



 POLITECNICO DI MILANO



## Dynamic Trade-off Analysis of QoS and Energy Saving in Admission Control for Web Service Systems

Charles Poussot Vassal<sup>a</sup>, Mara Tanelli<sup>b</sup>, Marco Lovera<sup>b</sup>

<sup>a</sup> ONERA/DCSD, Toulouse Cedex, France. E-mail: [charles.poussot-vassal@onera.fr](mailto:charles.poussot-vassal@onera.fr)

<sup>b</sup> Dipartimento di Elettronica e Informazione, Politecnico di Milano.  
E-mail: {tanelli, lovera}@elet.polimi.it




- Reference scenario: autonomic systems
- Problem statement and notation
- LPV models for Web Service systems
  - identification approach
  - experimental framework for data collection
  - validation
- LPV-Model Predictive Control (LPV-MPC)
  - brief overview of MPC control
  - problem formulation
  - trade-off analysis
  - closed-loop performance analysis
- Concluding remarks and outlook



- Large service centers providing computational capacity on demand

- Problems:

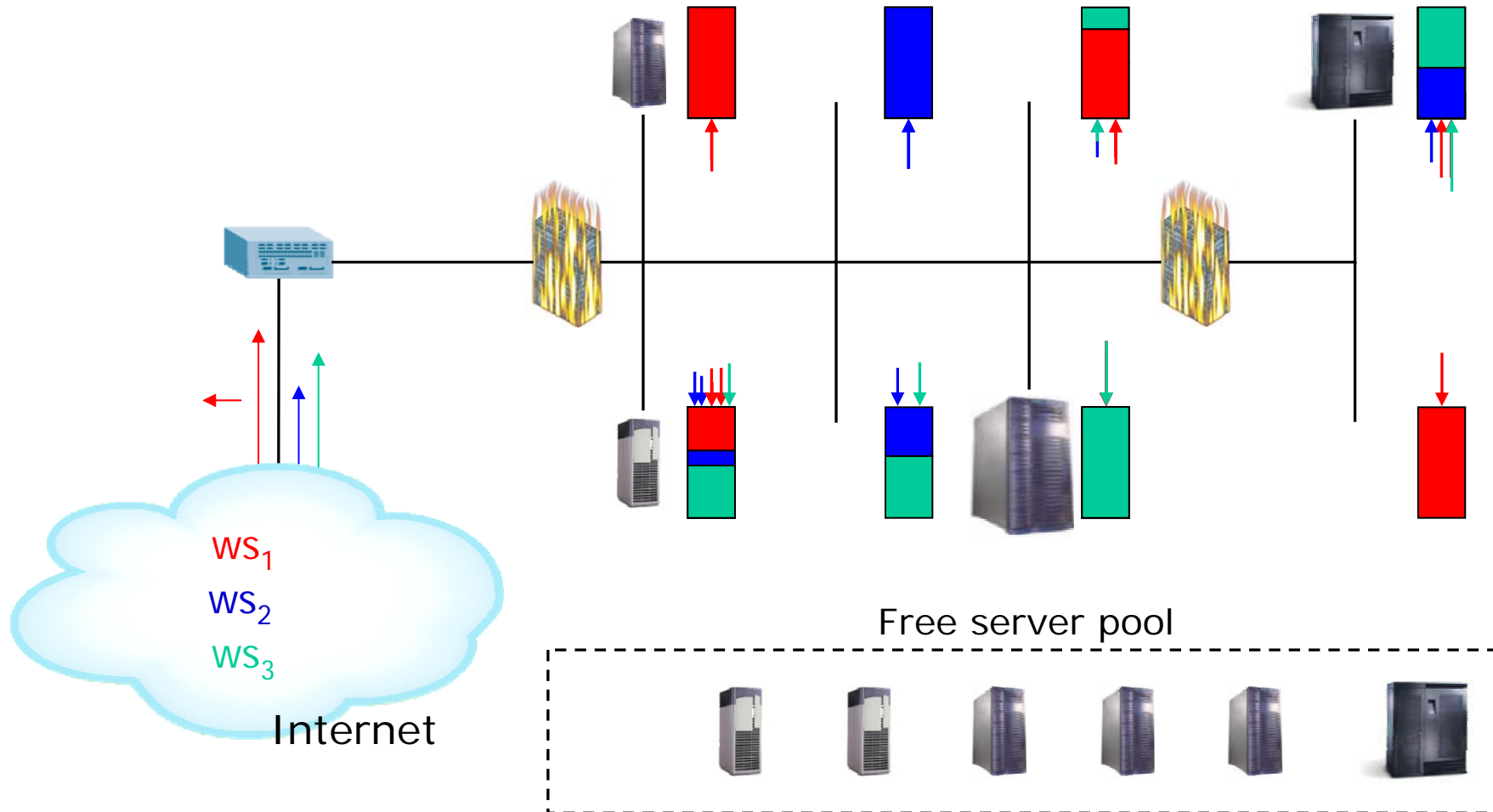
- Workload fluctuations
- Quality of Service (QoS) guarantees
- Energy (and related) costs



Trade-off between energy-saving objectives and QoS goals.  
Need to devise quantitative methods to evaluate it at design time

- Solutions:

- Autonomic self-management techniques
- Reconfiguration of service center infrastructures in order to determine Performance vs. Energy trade-off

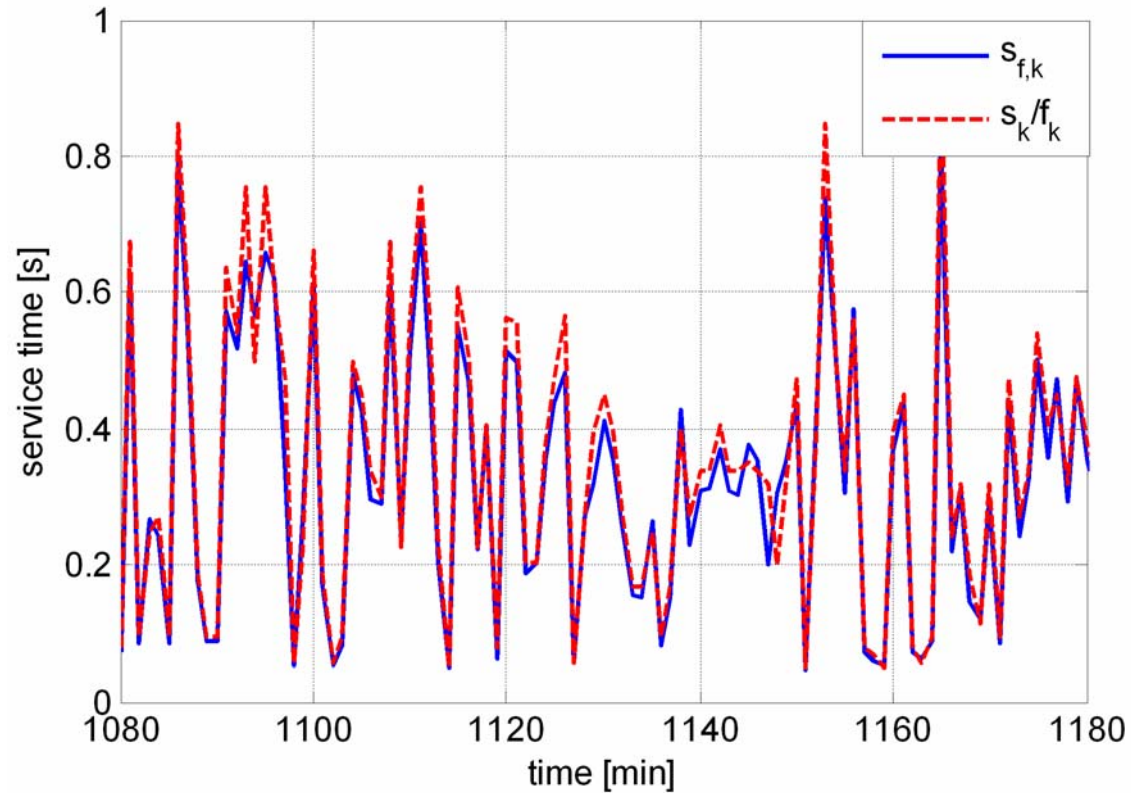




- Utility Based Approach: Queueing Network model + Optimization framework (e.g., Tivoli)
  - Multiple decision variables
  - Long term time horizon (several minutes)
- Control Theory Approach
  - Short time frame (minutes, seconds)
  - System identification used to develop models for:
    - Capturing system transients
    - Taking into account workload variability
  - Advanced control design techniques used to:
    - Ensure closed-loop stability
    - Guarantee performance (QoS) levels *a priori*



- Single class Web server with FIFO scheduling
  - $\lambda_k$ : requests arrival rate
  - $s_k$ : service time, CPU time required to serve a single request
  - $T_k$ : response time, overall time a request stays in the system
  - $X_k$ : system throughput, requests service rate
  - $P_k$ : admission probability
  - $f_k$ : ratio between current server CPU frequency and maximum CPU frequency.
- Dynamic voltage scaling (DVS) modeling:  $s_{u,k} = s_k / f_k$  effective service time
- Queuing theory:
  - Steady state assumption
  - Average response time:  $\bar{T} = \frac{\bar{N}}{X} = \frac{\bar{N}}{\lambda}$
- We aim at: identifying a reliable model of the Web service dynamics based on which to design an optimal LPV-MPC controller to analyse the trade off between energy saving and QoS



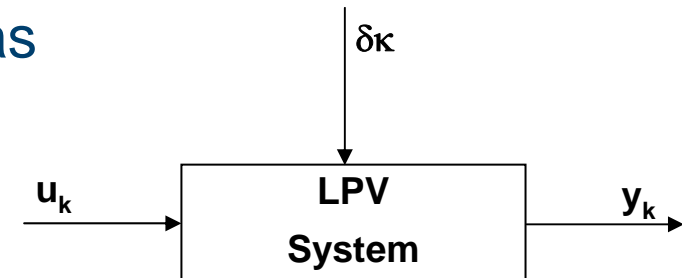
Dynamic voltage scaling (DVS) modeling:  $s_{u,k} = s_k / f_k$  effective service time



Linear Parameter Varying systems are a particular class of time-varying systems.

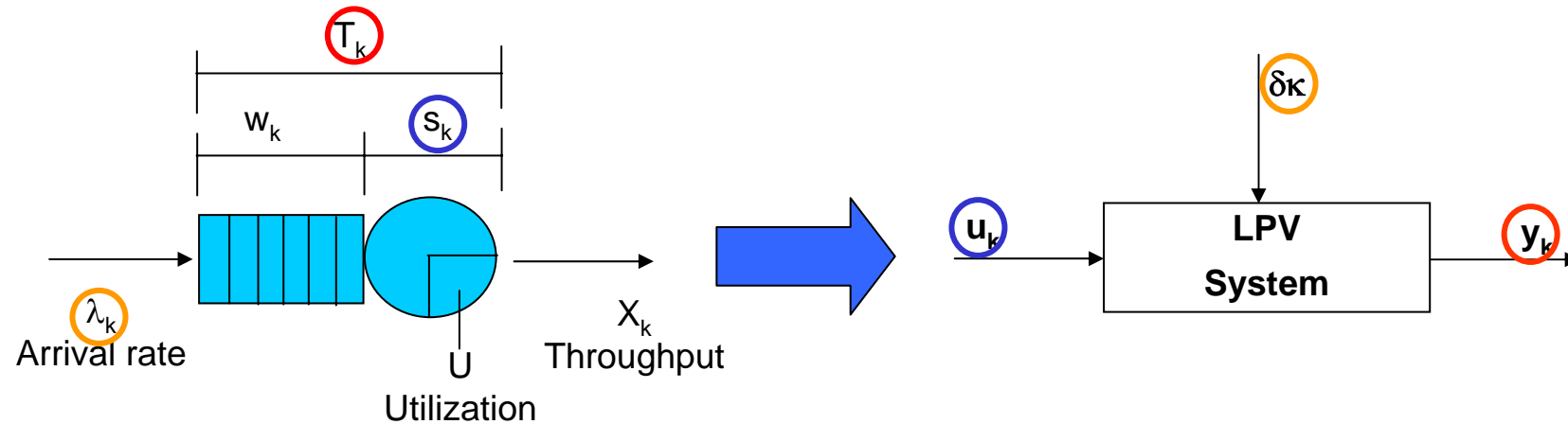
In state space form they are described as

$$\begin{aligned}x_{k+1} &= A(\delta_k)x_k + B(\delta_k)u_k \\y_k &= C(\delta_k)x_k + D(\delta_k)u_k\end{aligned}$$



“Time varying systems, the dynamics of which are functions of a measurable, time varying parameter vector  $\delta$ .”

Models for LTV systems or linearizations of non linear systems along the trajectory of  $\delta$  ) gain scheduling control problems.



We use models with:

- Affine parameter dependence (LPV-A), that is

$$A(\delta_k) = A_0 + A_1\delta_{1,k} + \dots + A_s\delta_{s,k}$$

and similarly for the B, C and D matrices

- Input-Affine (LPV-IA) parameter dependence, i.e., only the B and D matrices are parametrically varying



- A workload generator
  - Apache JMeter custom extension
- Micro benchmarking web application
  - CPU service time generated according to deterministic (identification), exponential, lognormal, Pareto (validation) distributions
- Application instrumentation (otherwise, ARM API or kernel-based measurement)
- Validation: synthetic workload inspired by a real-world usage (adapted from a banking application trace, 24 hours)

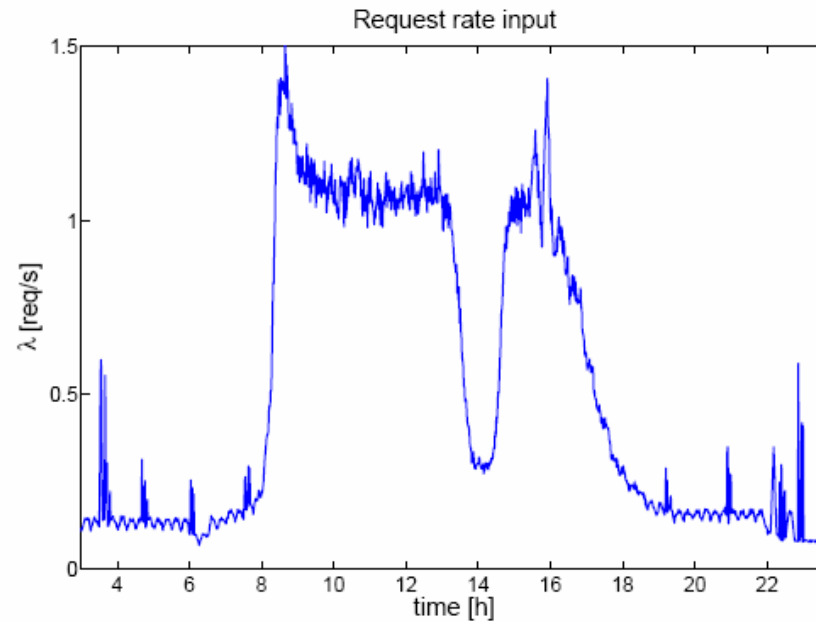
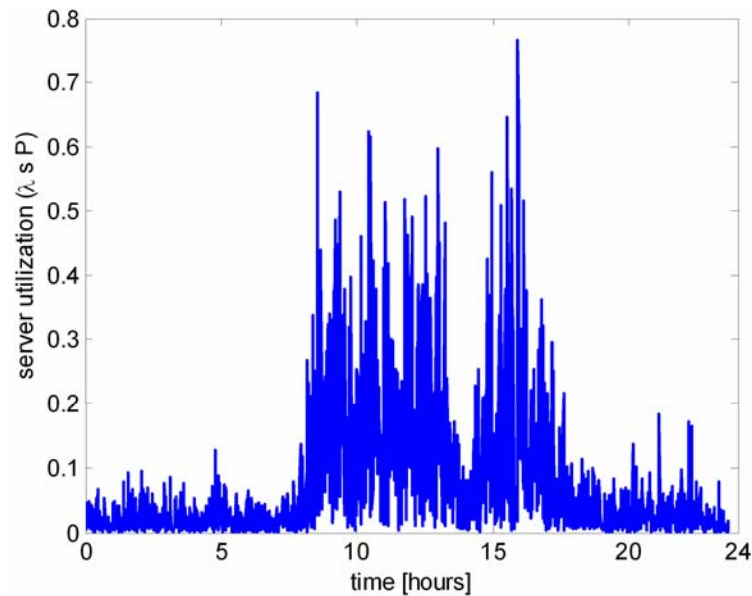


- The problem is set up as in the classical output error minimization framework
- The system is described by a set of parameters  $\theta$ , identification is performed minimising the cost function

$$V_N(\theta) := \sum_{k=1}^N \|y_k - \hat{y}_k(\theta)\|_2^2$$

with respect to  $\theta$

- Minimization carried out via a gradient search method (Levenberg-Marquardt algorithm)



Performance metrics:

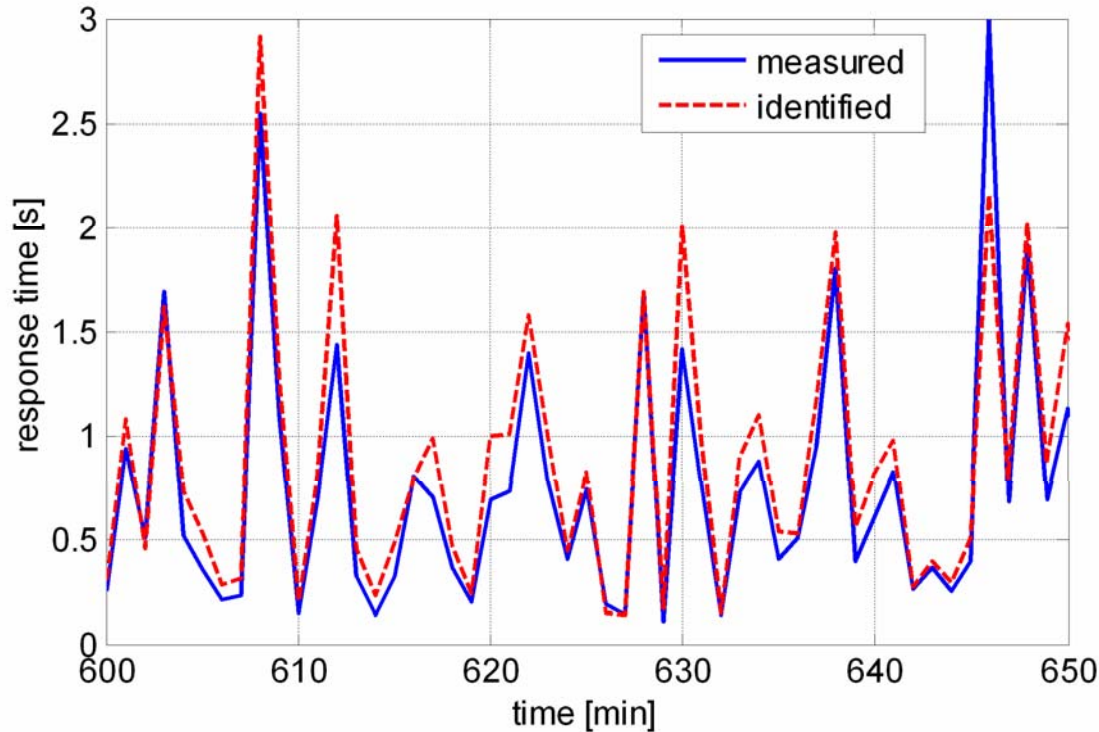
- Variance accounted for (VAF)
- Average simulation error ( $e_{avg}$ )

$$VAF = 100 \left( 1 - \frac{Var[y_k - y_{sim,k}]}{Var[y(k)]} \right)$$

$$e_{avg} = 100 \left( \frac{E[|y_k - y_{sim,k}|]}{E[|y_k|]} \right)$$



$$\sigma[s_k] = q E[s_k], \quad q = \{2, 6\}$$



$$\begin{aligned}
 x_{k+1} &= Ax_k + \left( B_0 + B_1 s_{f,k} + B_2 s_{f,k} \lambda_k \right) P_k \\
 \xi_k &= Cx_k + \left( D_0 + D_1 s_{f,k} + D_2 s_{f,k} \lambda_k \right) P_k \\
 T_k &= \xi_k + s_{f,k},
 \end{aligned}$$

Valid. Performance $\Delta t = 1$ min	$q = 2$	$q = 6$
VAF on 24h	78.38%	74.96%
VAF light load	92.63%	83.01%
VAF heavy load	73.79%	63.57%
$e_{avg}$ on 24h	3.35%	6.60%
$e_{avg}$ light load	0.42%	2.85%
$e_{avg}$ heavy load	5.48%	10.74%



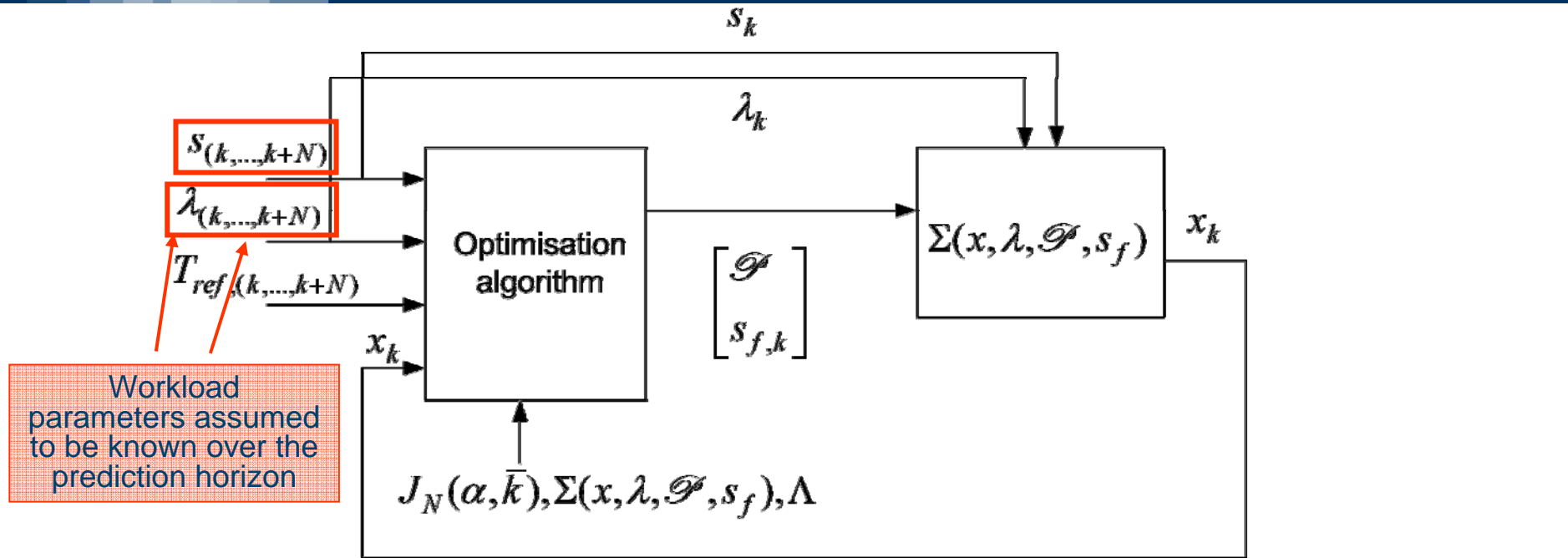
- Global performance requirements:
  - guaranteeing a given service time associated to a request, denoted with  $T_{ref}$ , defined and negotiated between the customers and the service provider
  - guaranteeing a certain minimal amount of accepted requests (modeled as an admission probability  $P_k$ )

while either:

1. Maximizing the number of served users: Quality of Service (QoS) objective.
2. Minimizing the power consumption (i.e., maximising the effective service time ): Energy saving objective.



- MPC: a widely employed control approach for large scale, multivariable, possibly constrained systems
- Main idea:
  1. the control problem is formulated as an optimisation one, based on
    - a mathematical model for the system (and known external disturbances)
    - a cost function expressing the desired performance over a fixed time horizon
    - all the relevant constraints on input, state and output variables
  2. the control action over the future horizon is computed solving the optimisation problem on line (via dynamic programming)
  3. the implementation of the control action is based on the *receding horizon principle*, i.e., at each time step only the first sample of the computed control sequence is actually applied and the control problem is re-solved at the subsequent time step



$$J_N(\alpha, \bar{k}) = \alpha J_{QoS}(\bar{k}) + (1 - \alpha) J_{ES}(\bar{k}) = \alpha \sum_{k=\bar{k}}^{\bar{k}+N-1} \left| \frac{\bar{p} - p_k}{\bar{p} - \underline{p}} \right| + (1 - \alpha) \sum_{k=\bar{k}}^{\bar{k}+N-1} \left| \frac{\bar{s}_f - s_{f,k}}{\bar{s}_f - \underline{s}_f} \right|$$



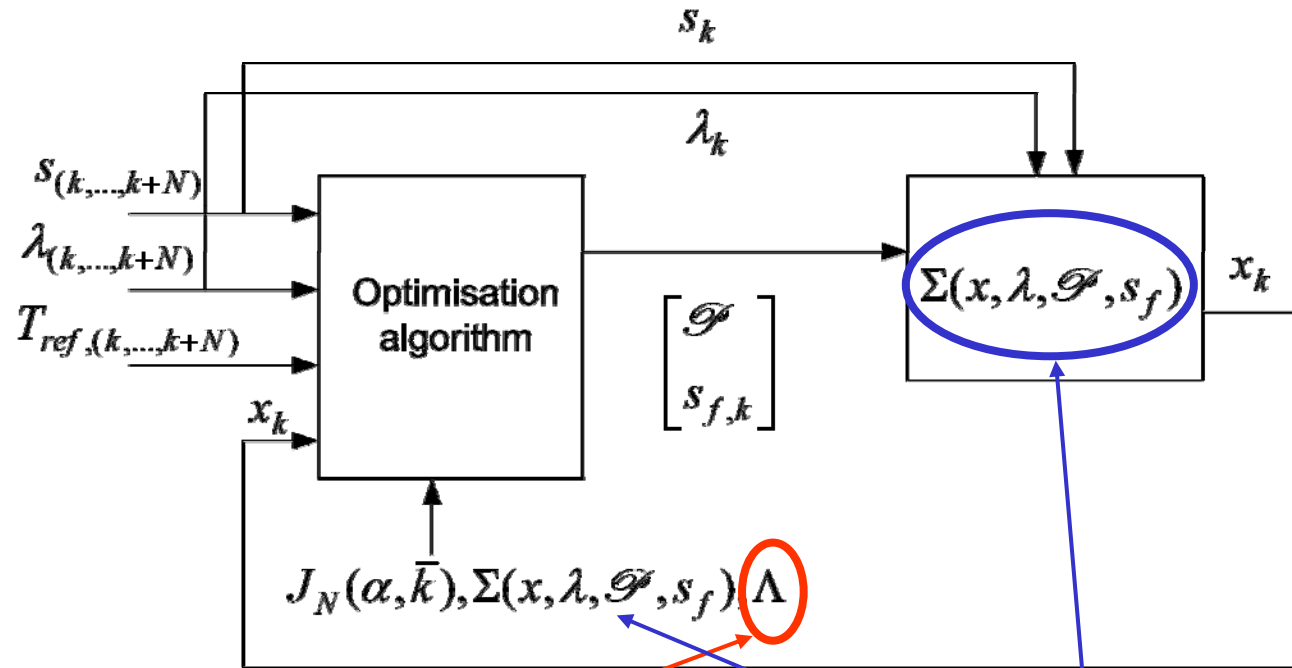
$$J_N(0, \bar{k}) = \sum_{k=\bar{k}}^{\bar{k}+N-1} \left| \frac{\bar{s}_f - s_{f,k}}{\bar{s}_f - \underline{s}_f} \right|$$

Energy-Saving objective

$$J_N(1, \bar{k}) = \sum_{k=\bar{k}}^{\bar{k}+N-1} \left| \frac{\bar{p} - p_k}{\bar{p} - \underline{p}} \right|$$

QoS objective





State and input inequality constraints

$$\Lambda : \begin{cases} 0 \leq \xi_k \\ \underline{\mathcal{P}} \leq \mathcal{P}_k \leq \overline{\mathcal{P}} \\ \underline{s}_f \leq s_{f,k} \leq \overline{s}_f \\ -\Delta \leq T_k - T_{ref} \leq \Delta \end{cases}$$

LPV model (also dynamic equality constraints)

$$\Sigma(x, \lambda, \mathcal{P}, s_f) = \begin{aligned} x_{k+1} &= Ax_k + (B_0 + B_1 s_{f,k} + B_2 s_{f,k} \lambda_k) \mathcal{P}_k \\ \xi_k &= Cx_k + (D_0 + D_1 s_{f,k} + D_2 s_{f,k} \lambda_k) \mathcal{P}_k \\ T_k &= \xi_k + s_{f,k}, \end{aligned}$$



# Trade-Off Analysis

$$\forall k \in [\bar{k}, \bar{k} + N - 1]$$

$$\text{solve } J_N^*(\alpha, \bar{k}) = \min J_N(\alpha, \bar{k}), \text{ subject to } \begin{cases} x_{k+1} \xi_k T_k = \text{LPV Model} \\ \Lambda = \text{Input and State Constr.} \end{cases}$$

Cost functions for performance evaluation

$$J_{QoS} = \sum_{k=1}^{N_f} \frac{\|\mathcal{P}_k - \underline{\mathcal{P}}\|_2}{\|\bar{\mathcal{P}} - \underline{\mathcal{P}}\|_2}$$

$$J_{ES} = \sum_{k=1}^{N_f} \frac{\|s_{f,k} - \underline{s}_f\|_2}{\|\bar{s}_f - \underline{s}_f\|_2}$$

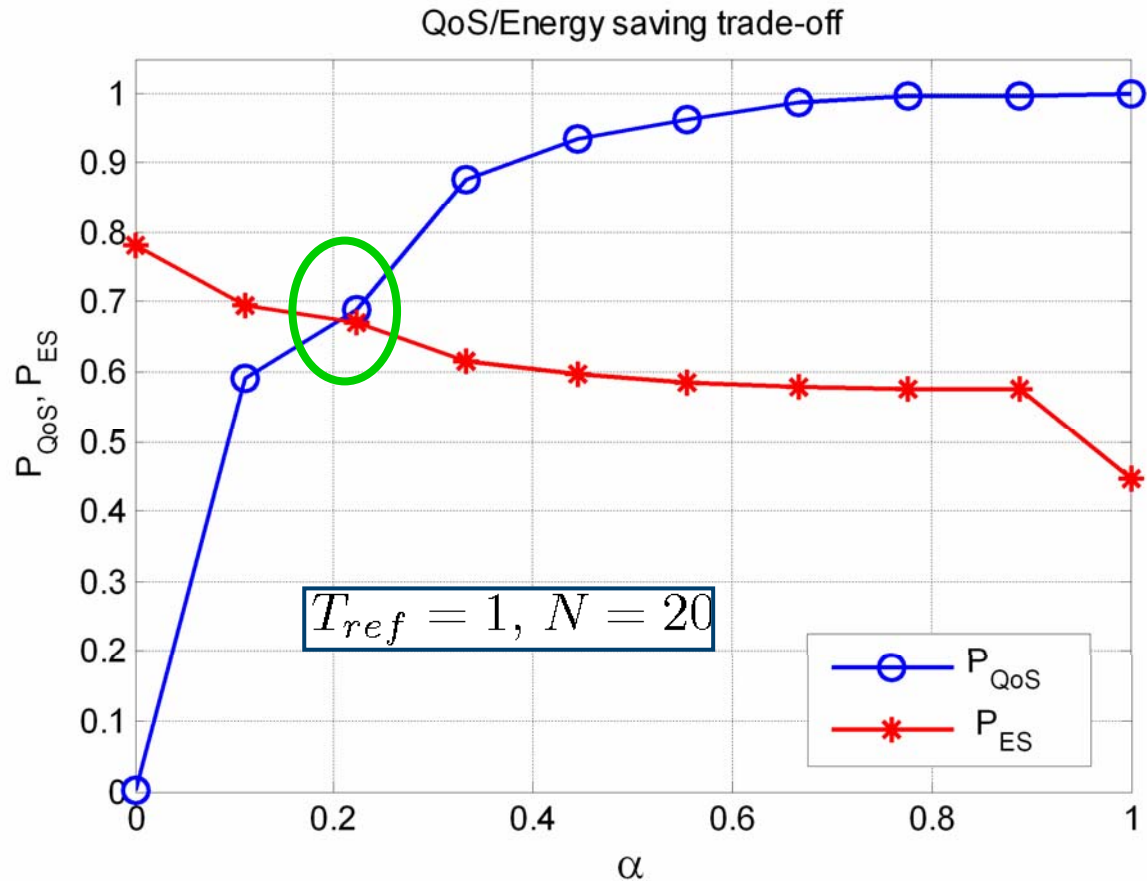


$J_{QoS}=1 \rightarrow$  max # of requests accepted  $\rightarrow$  QoS maximised

$J_{ES}=1 \rightarrow$  effective service time maximised (i.e., minimum CPU freq.)  $\rightarrow$  ES maximised

$$[\underline{\mathcal{P}}, \bar{\mathcal{P}}] = [0.5, 1]$$

$$[\underline{s}_f, \bar{s}_f] = [0.5 s_k, s_k]$$



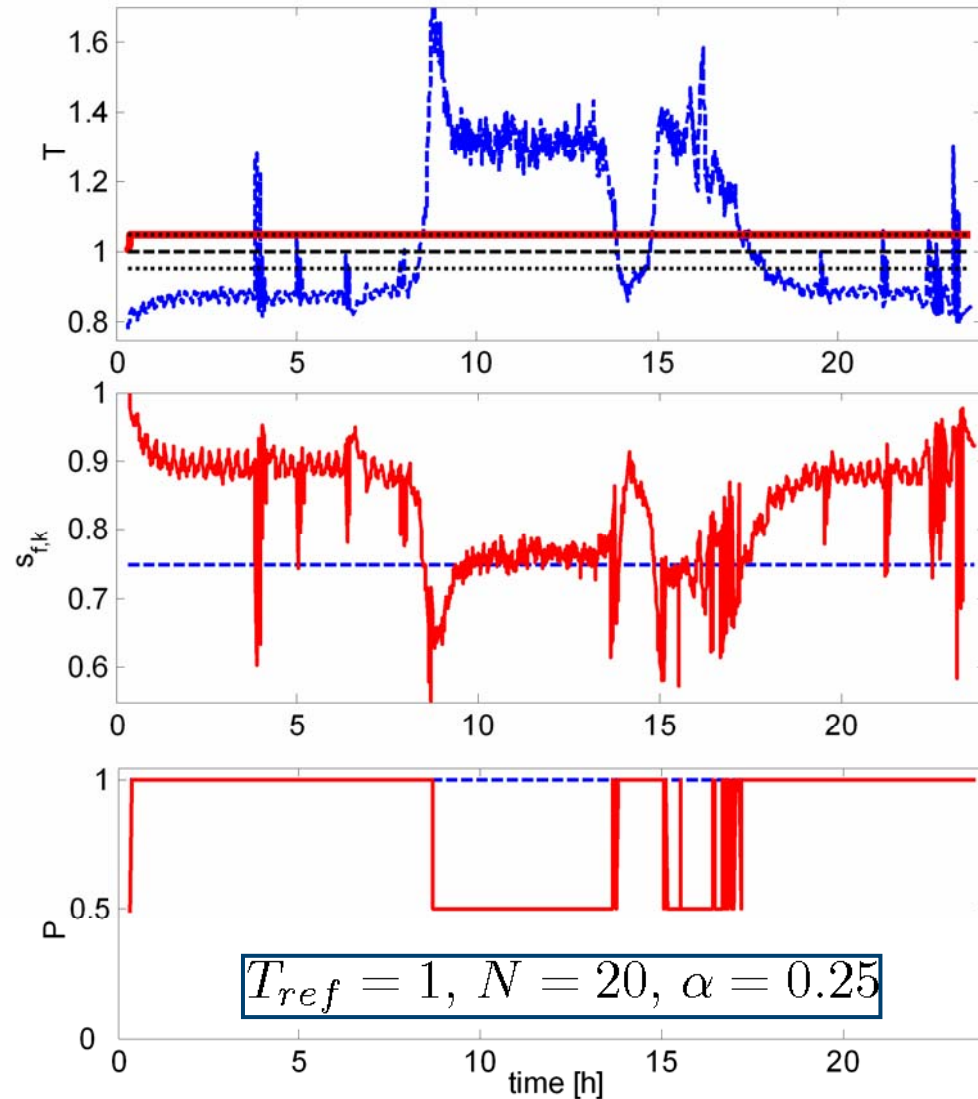


$$[\underline{P}, \overline{P}] = [0.5, 1]$$

$$[\underline{s}_f, \overline{s}_f] = [0.5 s_k, s_k]$$

When the number of requests is large, (between 9 and 16 h), the admission probability is reduced to obtain an effective service time  $s_{f,k}$  as large as possible to save energy.

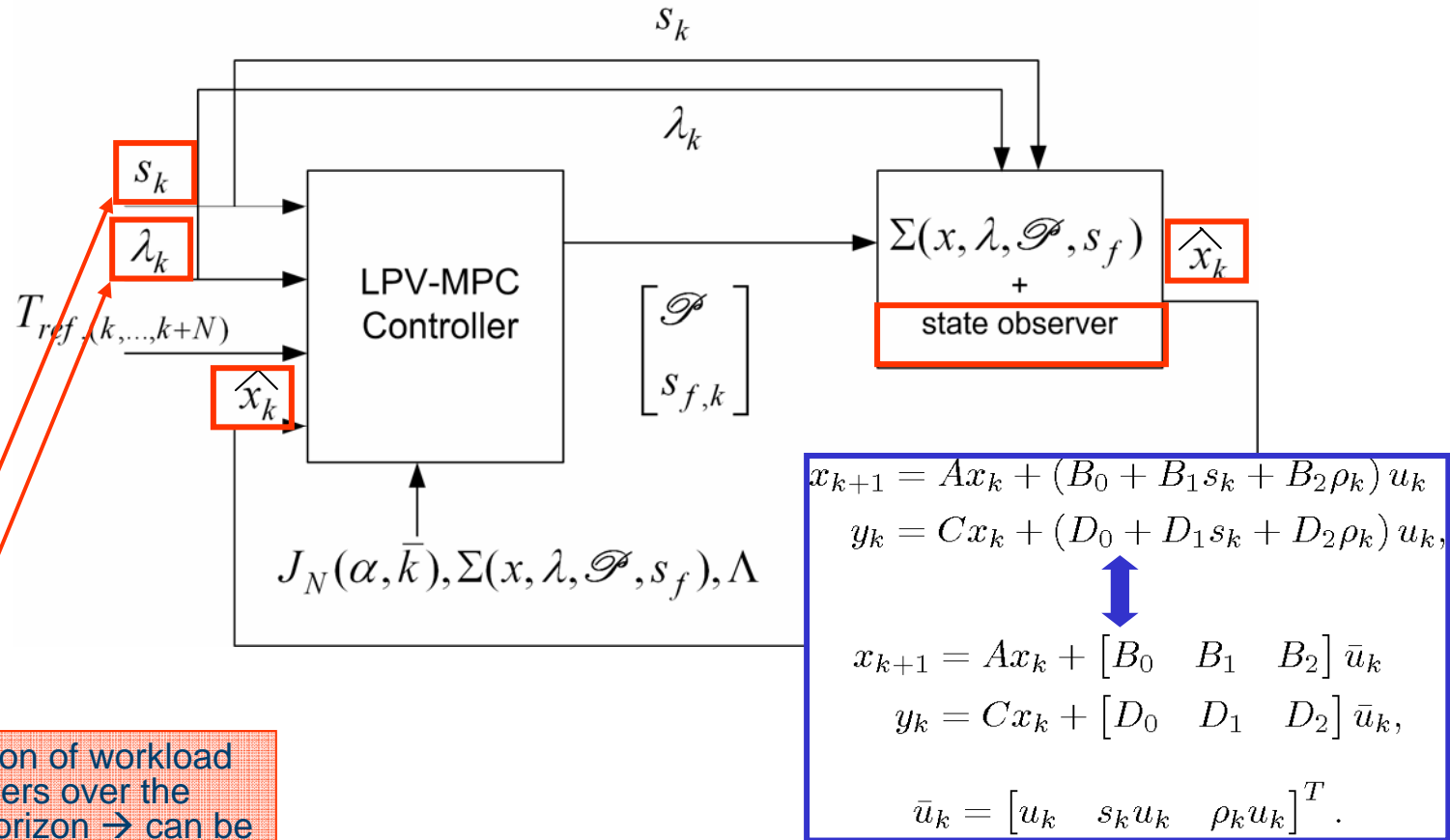
When  $\lambda_k$  is low the effective service time is increased, meaning that the CPU frequency is lowered, thus resulting in reduced energy consumption and the admission probability is at its upper bound to ensure good QoS.





# Sub-optimal LPV-MPC control

(results submitted to American Control Conference, ACC 2010)



No prediction of workload parameters over the prediction horizon → can be complemented with workload estimators [andreolini et al., Valuetools 2006]

$$x_{k+1} = Ax_k + (B_0 + B_1 s_k + B_2 \rho_k) u_k$$

$$y_k = Cx_k + (D_0 + D_1 s_k + D_2 \rho_k) u_k$$

$$x_{k+1} = Ax_k + [B_0 \quad B_1 \quad B_2] \bar{u}_k$$

$$y_k = Cx_k + [D_0 \quad D_1 \quad D_2] \bar{u}_k$$

$$\bar{u}_k = [u_k \quad s_k u_k \quad \rho_k u_k]^T$$

Observer design for a linear, time invariant system → easy!



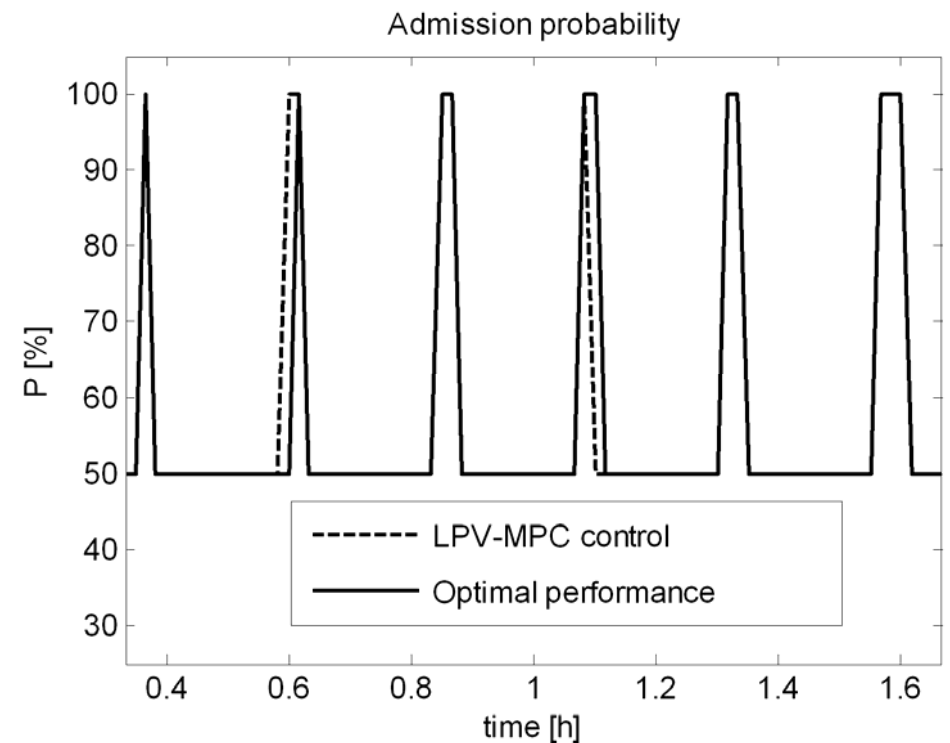
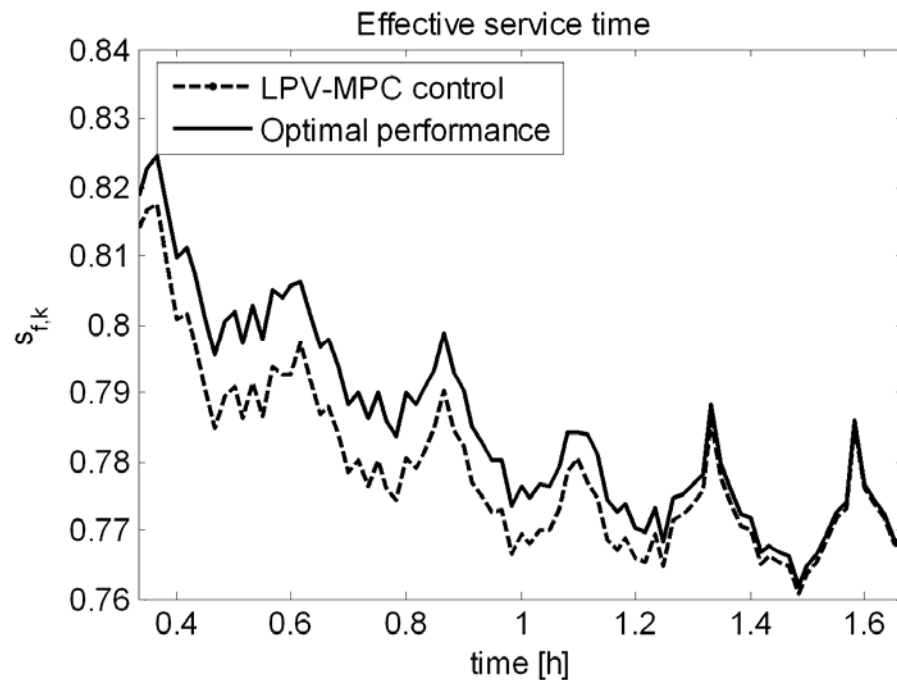
# Comparison of the closed-loop performance

(results submitted to American Control Conference, ACC 2010)

Test similar to the previous one

The control inputs of the sub-optimal (and really implementable) solution are very close to the optimal ones

This confirms the effectiveness of the proposed approach also under realistic assumptions





- This work presented a control theoretic framework for the dynamic analysis of QoS/Energy trade-offs in Web service systems
- Main contributions
  - effective identification approach to estimate reliable dynamic models
  - performance analysis method to evaluate the best achievable server performance using a numerical optimization approach
- Advantages
  - the approach can be used with no conceptual difference considering all available QoS-actuators (e.g., virtualisation) and different QoS and ES metrics
  - the sub-optimal controller is easy to implement and allows performance evaluation at design time