Exploring Model Predictive Control for Queueing Networks

Johan van Leeuwaarden\textsuperscript{a,b}, Erjen Lefeber\textsuperscript{c,*}, Yoni Nazarathy\textsuperscript{b,c,*} and Koos Rooda\textsuperscript{c}.

Twente Industrial Engineering Seminar,
July 6, 2010.

\textsuperscript{a}Department of Mathematics and Computer Science, Eindhoven University of Technology, The Netherlands.
\textsuperscript{b}EURANDOM, Eindhoven University of Technology, The Netherlands.
\textsuperscript{c}Department of Mechanical Engineering, Eindhoven University of Technology, The Netherlands.

*Supported by the Netherlands Organization for Scientific Research (NWO-VIDI grant 639.072.072).
Overview and Main Story: Control of Queueing Networks

- Queueing Networks: Jobs, Servers, Queues, Routes, Scheduling Policy
- Desired:
  - High throughout, low WIP, steady output
  - Sensible computable control
  - Methodological and mathematical structure of the control
- In this talk: A control methodology based on MPC
- Not yet: Theory, stability, adaptiveness, robustness, observers...

Exploring Model Predictive Control for Queueing Networks
Our Queueing Network Models

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

A Single Server Queue, \((c_1, c_2, m_1)\)

The Acquisition Queue – Similar to – A Simple Re-Entrant Line

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 $U(n) = f(X(n))$

$$X_k(n + 1) = \begin{cases} 
X_k(n) + \sum_{k' \in K_D} X_{k'}(n)p_{k'k} + \sum_{k' \in K\{Q,\infty\}} \tilde{u}_{k'}^{\ast U_k(n)} p_{k'k} - U_k(n), & k \in K_Q \text{ (queue)} \\
\sum_{k' \in K_D} X_{k'}(n)p_{k'k} + \sum_{k' \in K\{Q,\infty\}} \tilde{u}_{k'}^{\ast U_k(n)} p_{k'k}, & k \in K_D \text{ (delay)} \\
X_k(n) + \sum_{k' \in K_D} X_{k'}(n)p_{k'k} + \sum_{k' \in K\{Q,\infty\}} \tilde{u}_{k'}^{\ast U_k(n)} p_{k'k}, & k \in K_S \text{ (sink)} 
\end{cases}$$

Matrix form

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

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

s.t.

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

Exploring Model Predictive Control for Queueing Networks
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

Exploring Model Predictive Control for Queueing Networks
### Illustrative Example: 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

\[
\delta = \lim_{n \to \infty} \frac{1}{n} S(n) = m \frac{c}{1 + m},
\]

### A Reference Trajectory

\[
D_1^*(n) = D_2^*(n) = D_3^*(n) = Q^*(n) = U_\infty^*(n) = U_Q^*(n) = \delta, \quad S^*(n) = \delta n
\]

### Error Dynamics: $X_e(n) = X(n) - X^*(n)$, $U_e(n) = U(n) - U^*(n)$

Also satisfy $X_e(n+1) = A X_e(n) + B U_e(n)$

Our controller needs 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: Time horizon, $N$. Positive definite cost matrixes, $Q$, $R$
Precise Formulation of the QP (for illustration)

\[
\begin{align*}
\text{min}_{U^e} & \quad U^e' (B' Q B + R) U^e + 2 X_0^e' A' Q B U^e \\
\text{s.t.} & \quad \begin{bmatrix} C \\ S_{UQ}^+_1 - S_{XQ}^- B^- I \end{bmatrix} U^e \leq \begin{bmatrix} c \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ S_{XQ}^1 \\ S_{XQ}^- \end{bmatrix} X^r + \begin{bmatrix} -C \\ -S_{UQ}^1 \\ -S_{UQ}^- \end{bmatrix} U^r + \begin{bmatrix} 0 \\ S_{XQ} \\ S_{XQ} A \\ 0 \end{bmatrix} X^e
\end{align*}
\]

Q, R are block diagonal matrixes of Q and R. The S matrixes "select" elements. The following matrixes are used for prediction:

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

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

Exploring Model Predictive Control for Queueing Networks
Possibility for ”Explicit Form” the Solution

Multi-Parametric Quadratic Programming (MPQP)

Algorithms for an ”explicit solution” in terms of \( X_0^e \):
A piece-wise affine function

Example: Single Server Queue with \( c_1 = \infty \), \( c_2 = 20 \), \( m_1 = 1 \). \( Q = I \), \( R = I \), \( N = 5 \).

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
- Practical alternative: Use end point constraint. When QP is not feasible, don’t work on $\infty$ classes
Numerical Examples
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)

Example: \( d = 10, m = 3, \) no noise

Assume no noise, optimize \( \alpha: \alpha = 2 \) is best

Exploring Model Predictive Control for Queueing Networks
MPC Appears Better than Threshold

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

Add noise (Geometrically distributed acquisitions)
Simple Re-Entrant Line \((\rho = 1)\)

\[ c_1 = 10, \ c_2 = 100, \ m_1 = 20, \ \delta = m_1 \frac{c_1}{1+m_1} \approx 9.52 \]

Comparing LBFS and MPC \(Q = 1, \ R = I, \ N = 10\)

Note: Both controllers achieve desired throughput

Different Objective, \(Q = 1\)

Note: MPC does NOT achieve desired throughput
A More Complex Network

Use Linear Program (LP) to Find a Reference Trajectory

\[
\begin{align*}
\text{max} & \quad \sum_{i=1}^{4} w_i r_i \\
\text{s.t.} & \quad \begin{bmatrix}
1 & 0 & 0 & 0 \\
1 & 0 & 1/m_3 & 0 \\
1 & 1/m_2 & 0 & 1/m_4 \\
\end{bmatrix}
\begin{bmatrix}
r_1 \\
r_2 \\
r_3 \\
r_4 \\
\end{bmatrix} \leq 
\begin{bmatrix}
c_1 m_1 \land c_3 \land c_4 \\
c_2 \\
c_5 \\
\end{bmatrix}, \\
r_i \geq 0, \quad i = 1, 2, 3, 4.
\end{align*}
\]

\(\rho \in [0, 1]\)

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

Exploring Model Predictive Control for Queueing Networks
Conclusion

- Main idea: View queueing network as controlled linear system with noise – apply MPC – appears to "work well"
- General theory: (1) Stability properties (2) Bounds on performance... "hard to obtain"
- At least... Hope for explicit stochastic analysis of some toy examples
- Immediate extensions (in progress): (1) Observers (2) inverse optimality of other controllers - e.g. dead beat
- Interesting to experiment: Incorporating the effect of noise in the reference value for the queue levels
Questions?