Linear **M**odel **P**redictive **C**ontrol for Queueing Networks in Manufacturing and Road Traffic

Yoni Nazarathy

Swinburne University of Technology, Melbourne.

Joint work with:

Erjen Lefeber (manufacturing),

Hai Vu, Bao Vo, Tung Le and Serge Hoogendoorn (road traffic)

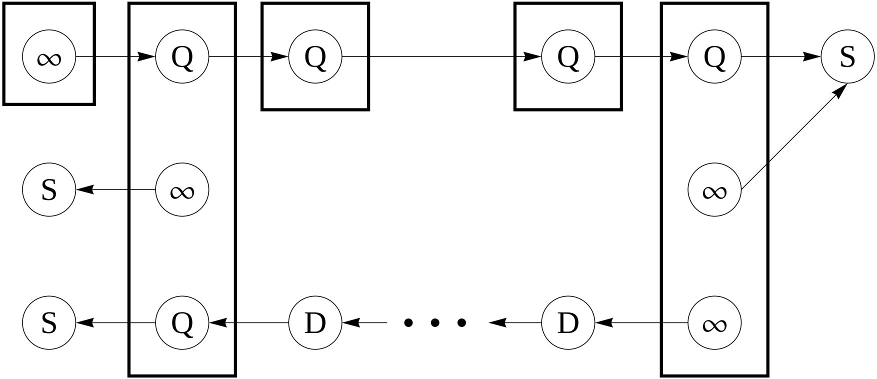
ANZIAM, Jan 30, 2012.

Overview and Main Story

- Central control of:
  - Manufacturing networks
  - Urban traffic networks
- Model as a Queueing Network:
  - Jobs (lots, cars)
  - Servers (manufacturing machines, signaled intersections)
  - Queues (lots waiting for processing, cars waiting at intersections)
  - Routes: Fixed, random, partially controlled, incentive driven
  - Delayed movement
  - **Policy** for **resource allocation** and **routing**
- Desired:
  - High throughout, low WIP, steady output, fairness...
  - Sensible computable control
  - Methodological and mathematical structure of the control
- In this talk:

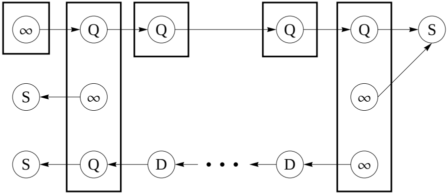
  A control (policy) methodology based on MPC

Most of the talk focuses on the manufacturing setting


∞
Q
Q
Q
Q
S
S
∞
∞
S
Q
D
• • •
D
∞

Precise Queueing Network Model Definition

- Discrete time $n = 0, 1, \ldots$
- K job classes, L servers
- Types of classes:
  - ∞ – Source
  - Q – Queue
  - D – Delay
  - S – Sink
- Deterministic routes
- Randomness due to "batch" arrivals" (∞ classes), $\tilde{u}_k$, m_k
- Processing capacity: jobs per server per time unit, $c_i, i = 1, \ldots L$
- Control Policy – How should servers allocate capacity?



A Controlled Markov Chain

- $P = \{p_{kk'}\}$ – routing matrix, C – constituency matrix
- i.i.d jobs generated at sources, $\tilde{u}^{*U}$ – generic r.v. of U-fold sum
- $\{X(n)\}$ is a controlled Markov chain, with control $U(n) = f(X(n))$

$$X_k(n+1) = \begin{cases} X_k(n) + \sum_{k' \in \mathcal{K}_D} X_{k'}(n) p_{k'k} + \sum_{k' \in \mathcal{K}_{\{Q,\infty\}}} \tilde{u}_{k'}^{*U_{k'}(n)} p_{k'k} - U_k(n), & k \in \mathcal{K}_Q \text{ (queue)} \\ \sum_{k' \in \mathcal{K}_D} X_{k'}(n) p_{k'k} + \sum_{k' \in \mathcal{K}_{\{Q,\infty\}}} \tilde{u}_{k'}^{*U_{k'}(n)} p_{k'k}, & k \in \mathcal{K}_D \text{ (delay)} \\ X_k(n) + \sum_{k' \in \mathcal{K}_D} X_{k'}(n) p_{k'k} + \sum_{k' \in \mathcal{K}_{\{Q,\infty\}}} \tilde{u}_{k'}^{*U_{k'}(n)} p_{k'k}, & k \in \mathcal{K}_S \text{ (sink)} \end{cases}$$

Matrix form

$$\begin{bmatrix} X_Q(n+1) \\ X_D(n+1) \\ X_S(n+1) \end{bmatrix} = \begin{bmatrix} I & P'_{DQ} & 0 \\ 0 & P'_{DD} & 0 \\ 0 & P'_{DS} & I \end{bmatrix} \begin{bmatrix} X_Q(n) \\ X_D(n) \\ X_S(n) \end{bmatrix} + \begin{bmatrix} P'_{\infty Q} M_\infty & P'_{QQ} - I \\ P'_{\infty D} M_\infty & P'_{QD} \\ P'_{\infty S} M_\infty & P'_{QS} \end{bmatrix} \begin{bmatrix} U_\infty(n) \\ U_Q(n) \end{bmatrix} + \begin{bmatrix} P'_{\infty Q} \\ P'_{\infty D} \\ P'_{\infty S} \end{bmatrix} \tilde{u}(U_\infty(n))$$

Elements of $\tilde{u}(\cdot)$ are $\tilde{u}_k^{*U_k(n)} - U_k(n) m_k$

s.t.

$$\begin{bmatrix} 0 & 0 & 0 & -I & 0 \\ 0 & 0 & 0 & 0 & -I \\ -I & 0 & 0 & 0 & I \\ 0 & 0 & 0 & C_\infty & C_Q \end{bmatrix} \begin{bmatrix} X_Q(n) \\ X_D(n) \\ X_S(n) \\ U_\infty(n) \\ U_Q(n) \end{bmatrix} \leq \begin{bmatrix} 0 \\ 0 \\ 0 \\ c \end{bmatrix}$$

Concept: Control as a Linear System

$$X(n+1) = A\,X(n) + B\,U(n) + \text{zero mean noise}$$

s.t.

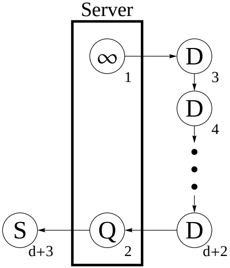
$$F \begin{bmatrix} X(n) \\ U(n) \end{bmatrix} \leq g$$

Our Control Methodology

- Ignore noise
- Assume state and control $(X(\cdot), U(\cdot))$ are continuous in value
- Find a **reference trajectory**
- Apply "standard" control-theoretic methods for **tracking** the reference trajectory
- Use **Model Predictive Control** (MPC) using a Quadratic Programming (QP) formulation

A Structured Example: The Aquisition Queue

D. Denteneer, J. van Leeuwaarden, and I. Adan. The acquisition queue. *Queueing Systems*, 56(3):229240, 2007.



Acquisition Queue with $d = 3$

$$\begin{bmatrix} D_1(n+1) \\ D_2(n+1) \\ D_3(n+1) \\ Q(n+1) \\ S(n+1) \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} D_1(n) \\ D_2(n) \\ D_3(n) \\ Q(n) \\ S(n) \end{bmatrix} + \begin{bmatrix} m & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & -1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} U_\infty(n) \\ U_Q(n) \end{bmatrix}$$

Maximal Throughput

$$\delta = \lim_{n\to\infty} \frac{1}{n} S(n) = \left(\frac{1}{m} + 1\right)^{-1} c$$

A Reference Trajectory

$$U^r_\infty(n) = U^r_Q(n) = D^r_1(n) = D^r_2(n) = D^r_3(n) = \delta, \quad Q^r(n) = \text{const}, \quad S^r(n) = \delta\, n$$

Error Dynamics: $X^e(n) = X(n) - X^r(n)$, $U^e(n) = U(n) - U^r(n)$

Also satisfy: $\quad X^e(n+1) = A\, X^e(n) + B\, U^e(n)$

Our controller tries to regulate $X^e(n)$ on 0

The MPC Approach

Action of Controller at Time n

- Look at $X^e(n)$
- Plan an optimal schedule for a **time horizon** of N time units:
  - Optimize the variables $U^e(n), \ldots, U^e(n+N-1)$
  - These yield predictions of $X^e(n+1), \ldots, X^e(n+N)$
  - Practical objective (QP):

$$\sum_{i=n}^{n+N-1} \hat{X}^e(i+1)' Q \hat{X}^e(i+1) + U^e(i)' R U^e(i)$$

- After optimizing – **use first step**:
  - $U(n) = U^e(n) + U^r(n)$
  - Round off $U(n)$ and insure feasibility
- Repeat in next time step

Parameters of controller:

Time horizon, N. Positive definite cost matrixes, Q, R. Reference trajectory.

Precise Formulation of the QP (for illustration)

$$\min_{\underline{\mathrm{U}}^e} \quad \underline{\mathrm{U}}^{e\prime}\,(\underline{\mathrm{B}}'\,\underline{\mathrm{Q}}\,\underline{\mathrm{B}}+\underline{\mathrm{R}})\underline{\mathrm{U}}^e+2X_0^{e\prime}\,\underline{\mathrm{A}}'\,\underline{\mathrm{Q}}\,\underline{\mathrm{B}}\,\underline{\mathrm{U}}^e$$

s.t.

$$\begin{bmatrix} \underline{\mathrm{C}} \\ \underline{\mathrm{S}}^1_{UQ} \\ \underline{\mathrm{S}}^+_{UQ}-\underline{\mathrm{S}}^-_{XQ}\underline{\mathrm{B}} \\ -I \end{bmatrix}\underline{\mathrm{U}}^e \le \begin{bmatrix} \underline{\mathrm{c}} \\ 0 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ \underline{\mathrm{S}}^1_{XQ} \\ \underline{\mathrm{S}}^-_{XQ} \\ 0 \end{bmatrix}\underline{\mathrm{X}}^r + \begin{bmatrix} -\underline{\mathrm{C}} \\ -\underline{\mathrm{S}}^1_{UQ} \\ -\underline{\mathrm{S}}^+_{UQ} \\ I \end{bmatrix}\underline{\mathrm{U}}^r + \begin{bmatrix} 0 \\ S_{XQ} \\ \underline{\mathrm{S}}^-_{XQ}\underline{\mathrm{A}} \\ 0 \end{bmatrix}X_0^e$$

$\underline{\mathrm{Q}}$, $\underline{\mathrm{R}}$ are block diagonal matrixes of Q and R. The S matrixes "select" elements. The following matrixes are used for prediction:

$$\underline{\mathrm{A}} = \begin{bmatrix} A \\ A^2 \\ \vdots \\ A^N \end{bmatrix}, \quad \underline{\mathrm{B}} = \begin{bmatrix} B & 0 & \cdots & 0 \\ AB & B & & \vdots \\ \vdots & & \ddots & \\ A^{N-1}B & & \cdots & B \end{bmatrix}$$

Observe: If U^r is constant as well as X^r on the Q-classes then control law is a function of X_0^e only

Numerical Illustration

Acquisition Queue Threshold vs. MPC

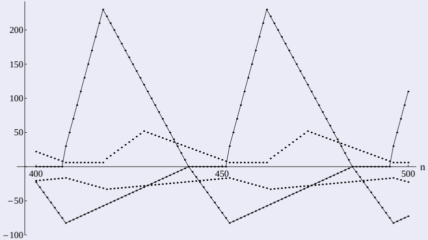
Example: $c = 10$, $d = 10$, $m = 3$

A Simple Threshold Control Law (Van Leeuwaarden et. al. 2007)

$$U_\infty(n) = \alpha + (c - Q(n))^+, \quad U_Q(n) = c - U_\infty(n)$$

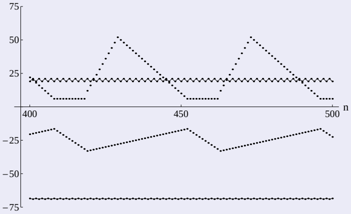
$\alpha < c/(1+m)$ (for stability)

Assume no noise, optimize α: $\alpha = 2$ is best

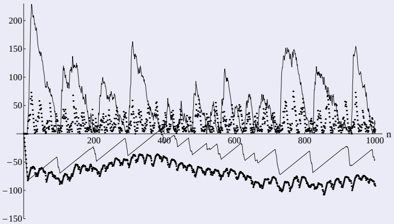


MPC Appears Better than Threshold

Compare $\alpha = 2$ with MPC ($Q = I$, $R = I$, $N = 30$)



Add noise (Geometrically distributed acquisitions)



What about road traffic?

Model Modifications for Urban Traffic Control

- Time-varying arrivals
- Servers operate on multiple queues in parallel (traffic phases)
- Free flow in links up to intersections
- Route selection of driver in two ways:
  - Fixed proportions of flow out of every intersection
  - Centrally controlled directives yielding a traffic split
- Spil-backs play a major role
- Instead of tracking a trajectory, minimization of $\Big(\text{cars in system}\Big)^2$ makes sense

Key point: We can do all of the above, yet maintain the linear QP structure for MPC

Example: A More Complex Manufacturing Network

Example: A More Complex Manufacturing Network

Use Linear Program (LP) to Find a Reference Trajectory

$$\max \quad \sum_{i=1}^{4} w_i r_i$$

s.t.

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 1/m_3 & 1 \\ 1 & 1/m_2 & 0 & 1/m_4 \end{bmatrix} \begin{bmatrix} r_1 \\ r_2 \\ r_3 \\ r_4 \end{bmatrix} \leq \begin{bmatrix} c_1 m_1 \wedge c_3 \wedge c_4 \\ c_2 \\ c_5 \end{bmatrix},$$

$$r_i \geq 0, \quad i = 1, 2, 3, 4.$$

$\rho \in [0, 1]$

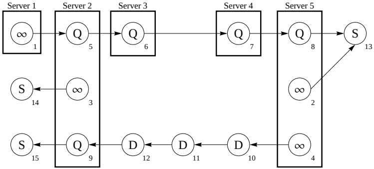
(route 1) $U_1^r(n) = \rho r_1^* / m_1, \quad X_5^r(n) = X_6^r(n) = X_7^r(n) = X_8^r(n) = U_5^r(n) = U_6^r(n) = U_7^r(n) = U_8^r(n) = \rho r_1^*,$

(route 2) $U_2^r(n) = \rho r_2^* / m_2,$

(route 3) $U_3^r(n) = \rho r_3^* / m_3,$

(route 4) $U_4^r(n) = \rho r_4^* / m_4, \quad X_{10}^r(n) = X_{11}^r(n) = X_{12}^r(n) = X_9^r(n) = U_9^r(n) = \rho r_4^*,$

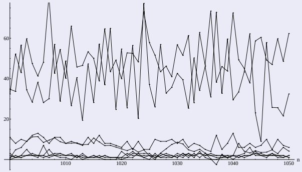
(sinks) $X_{13}^r(n) = \rho(r_1^* + r_2^*)n, \quad X_{14}^r(n) = \rho r_3^* n, \quad X_{15}^r(n) = \rho r_4^* n.$



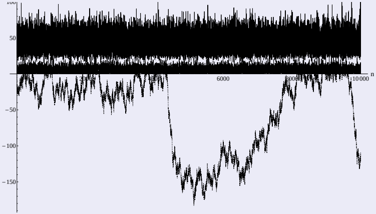
A More Complex Network (cont.)

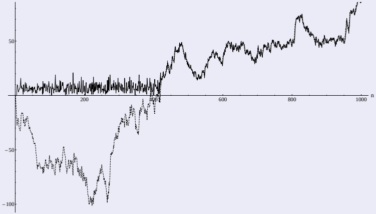
Parameters set such that $r_i^* > 0$, $i = 1, 2, 3, 4$, $Q = I$, $R = I$, $\rho = 1$

$N = 5$, Stable



$N = 4$, Stable in Q's (not in output). $N = 3$ Unstable





Stability?

Continuous Deterministic Case

- Add **end point constraint**: $X^e(N) = 0$
- Main Theoretical Result: If feasible solution exists then resulting system is asymptotically stable
- Alternative: Take $N = \infty$

Discrete Stochastic Case

- No general result
- Some hope of proving positive recurrence for "toy examples" by analyzing the solution of the QP when X_0^e is far from the origin

Conclusion

- Main idea: View queueing network as controlled linear system with self generated noise – apply MPC – appears to "work well"
- General theory: e.g. Stability properties ... "hard to obtain"
- At least... Hope for asymptotic explicit stochastic analysis of some toy examples
- To do: Set up "observers" (possibly using the noise structure)