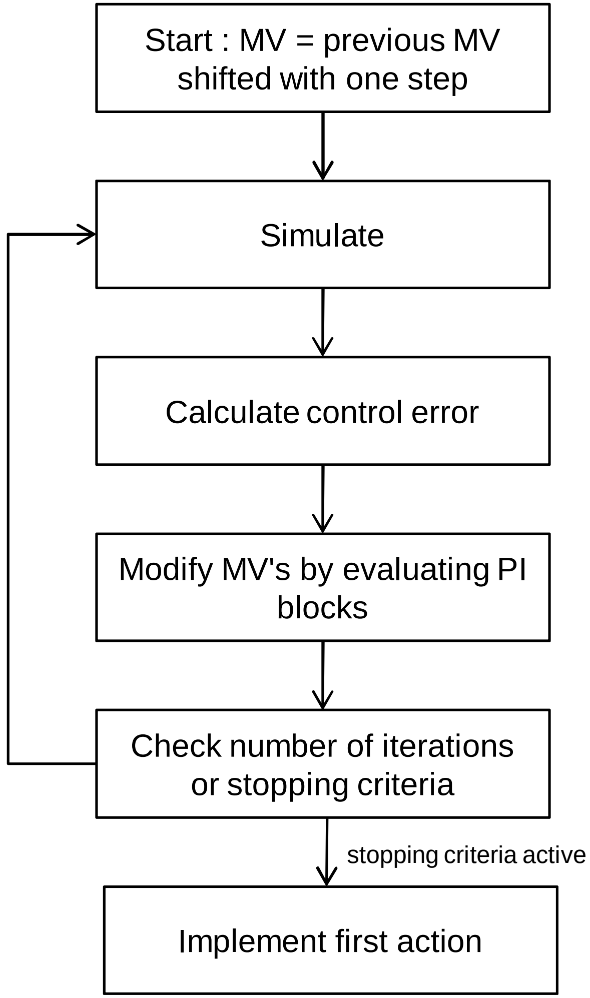
Modify MV

For each CV (Control Variable) trajectory, there are a pre-designed number *N* of single PI controllers working for minimizing the future control error and giving the MV (Manipulated Variable) trajectory as an outcome.

Multiple PI (MIPI) controllers

Principle for one CV

1. Future CV trajectory is calculated using an existing process model.
2. A pre-designed number *N* of PI controllers for the CV trajectory are placed.
3. Each PI controller is assigned to a pre-selected CV trajectory point having
   a. Targeted CV trajectory point as setpoint
   b. Predicted CV trajectory point as measurement
   c. MV trajectory point as controller output
4. Each of the MV trajectory points (*N*) is calculated using a single PI controller assigned to that particular point.
5. Previous step is repeated until reaching the stopping criterion (i.e. control error < *limit* or nr. of iterations < *nr*).
6. The first MV trajectory point is implemented and after receiving a new measurement the iterations (steps 4-5) will start again.



Content

Motivation

Multiple PI (MIPI) controllers

Practicalities

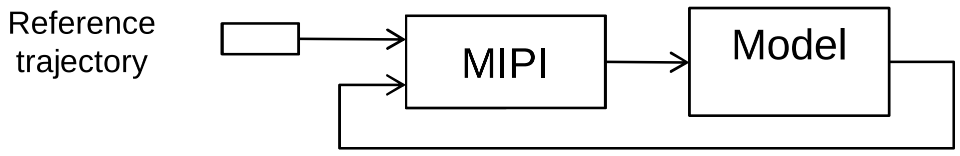
MIPI controller design

Simulation example

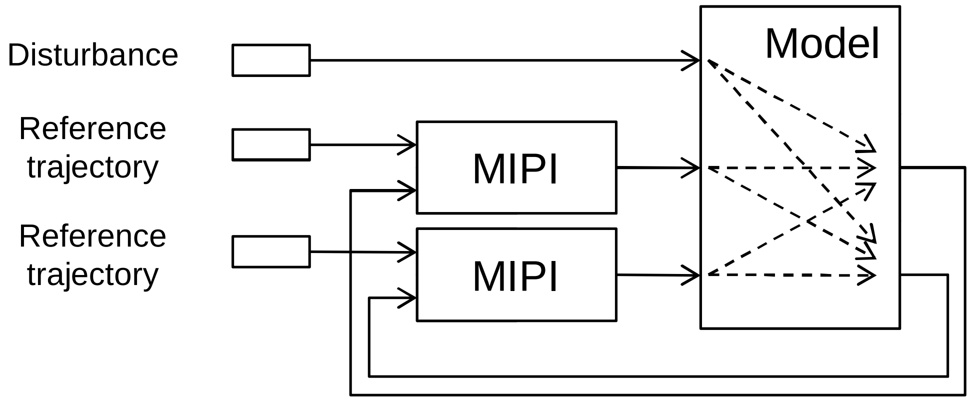
Conclusion

Practicalities
Multivariable control

- Expansion from single-variable (1 x 1) to multivariable (*M* x *M*) case requires *M* x MIPI controllers



Single-Input Single-Output control loop with 1 MIPI controller (*N* PI controllers)

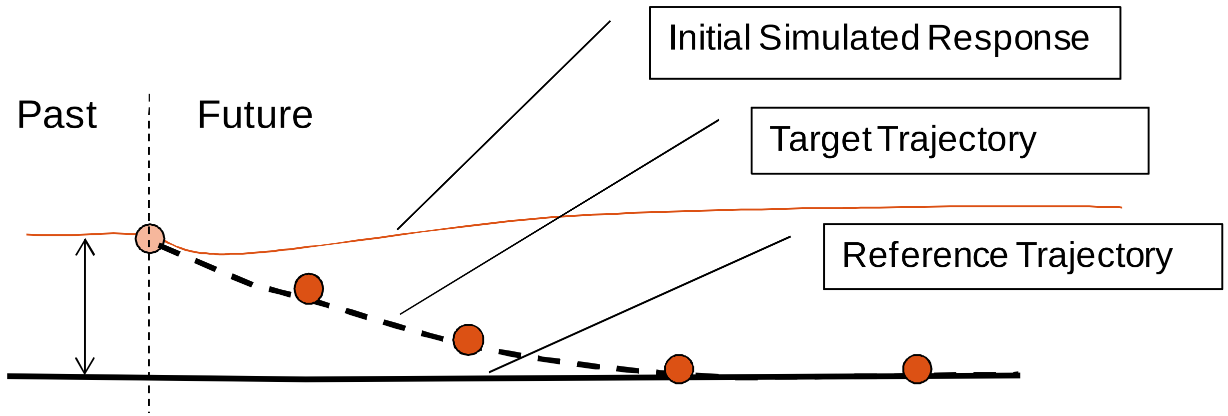


Multi-Input Multi-Output control loop with M MIPI controllers (*M*x*N* PI controllers)

Practicalities

Model mismatch and non-measured disturbances

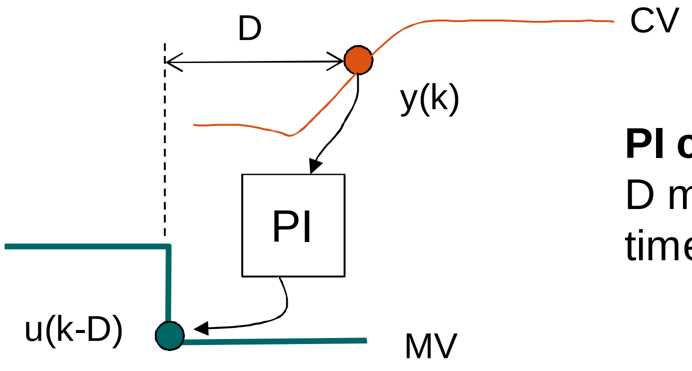
- For dealing with modelling uncertainties and unmeasured disturbances, it is suggested to introduce a target CV trajectory which is the original reference trajectory filtered
  - Filtered CV trajectory (= Target trajectory) is given as a setpoint for a MIPI controller in charge of that particular CV trajectory
  - Low-pass filtering may do the trick



Practicalities

Pairing CV and MV trajectory points

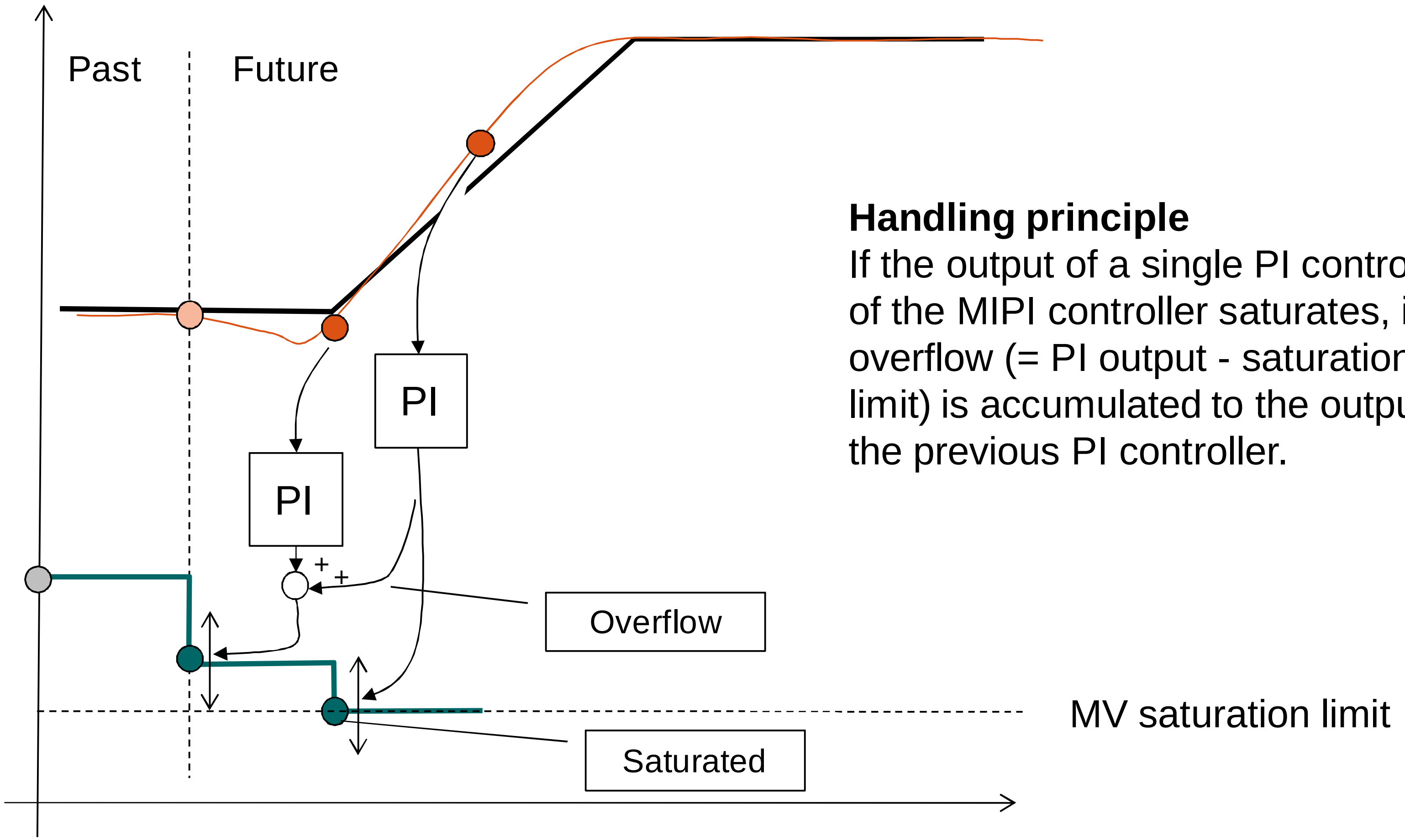
- Each PI controller in a MIPI controller is assigned between pre-selected CV and MV points which are at different time instants
- To have impact from MV to CV, there should be at least a dead time between the selected time instants
  - Dead time value or its estimate must be available



PI controller assignment in MIPI controller
D must be larger than or equal to real dead time.

Practicalities

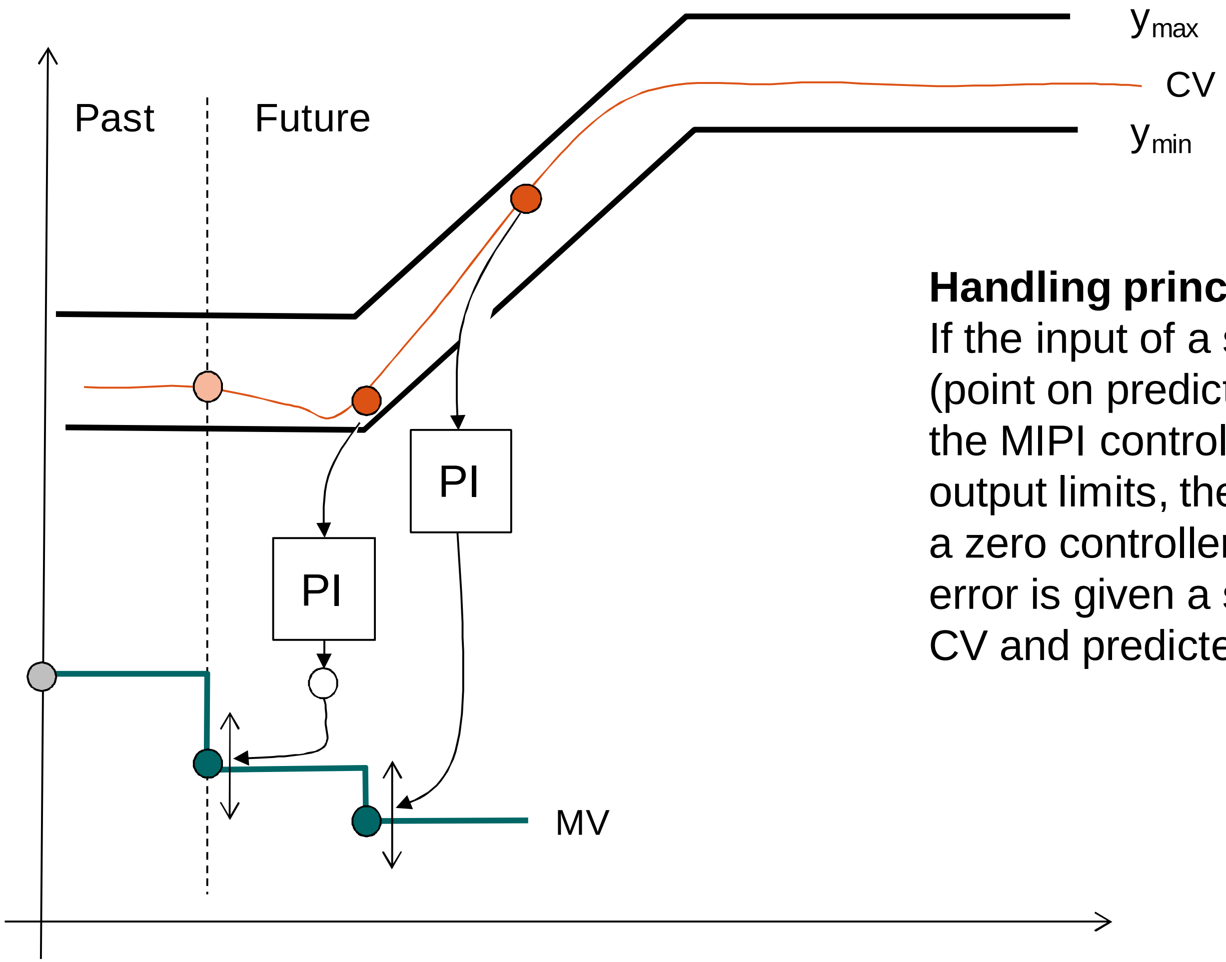
Handling of input constraints



Handling principle
If the output of a single PI controller of the MIPI controller saturates, its overflow (= PI output - saturation limit) is accumulated to the output of the previous PI controller.

Practicalities

Handling of output constraints



Handling principle
If the input of a single PI controller (point on predicted CV trajectory) of the MIPI controller are inside given output limits, the PI controller is given a zero controller error. Otherwise, the error is given a such between target CV and predicted CV point.

Content

metso

MIPI controller design

- Wannabe-MPC requires a process model for predicting CV like any other predictive controller
- Design parameters are
  - Prediction horizon, control period (*h*)
  - Input and output constraints
  - Number of MIPI controllers (*M*) and number of PI controllers in each MIPI (*N*)
  - Iteration criteria: control error limit and/or number of iterations
  - Filter design (or parameter) for filtering target CV trajectories
- No weighting matrices for CV or MV
- It is advised to tune all the PI controllers of a single MIPI controller similarly (autotuning is recommended for simplicity)
- Stability criteria of the proposed Wannabe-MPC is an open issue

Content

metso

Simulation example

Wood-Berry (1973) distillation column model

- Continuous-time multivariable FOPDT transfer function model (2 x 2) with an unmeasured but modelled disturbance
  - Time constant dominating system
  - Rather strong interactions with $K = \begin{bmatrix} 12.8 & -18.9 \\ 6.6 & -19.4 \end{bmatrix}$ resulting in $RGA = \begin{bmatrix} 2.01 & -1.01 \\ -1.01 & 2.01 \end{bmatrix}$

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} \frac{12.8e^{-s}}{16.7s+1} & \frac{-18.9e^{-3s}}{21s+1} \\ \frac{6.6e^{-7s}}{10.9s+1} & \frac{-19.4e^{-3s}}{14.4s+1} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} + \begin{bmatrix} \frac{3.8e^{-8s}}{14.9s+1} \\ \frac{4.9e^{-3s}}{13.2s+1} \end{bmatrix} d$$

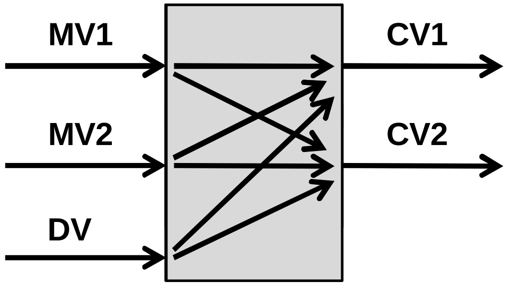
where distillate composition y_1 (%) and bottom composition y_2 (%) are CVs being controlled by MVs reflux flow u_1 (lb/min) and steam flow u_2 (lb/min). The feed flow rate d acts as an unmeasured disturbance.

- MIPI control design:
  - Control period *h* = 1 min, prediction horizon = 10 min
  - 2 MIPI controllers with *N* = 10 PI controllers each
  - Target CV filtering: time constant 40 min for u_1/y_1 and 60 min for u_2/y_2
  - Number of iterations = 11

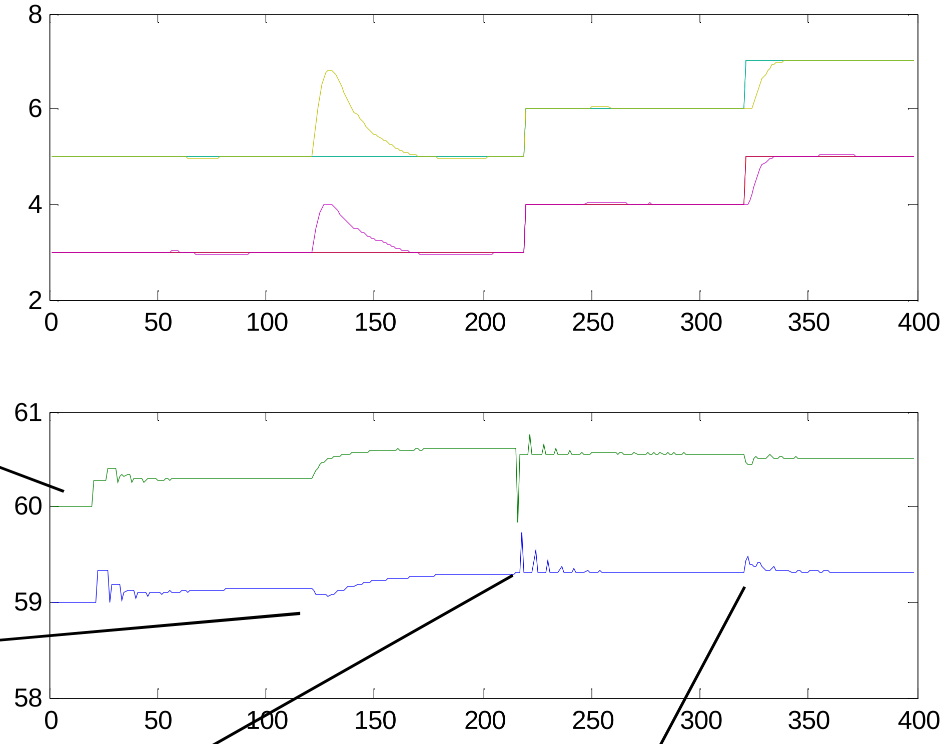
Simulation example

Results with Wannabe-MPC when perfect model

2x2 Model with Interactions and Disturbance



Ideal Simulation with Perfect Model



time = 20
A known disturbance enters system.

time = 120
Same disturbance enters system, but it is unmeasured.

time = 220
A planned set-point change

time = 320
An unexpected set-point change.

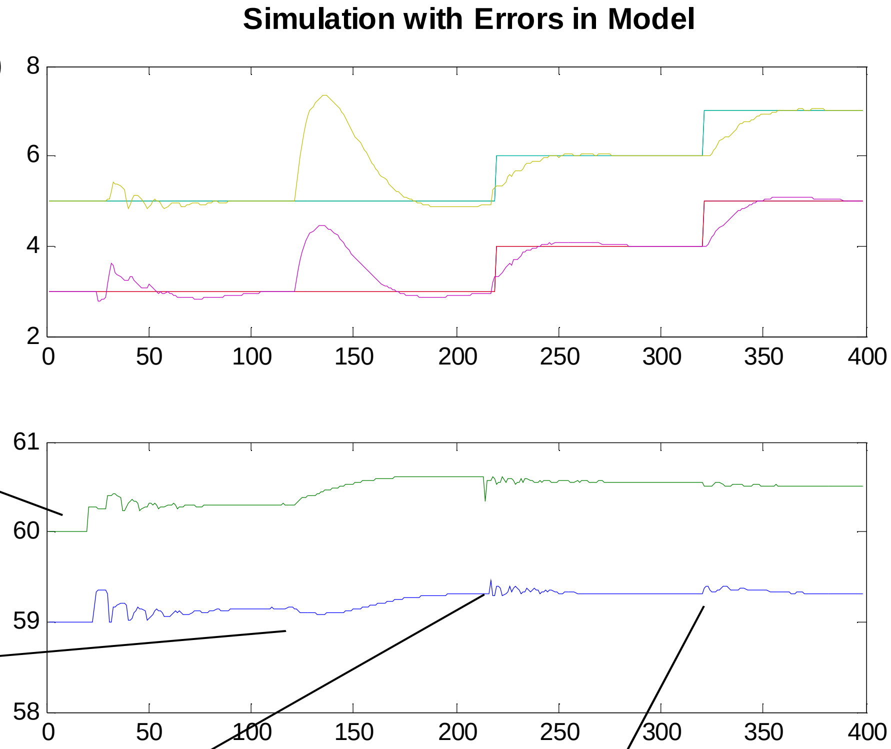
metso

Simulation example

Results with Wannabe-MPC when model mismatch

Errors:
Model Time constants = T/2 (-50% error)
Model Gains = K*1.5 (+50% error)
Model Delays = ceil(L*1.5) (+50% error)

T= simulated time constants of process
K = simulated gains of process
L = simulated delays of process



time = 20
A known disturbance enters system.

time = 120
Same disturbance enters system, but it is not measured.

time = 220
A planned set-point change

time = 320
An unexpected set-point change.

metso

Content

Conclusion

- Wannabe-MPC based on usage of Multiple PI controllers (MIPI) was presented
  - Prediction-based controller
  - Capable of dealing with input/output constraints
  - Suitable for multivariable and complex (non-linear) control systems
  - Does not require optimizer or design of cost function with weights
  - Fewer tuning parameters compared to MPC
- Stability is an uninvestigated issue
- No clear guidelines for filtering parameters