Friday-seminar:

Delay Compensated Feedback-based Autoscaling of Cloud Applications

Manfred Dellkrantz

Automatic Control LTH
Lund University
2016-12-16

Outline

Cloud (Application) Introduction

The Autoscaling Problem

Delay Compensation

`mandel` Testbed

Robustness & Convergence

Model Adaptivity

Cloud is a Business Model

Cloud is about renting computing resources in some form.

- (Virtualized) Hardware
- Data storage
- Computation
- Application
- ...

No initial investment, only operational costs.

Cloud Application

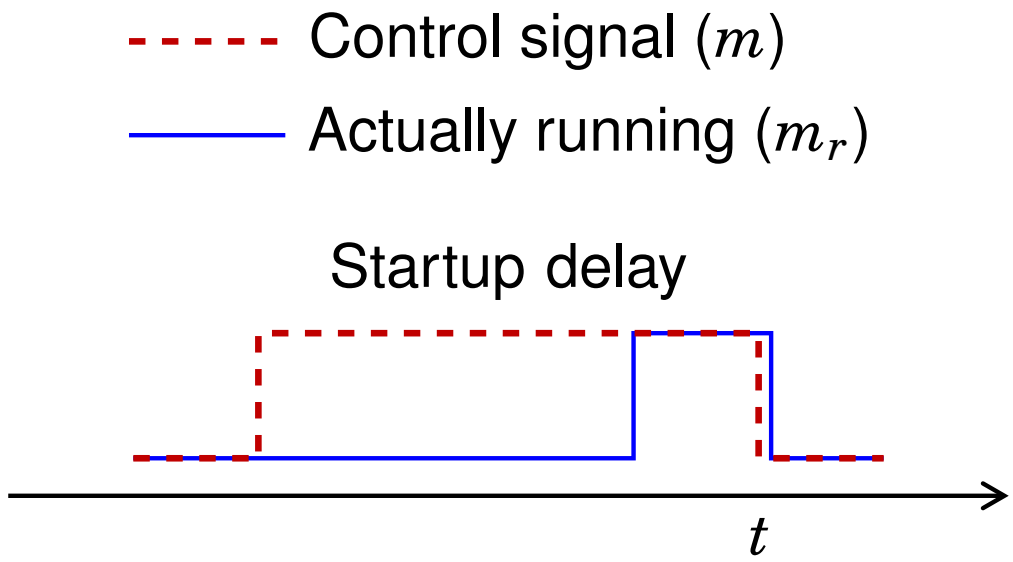
- Some kind of server software
- Runs on rented virtual machines (VMs)
- Handles incoming requests from users

The Autoscaling Problem

- We operate a cloud application
- Minimize rented resources (VMs)
- Maintain high Quality of Service (QoS)
  - Responsiveness
  - Stability
- Handle load variations
  - Daily variations
  - Flash crowds
  - ...

Problem: Starting VMs takes up to minutes. Delays vary heavily.

Startup times as delay



Typical solution

Typical solutions include:

- Slow feedback to avoid problems with delay
- Feedforward from load
- Ignore the delay

Typical solution

Typical solutions include:

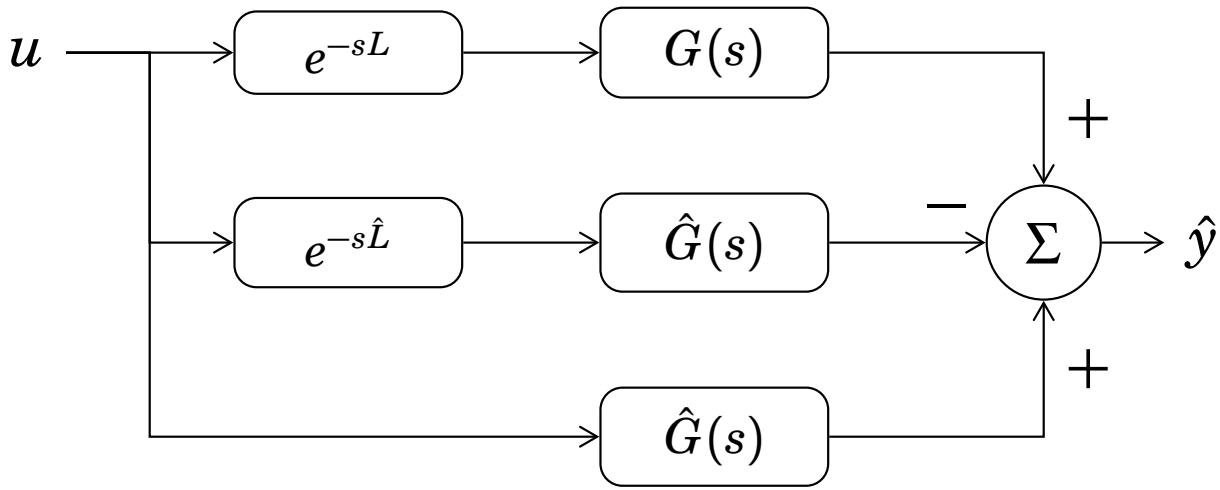
- Slow feedback to avoid problems with delay
- Feedforward from load
- Ignore the delay

Let's do real feedback!

Delay compensation

To use feedback with delays we often use delay compensation.

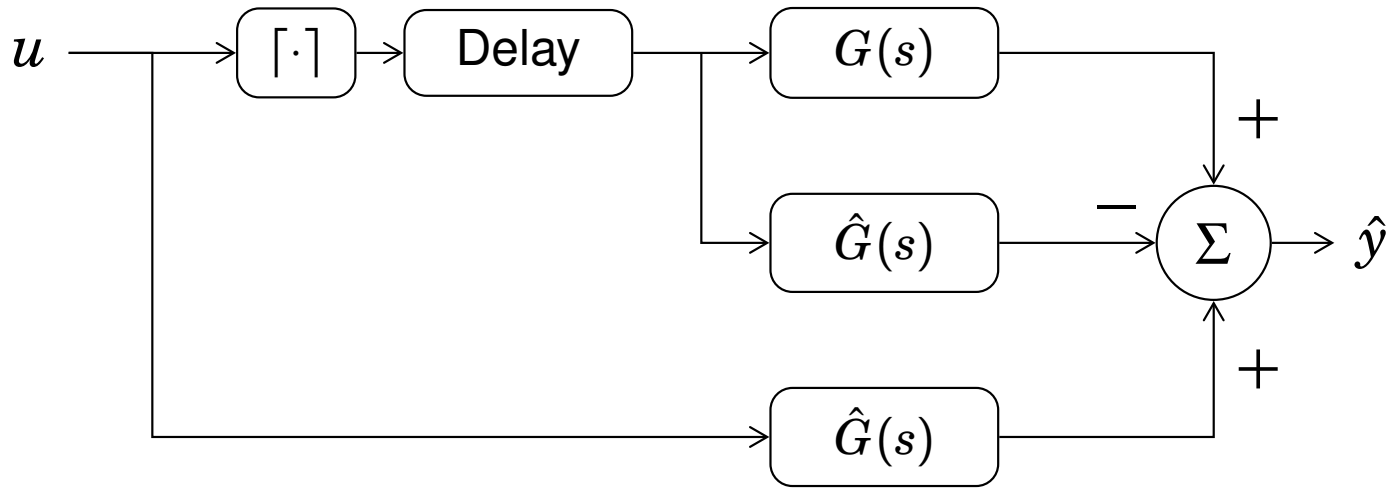
Standard Smith predictor for constant delays.



Delay compensation

To use feedback with delays we often use delay compensation.

Modified delay compensator for varying delays but with measurable delayed signal.

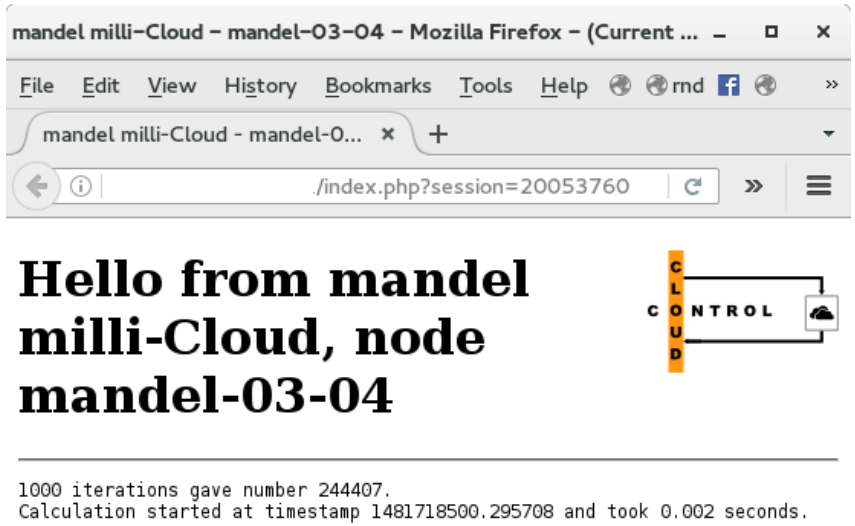


Introducing `mandel`

- Heater (cluster/“cloud”) for the thesis worker room
- 15 computers retired from labs
- 60 cores
- 60 GB RAM
- Gigabit ethernet

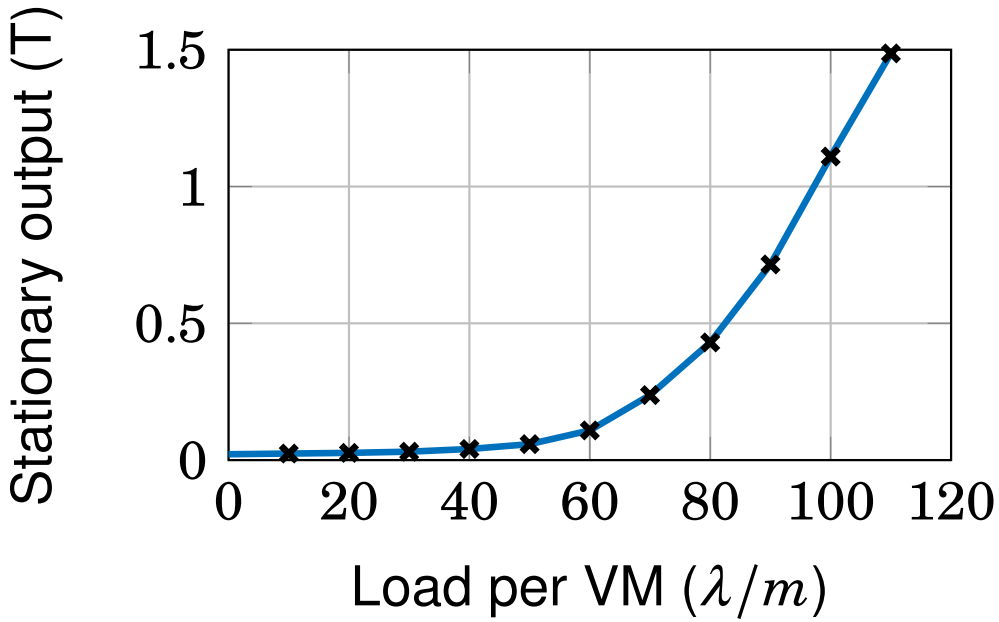
`mandel` application

- Dummy service multiplying random numbers
- Runs 54 VMs
  - 1 generating load
  - 1 balancing load
  - 52 serving requests
- Load (λ) is number of active closed loop clients/users
- Control number of VMs
- Maintain response times T at set point $T_{ref} = 0.25$ s



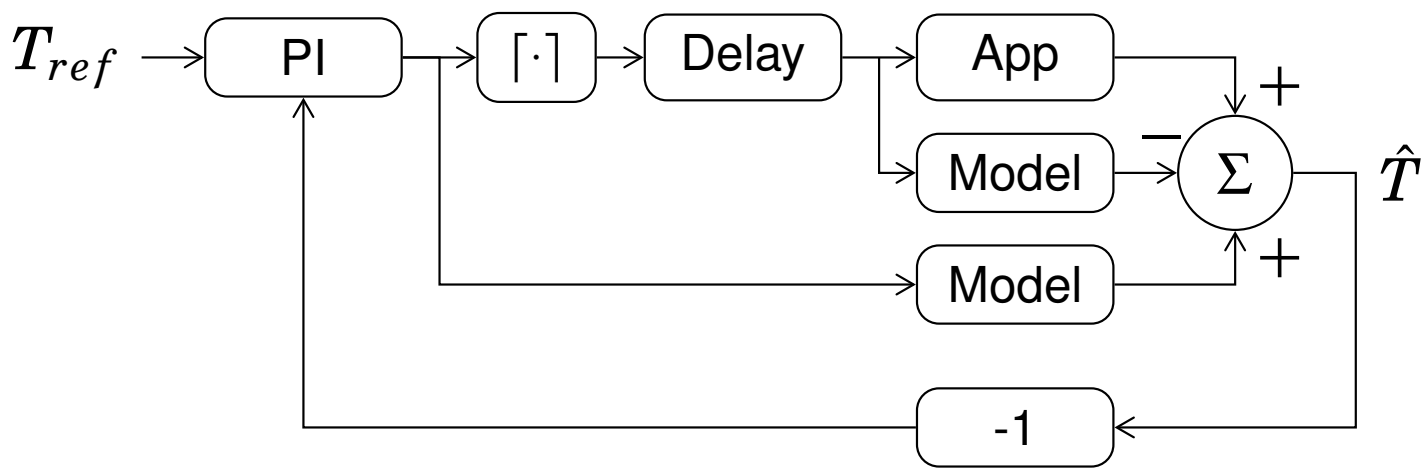
`mandel` model

Approximated well with static nonlinearity and first order filter.



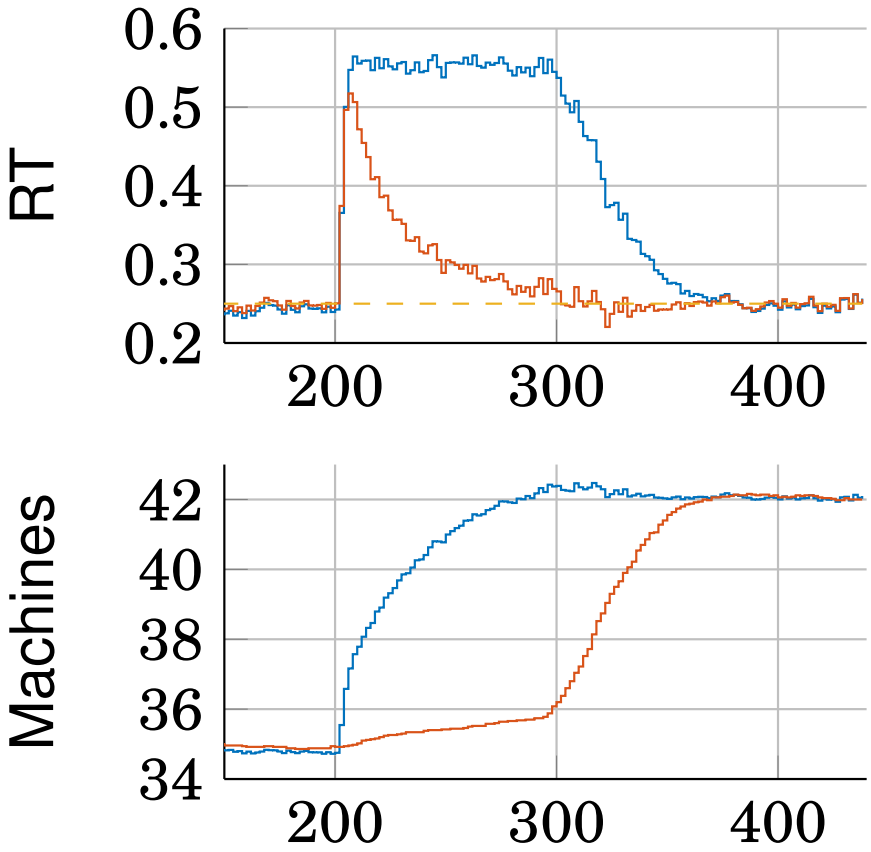
$$H_{\text{filter}}(z) = \frac{1-p}{z-p}$$

Control design



- Do control design on delay-free model
- First order dynamics → PI-controller
- Pole placement, fully damped, not too aggressive

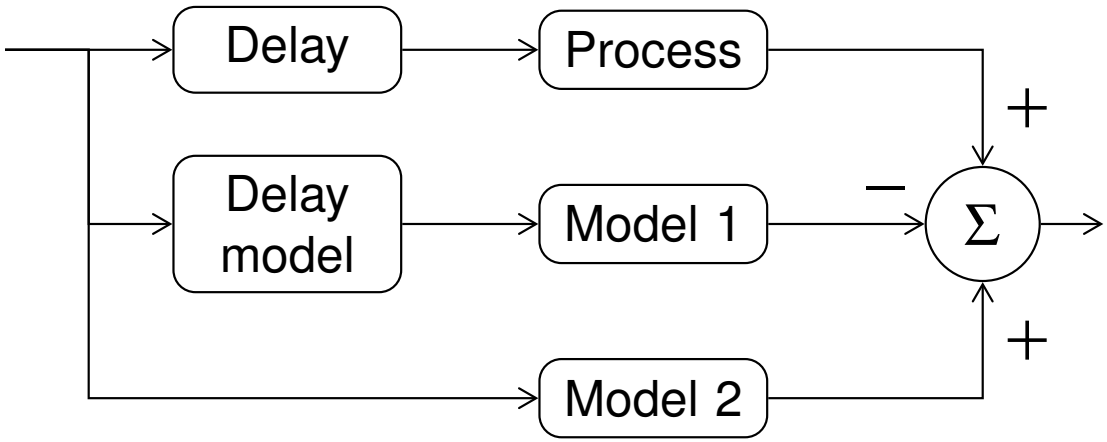
Experiments



- Plot shows average over 50 experiments
- Step in load at $t = 200$
- Responds fast
- Damped response avoids unnecessary VM boots

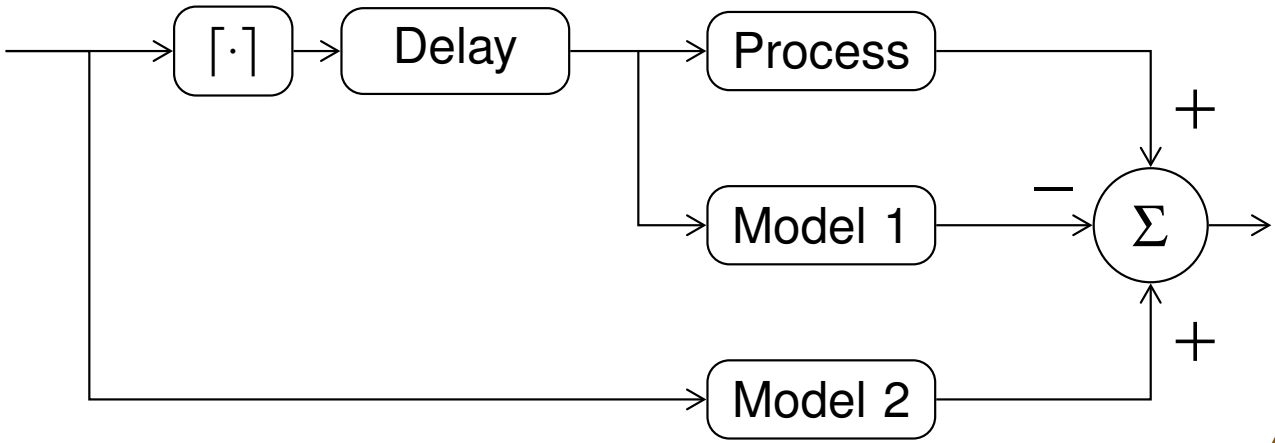
Delay compensation in stationarity

- Process and Model 1 cancel out (perfect model)
- In stationarity, Model 1 and Model 2 cancel out, eliminating model errors, $e = 0 \Rightarrow y = r$



Delay compensation in stationarity

- Process and Model 1 cancel out (perfect model)
- In stationarity, Model 1 and Model 2 WON'T cancel out since model inputs differ, $e = 0 \not\Rightarrow y = r$
  - (Gives us quantization compensation, allowing us to reach stationarity even though we have quantization)



Where do we end up?

Compensated output is

$$\hat{T} = \underbrace{T(t, \lambda, m_r)}_{\text{Process output}} - \underbrace{T_m(t, \lambda, m_r)}_{\text{Delayed model}} + \underbrace{T_m(t, \lambda, m)}_{\text{Delay-free model}}$$

where

$$m_r = \text{Delay}\{\lceil m \rceil\}.$$

Where do we end up?

Assuming we reach stationarity with control signal m_0 for some constant load we have (omitting some arguments)

$$T_{ref} = T^0(\lceil m_0 \rceil) - T_m^0(\lceil m_0 \rceil) + T_m^0(m_0)$$

T^0 and T_m^0 are stationary metric for process and model.

Since $\lceil m_0 \rceil \geq m_0$ and T_m^0 is strictly decreasing we get

$$T_{ref} \geq T^0(\lceil m_0 \rceil)$$

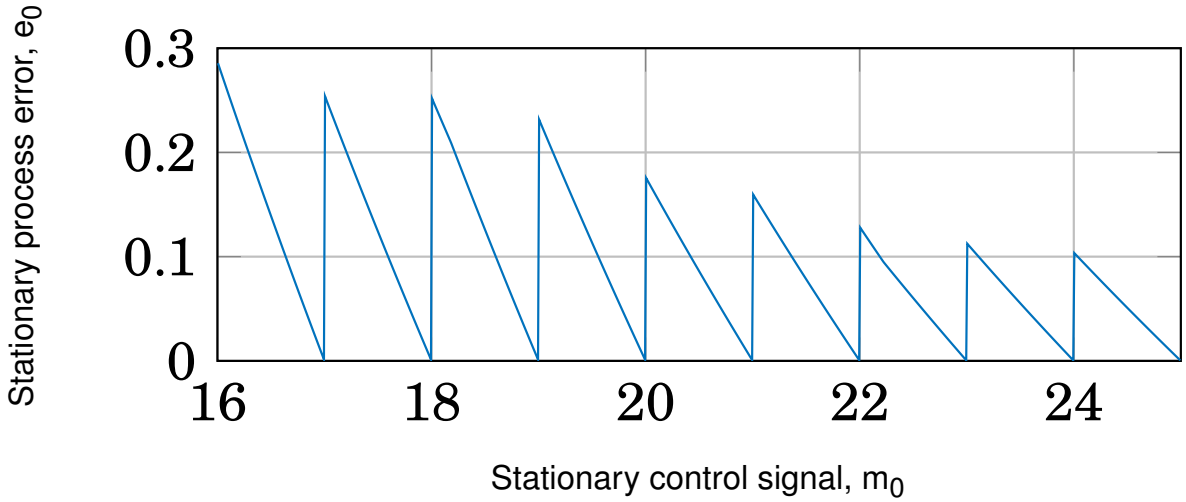
and

$$e_0 = T_{ref} - T^0(\lceil m_0 \rceil) = T_m^0(m_0) - T_m^0(\lceil m_0 \rceil).$$

Stationary metric is always better than the set point, **regardless of model errors**!

e_0, how far from set point

Plot process error $e_0 = T_m^0(m_0) - T_m^0(\lceil m_0 \rceil)$ for `mandel` model



$$e_0 = T_{ref} - T^0(\lceil m_0 \rceil) \Leftrightarrow T^0(\lceil m_0 \rceil) = T_{ref} - e_0(m_0)$$

If $T_{ref} = 0.25$ we can get the requirement $T^0(\lceil m_0 \rceil) < 0$.

Negative response time?!?

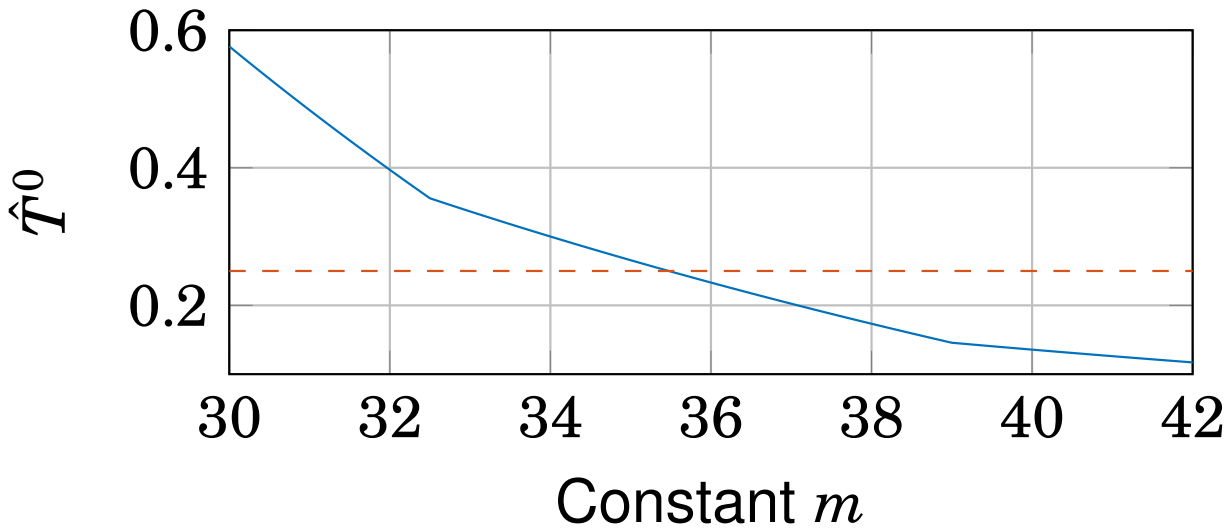
$\hat{T}^0$ for some model errors

Stationary compensated metric $\hat{T}^0$ for model errors (factor γ) with constant control signal m

$$T^0(m) = \gamma\, T_m^0(m)$$

$$\hat{T}^0(m) = \gamma\, T_m^0(\lceil m \rceil) - T_m^0(\lceil m \rceil) + T_m^0(m)$$

No error, $\gamma = 1$



Any T_{ref} possible!

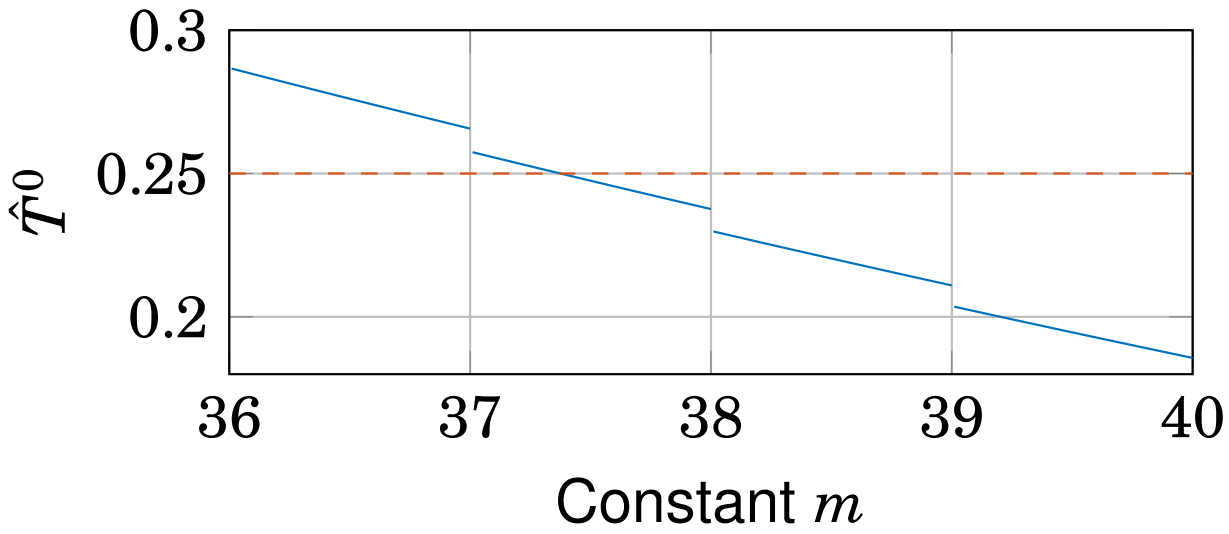
$\hat{T}^0$ for some model errors

Stationary compensated metric $\hat{T}^0$ for model errors (factor γ) with constant control signal m

$$T^0(m) = \gamma\, T_m^0(m)$$

$$\hat{T}^0(m) = \gamma\, T_m^0(\lceil m \rceil) - T_m^0(\lceil m \rceil) + T_m^0(m)$$

$\gamma = 1.4$



At least $T_{ref} = 0.25$ still possible

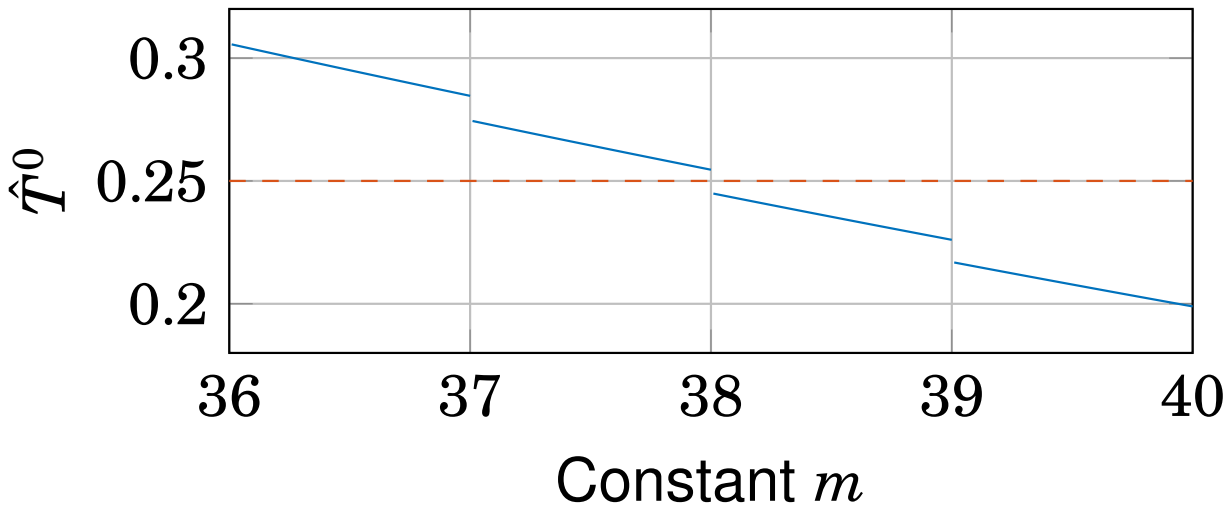
$\hat{T}^0$ for some model errors

Stationary compensated metric $\hat{T}^0$ for model errors (factor γ) with constant control signal m

$$T^0(m) = \gamma\, T_m^0(m)$$

$$\hat{T}^0(m) = \gamma\, T_m^0(\lceil m \rceil) - T_m^0(\lceil m \rceil) + T_m^0(m)$$

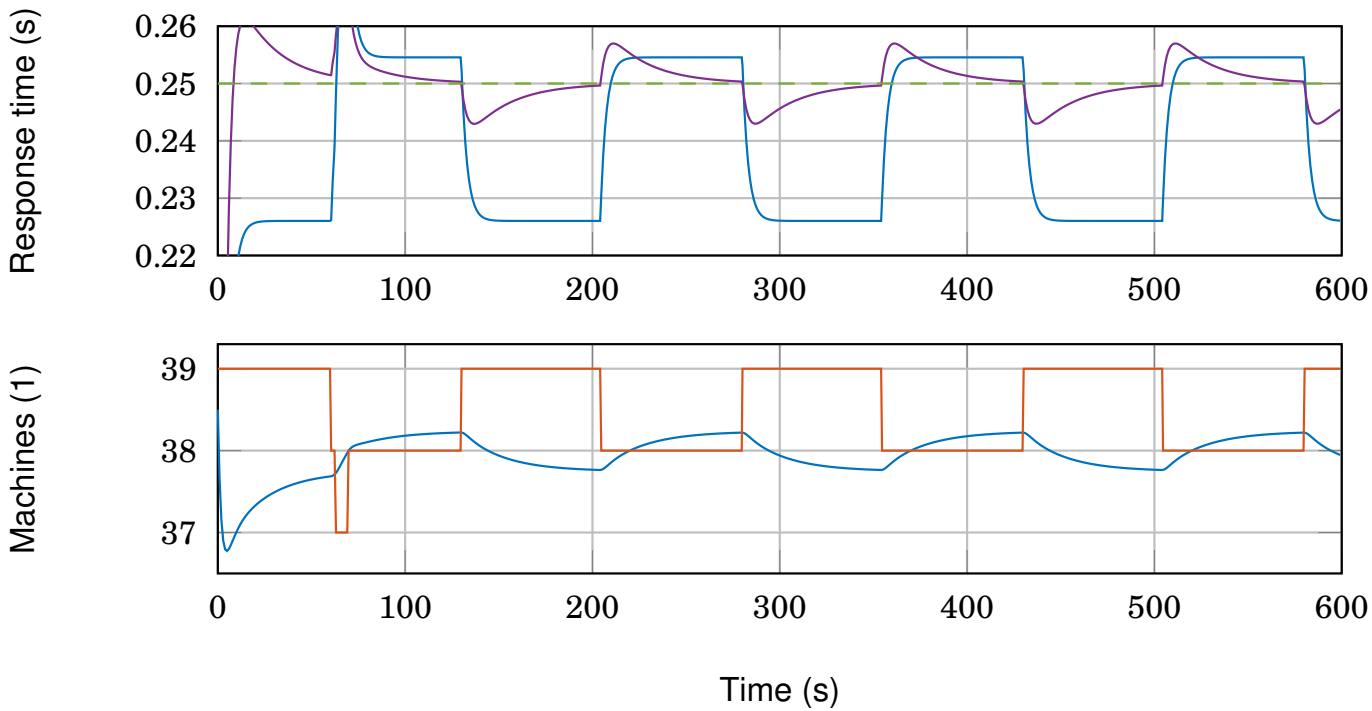
$\gamma = 1.5$



No m gives $\hat{T}^0 = T_{ref} = 0.25$.

Simulation with model error

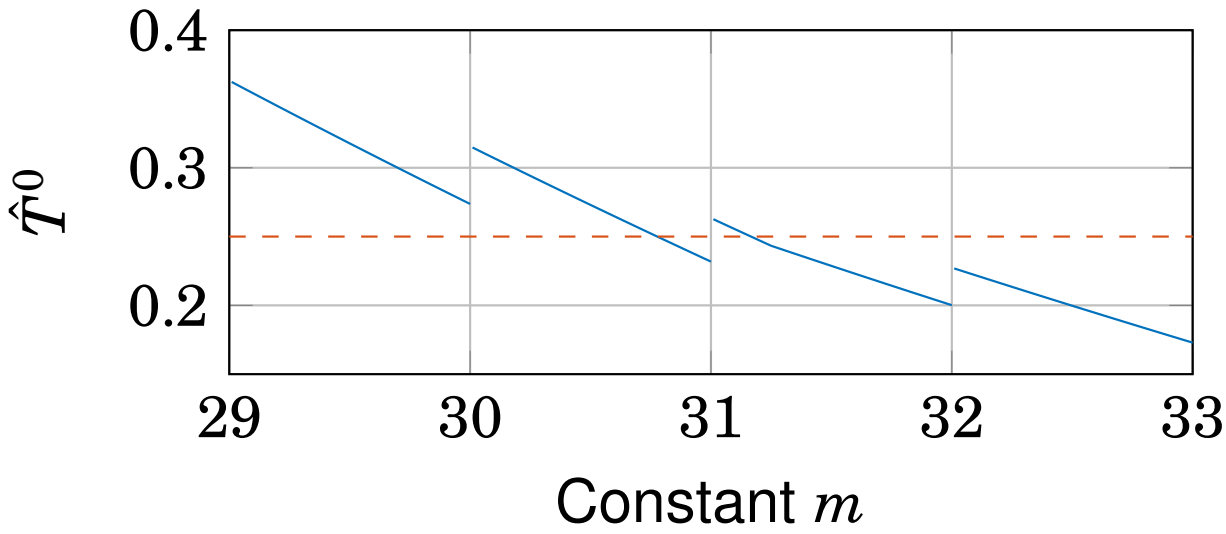
Simulation of models with factor $\gamma = 1.5$



Not (yet) reproduced in experiments.

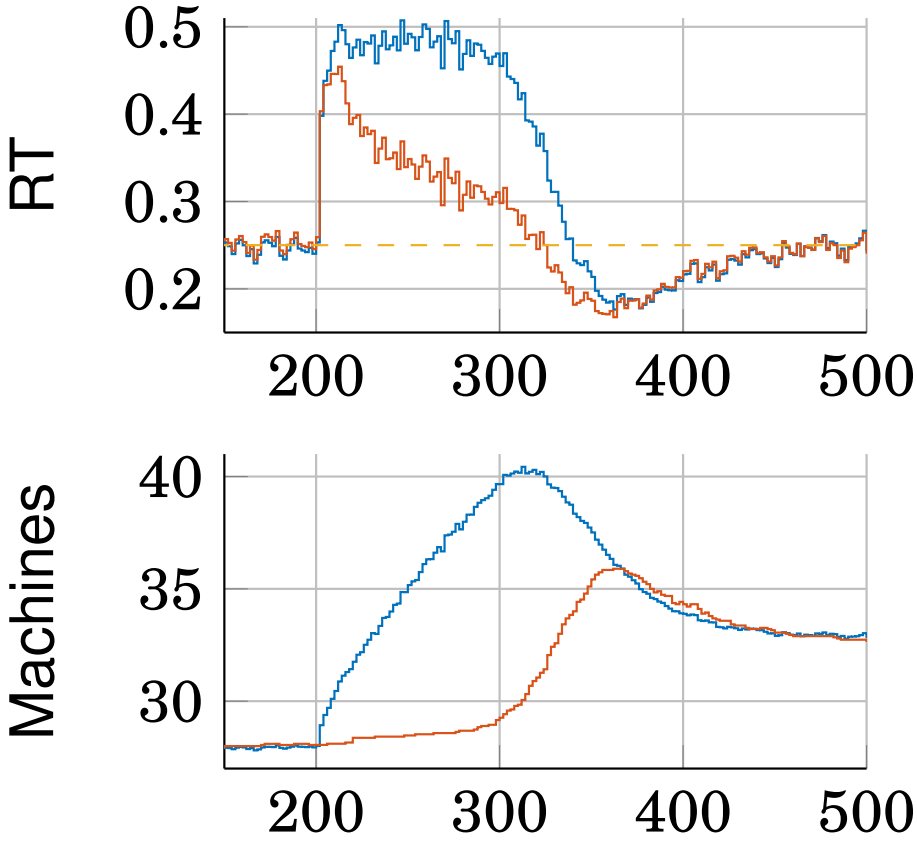
$\hat{T}^0$ for some model errors

$\gamma < 1$



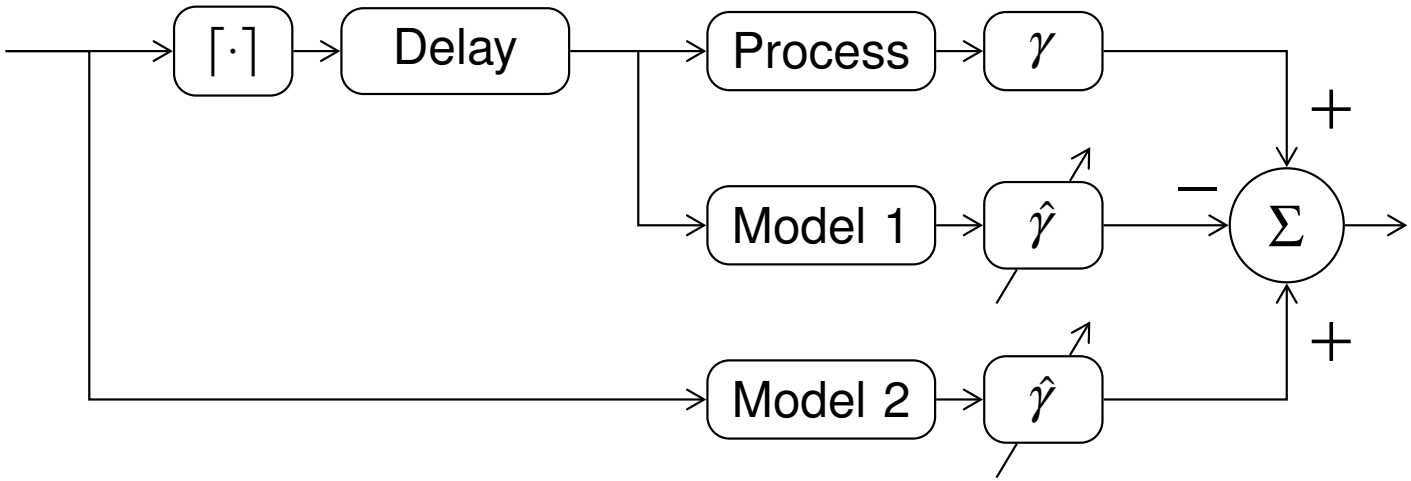
Stationary point depends on initial conditions

Model errors



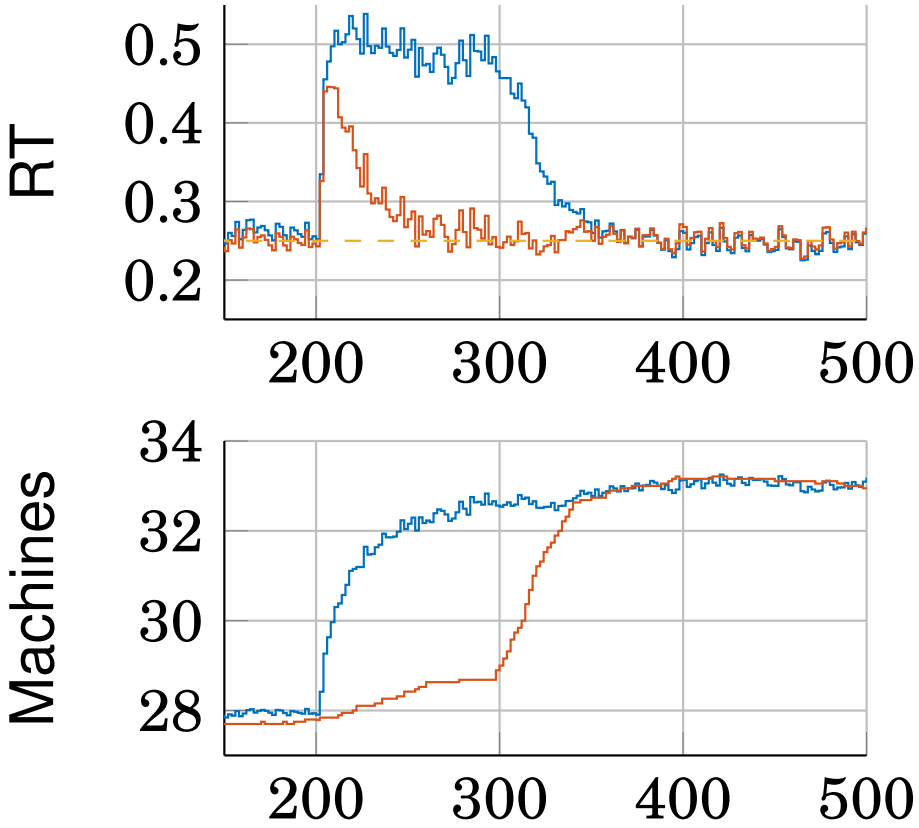
- Plot shows average over 20 experiments
- Step in work required per request at $t = 200$
- Responds fast
- Large, costly overshoot

Adaptive Model



Adaptation of “extra” gain γ based on Process and Model 1 output.

Experiments



- Plot shows average over 20 experiments
- Step in work required per request at $t = 200$
- Responds fast
- . . . ?

Future Work

- Online estimation of model
- Stability/Robustness
- Noise rejection

The End

Questions?

...

Extra frames coming...

$$T_{ref} = \underbrace{T^0(\lambda_0, \lceil m_0 \rceil) - T^0_m(\lambda_0, \lceil m_0 \rceil)}_{=0} + T^0_m(\lambda_0, m_0) = T^0_m(\lambda_0, m_0).$$

$$\lceil m_0 \rceil - 1 < m_0.$$

$$T^0_m(\lambda_0, \lceil m_0 \rceil - 1) > T^0_m(\lambda_0, m_0) = T_{ref}$$

Since $T^0_m(\lambda, m) = T^0(\lambda, m) \forall m \in \mathbb{N}$ we get

$$T^0(\lambda_0, \lceil m_0 \rceil - 1) > T_{ref}.$$

In other words, assuming we were to remove one VM response times would instead exceed the reference.

Startup times as delay