YORK UNIVERSITÉ UNIVERSITY

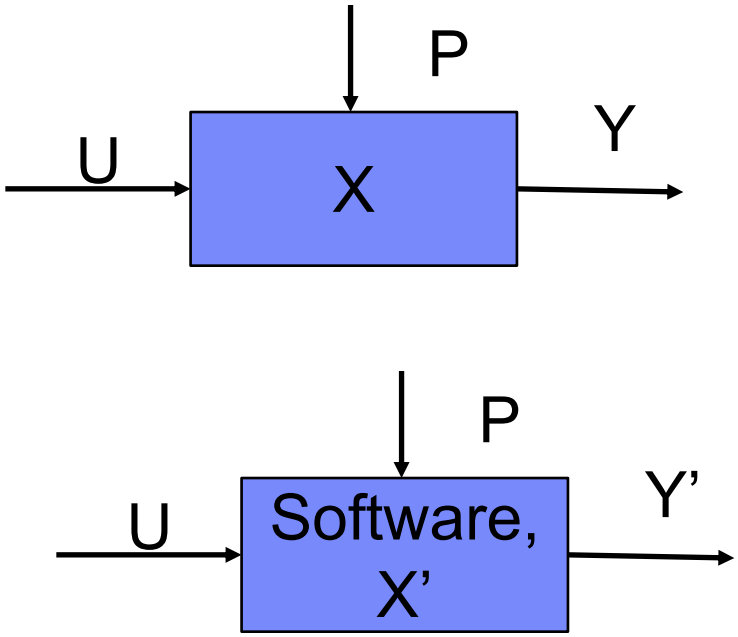
ASRL
Adaptive Systems Research Lab

Model Based Adaptive Systems
Control Theory Models

Marin Litoiu
Department of Electrical Engineering and Computer Science
York University
mlitoiu@yorku.ca
http://www.ceraslabs.com

The Role of Models

- **Y= F(U, P, X)**
  - Y outputs: response time, etc
  - P disturbances: number of users, arrival rate, etc..
  - X states: utilization, queue length, etc
- **F a (non-linear) function**
- **An autonomic manager, periodically**
  - Will find U that gives the desired Y, given the P and X
  - Will execute a plan that implements U
- **An expert can use the model to design the controller**
  - See ki, kp, kd tuning for PID controller



Control Theory Linear Models

In many cases, the non-linear model can be simplified or approximated with a linear one, such as the one below. In this model we assumed an additive external perturbation:

$$x(k+1) = A * x(k) + B * u(k) + F * p(k) \tag{3}$$

$$y(k) = C * x(k) + D * u(k) \tag{4}$$

where A, B, C, D, F are constant matrices that can be determined experimentally for a specific deployments and under particular perturbations[3].

Example: the linear model for an web server

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} + \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix} \begin{bmatrix} u_1(k) \\ u_2(k) \end{bmatrix} \quad (5)$$

$$y(k) = \begin{bmatrix} C_1 & 0 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} \quad (6)$$

- **Consider a web server**
- **x1: server utilization**
- **x2: memory utilization**
- **u1: no of http connections**
- **u2: keep alive interval**
- **y: response time**

Your turn (10 min)

- **Write the structure of the model for the following system**
  - Consider the web application as in your Assignment 1: two tiers, container based; assume that database tier is autoscaling (uses a non SQL database), therefore you can control the number of containers in both tiers
  - Consider response time as the output variable
  - Hint: to understand the dependencies, have a look at the open queuing model formulas

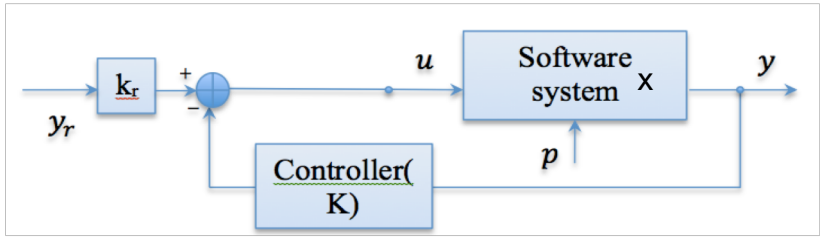
Properties of Control Systems: System Identification

- **The process of determining the equations that govern the model's dynamics is called system identification**
- **This can be done off-line: for example, executing a series of measures from which to calculate an approximated mathematical model, typically its transfer function or matrix**
- **Such identification from the output, however, cannot take account of unobservable dynamics**
- **Even assuming that a “complete” model is used in designing the controller, all the parameters included in these equations (called "nominal parameters") are never known with absolute precision; the control system will have to behave correctly even when connected to physical system with true parameter values away from nominal.**

How are the control theory models used?

- **Model used to study**
  - Controlability
  - Observability
  - Stability
  - Robustness
- **Models used to design the controller such a way that the qualities of the control are met (overshoot, stability, etc..)**
  - With a model, you can quickly tune a PID controller
  - There are “recipes/ procedures” for designing controllers
    - Linear Quadratic
    - Look-ahead, etc…

Linear Quadratic Control (LQR)



- **y_r: goal/setpoint/reference**
- **y: outputs**
- **u: inputs/commands (actionable by controller)**
- **x: states (internal variables)**
- **p: perturbations**
  - Workloads, faults, etc…
- **K and k_r – controller matrixes**

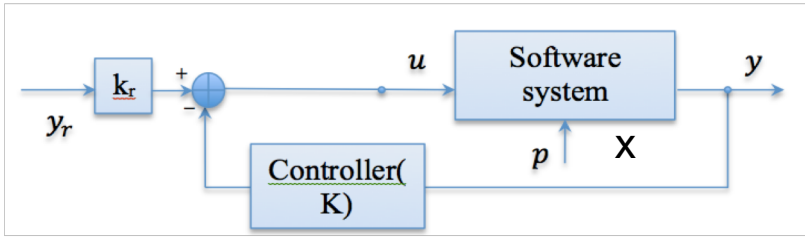
Linear Quadratic Regulator(LQR): K, k_r:

minimize J

$$J = \sum_{0}^{\infty} x^T Q_x x + u^T Q_u u$$

Q_x, Q_u, weight matrices

LQR Methodology



- **Methodology**
  - Have an adaptation goal (set point, objective function, etc…)
  - **Build a model (linear or linearized) of the software system**

$$\begin{cases} x(k+1) = Ax(k) + Bu(k) \\ y(k) = Cx(k) + Du(k) \end{cases}$$

  - Trough experiments, find A, B, C, D. Set Qx and Qu
  - Study the properties of system
    - Observability: can I estimate x if I only measure y?
    - Stability: bound inputs-> bound outputs?
    - Controllability: can I reach any x with set of inputs u?
  - **Synthesize a controller/Autonomic manager**

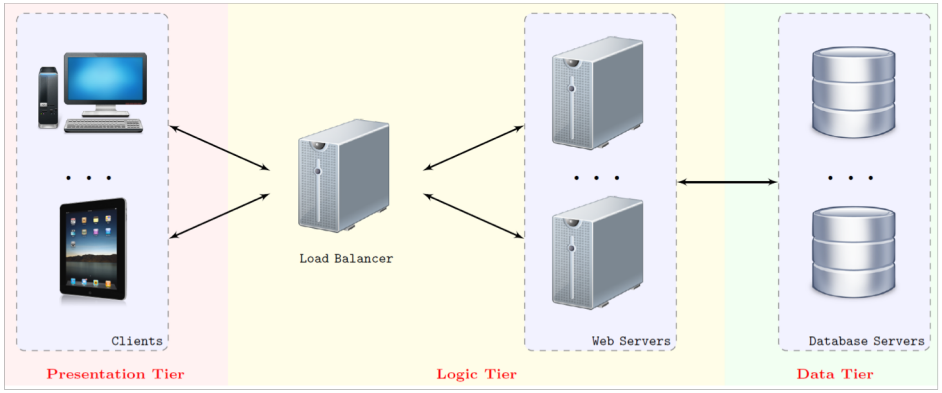
[K, k_r] r=h(goals, A, B,C,D)

$$K = -Q_u^{-1} B^T P x$$

$$PA + A^T P - PBQ_u^{-1} B^T P + Q_x = 0$$

$$1 = C(A - BK)^{-1} B k_r$$

Case Study



u=[webServers, threadsWS, DBServers, threadsDB]
y=x= [response time]
p= [workload]

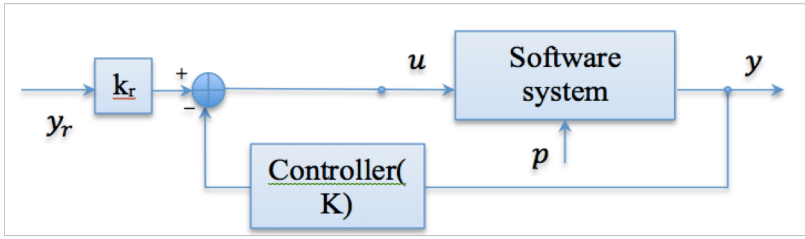
$$\begin{cases} x(k+1) &= Ax(k) + Bu(k) \\ y(k) &= Cx(k) + Du(k) \end{cases}$$

C=1, D=0; A=0.9 ; B=…

$$J = \sum_{0}^{\infty} x^T Q_x x + u^T Q_u u$$

$$Q_x = [1]; \quad Q_u = \begin{bmatrix} 100\,000 & 0 & 0 & 0 \\ 0 & 1\,000 & 0 & 0 \\ 0 & 0 & 100\,000 & 0 \\ 0 & 0 & 0 & 1\,500 \end{bmatrix}$$

What Limits LQR Robustness?



- **The model is static and linear**
  - not accurate for real software systems
- **The controller is designed statically**
- **In practice, it won't perform satisfactory**
- **We need a dynamic and adaptive LQR**

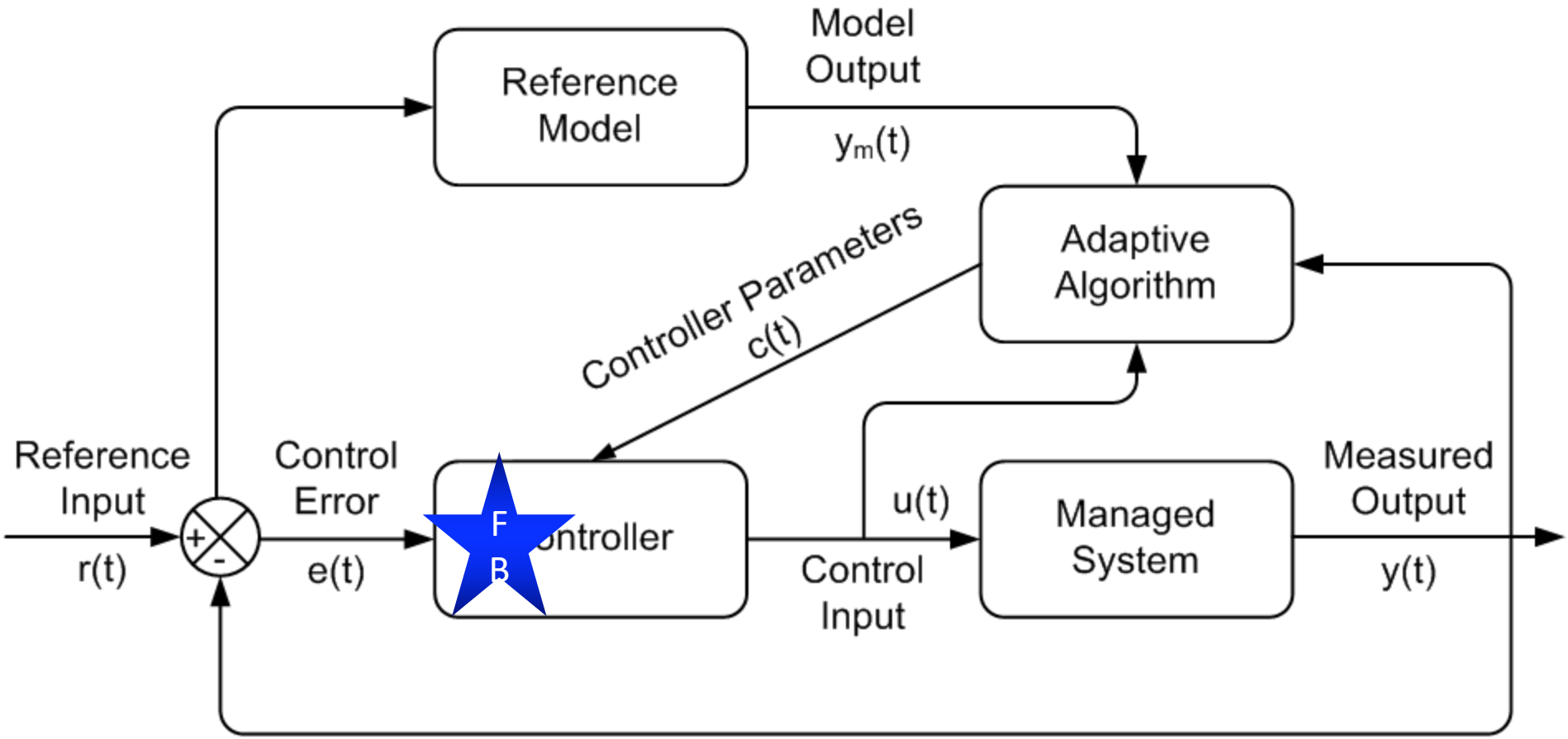
More on LQR

- **[1] 2015. Octave Optim Package v1.4.1. http://octave.sourceforge.net/optim/ overview.html. (2015).**
- **[2] Karl Johan Aström and Richard M Murray. 2010. Feedback systems: an introduction for scientists and engineers. Princeton university press (also, check on line)**
- Cornel Barna, Marin Litoiu, Marios Fokaefs, Mark Shtern, and Joe Wigglesworth. 2018. Runtime Performance Management for Cloud Applications with Adaptive Controllers. In Proceedings of the 2018 ACM/SPEC International Conference on Performance Engineering (ICPE '18). ACM, New York, NY, USA, 176-183. DOI: https://doi.org/10.1145/3184407.3184438

Model Reference Adaptive Controllers—MRAC

- **Also referred to as Model Reference Adaptive System (MRAS)**
- **Closed loop controller with parameters that can be updated to change the response of the system**
- **The output of the system is compared to a desired response from a reference model (e.g., simulation model)**
- **The control parameters are updated based on this error**
- **The goal is for the parameters to converge to ideal values that cause the managed system response to match the response of the reference model.**

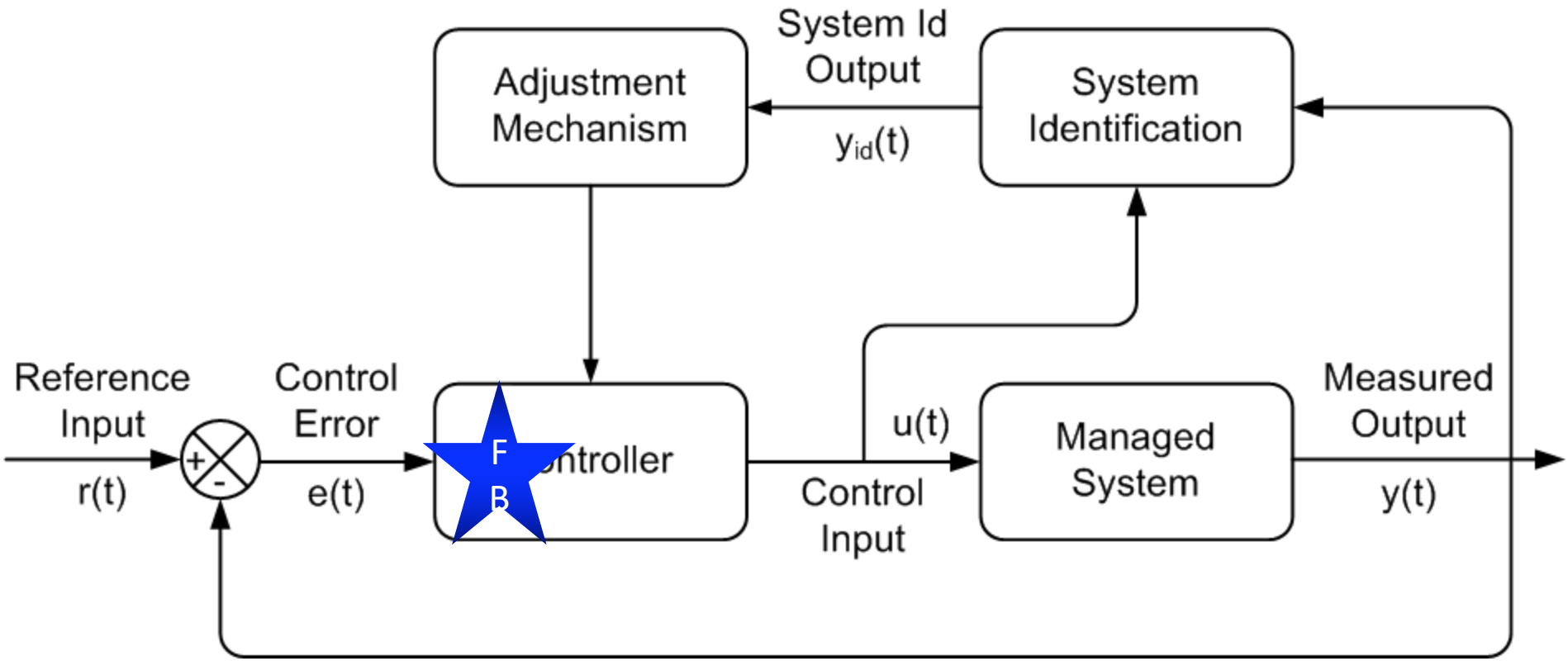
Model Reference Adaptive Controllers—MRAC



Model Identification Adaptive Controllers—MIAC

- **Perform system identification while system is running to modify the control laws**
  - Create model structure and perform parameter estimation using the Least Squares method
- **Cautious adaptive controllers**
  - Use current system identification to modify control law, allowing for system identification uncertainty
- **Certainty equivalent adaptive controllers**
  - Take current system identification to be the true system, assume no uncertainty
  - Nonparametric adaptive controllers
  - Parametric adaptive controllers

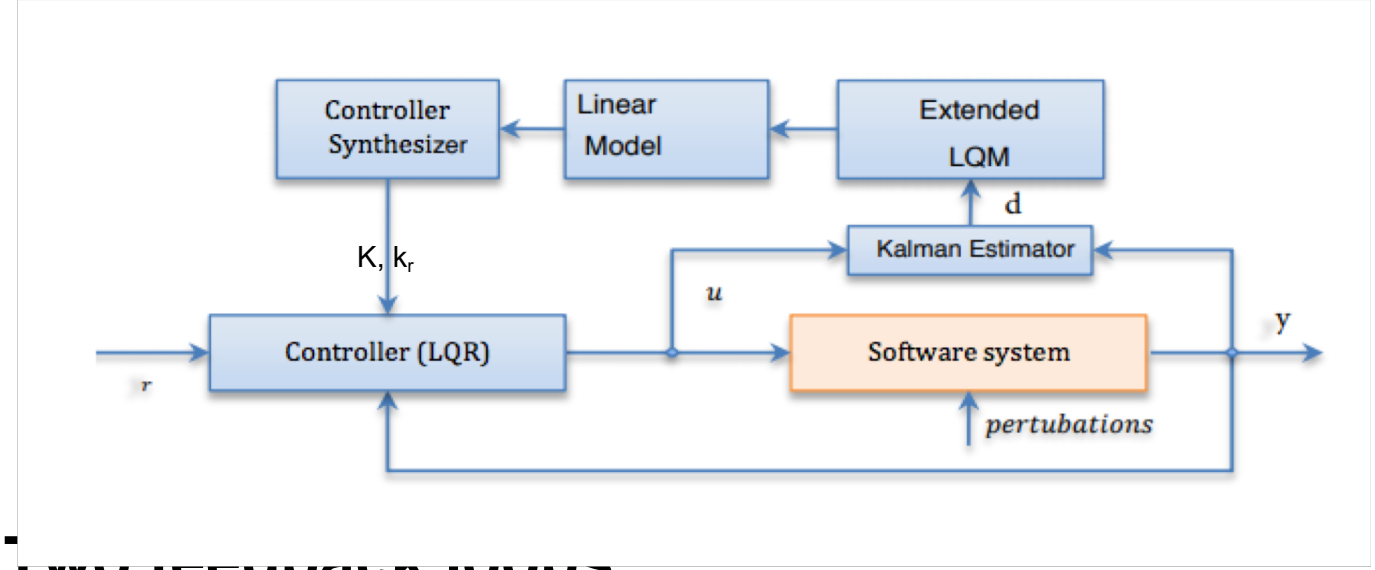
Model Identification Adaptive Controllers—MIAC



MIAC versus MRAC

- **In the MRAC approach, the reference model is static (i.e., given or pre-computed and not changed at run-time)**
- **In the MIAC approach, the reference model is changed at run-time using system identification methods**
- **The goal of both approaches is to adjust the control laws in the controller**

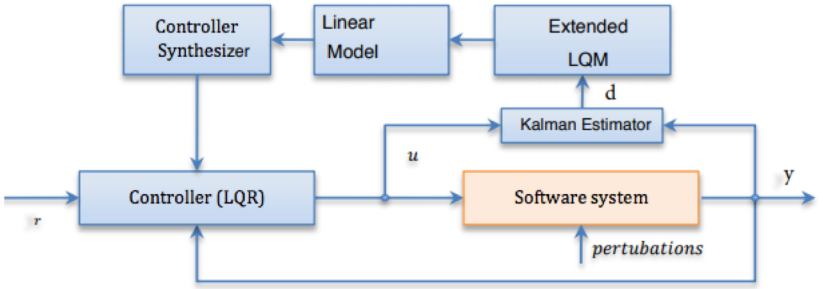
Model Identification Adaptive LQR for Performance Control in Cloud*



- Two feedback loops
  - Model identification and controller synthesis (top)
  - Controller feedback loop (bottom)

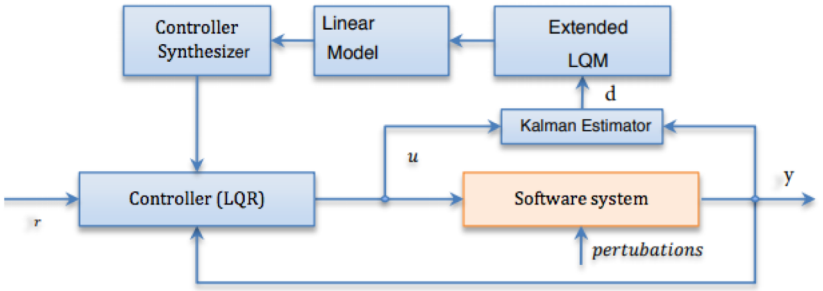
*Cornel Barna, Marin Litoiu, Marios Fokaefs, Mark Shtern, and Joe Wigglesworth. 2018. Runtime Performance Management for Cloud Applications with Adaptive Controllers. In Proceedings of the 2018 ACM/SPEC International Conference on Performance Engineering (ICPE '18). ACM, New York, NY, USA, 176-183.

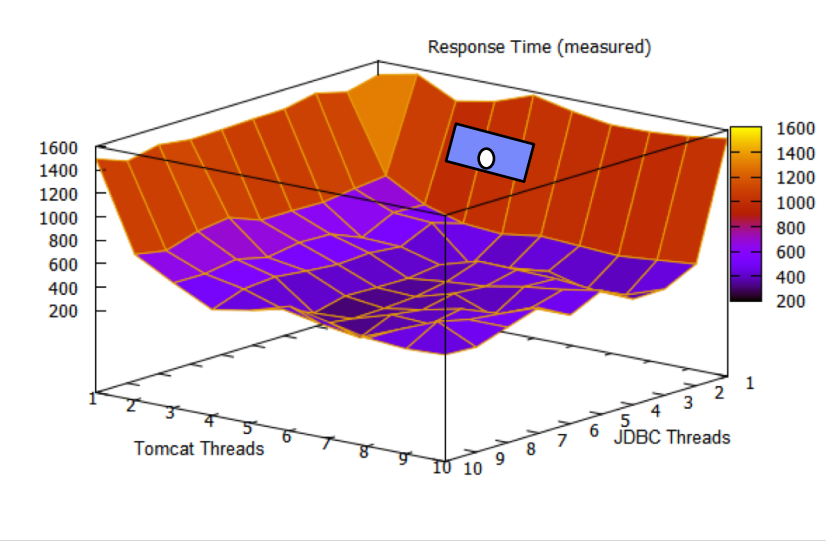
Extended LQM



- **Structure (known)**
  - y=f(u, d, p)
  - **y** is multi-dimensional [response time, throughput, utilization…]
  - **u** is multi-dimensional [threads, no of instances, bandwidth, etc…]
  - **d** service demands in queues
  - **p** workload (no of users, think time per class of request)
- **Uncertainties (unknown) in the model**
  - Model parameter uncertainties
    - use estimators, like Kalman filter, to estimate them at runtime
  - Perturbation uncertainties
    - the controller is designed to address these
  - Non-modeled dynamics
    - Add ghost queues and estimate their parameters at runtime

LQM ->LQR synthesis





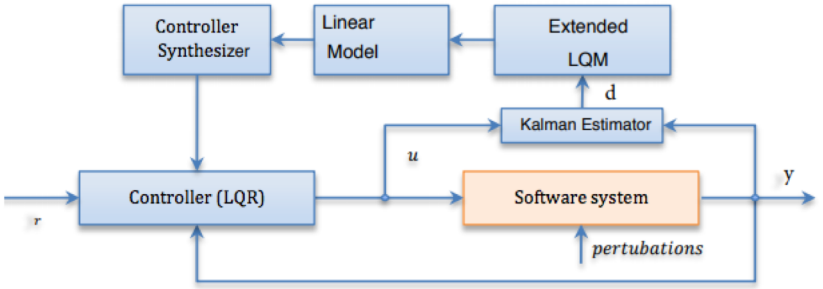
$$\begin{cases} x(k+1) &= Ax(k) + Bu(k) \\ y(k) &= Cx(k) + Du(k) \end{cases}$$

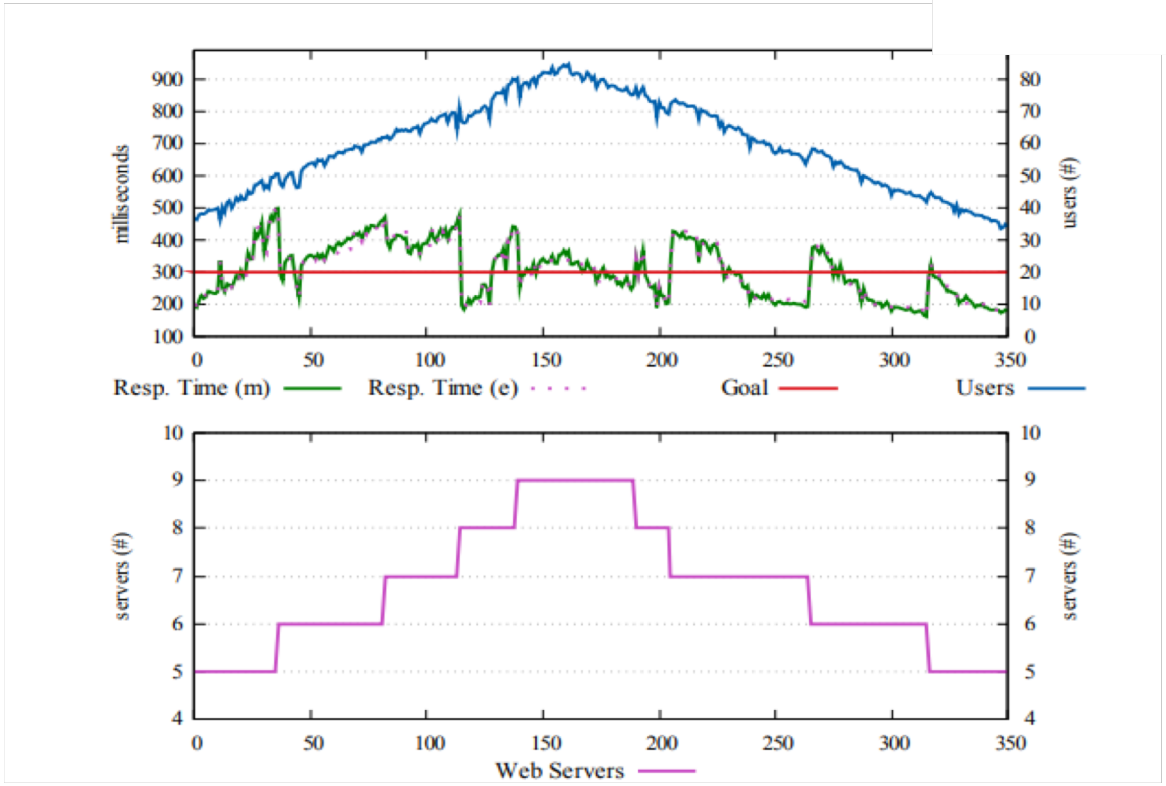
$$K = -Q_u^{-1} B^T P x$$

$$PA + A^T P - PBQ_u^{-1} B^T P + Q_x = 0$$

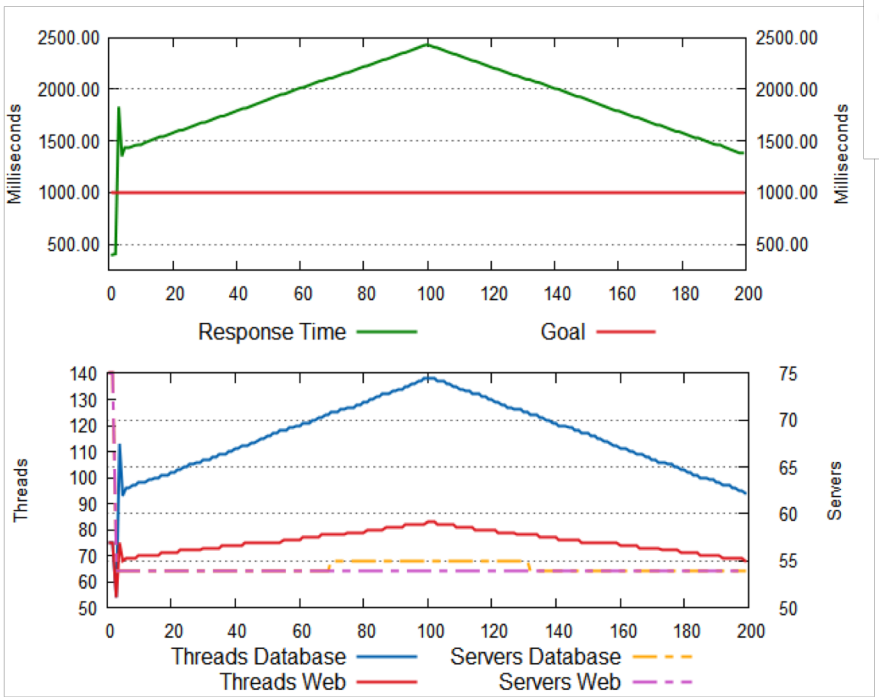
$$1 = C(A - BK)^{-1} Bk_r$$

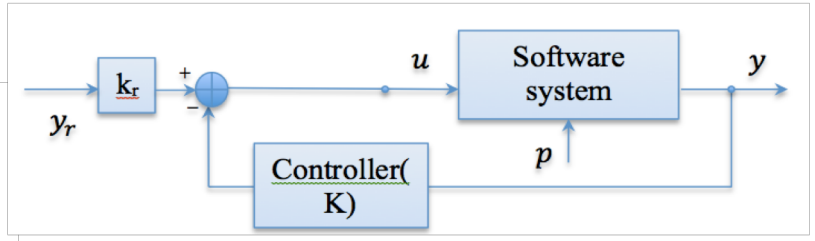
MIAC LQR works!





One Linear Model Fails to Control





Conclusions…

- **Static LQR is not robust**
- **Model Identification Adaptive LQR is robust**
  - Identifies a LQM nonlinear model
  - At runtime, linearizes it around the operational point
  - At runtime, synthesizes LQR controllers
- **Limitations and Lessons Learned**
  - Requires high technical skills to implement
  - Is there an easier way to specify trade-offs between cost and precision?
  - How general can it be?
  - How do we know the bounds of robustness?
  - Does it work for other type of controllers?

Your turn….for next lecture…

- **Compare QNMs with Control Theory Models**
  - What are the advantages and disadvantages of each one?
- **What are some of the challenges in applying Control Theory to software systems?**
  - M Litoiu, M Shaw, G Tamura, NM Villegas, H Müller… What Can Control Theory Teach Us About Assurances in Self-Adaptive Software Systems? Software Engineering for Self-Adaptive Systems 3: …, 2016