MAE 598: Multi-Robot Systems
Fall 2016

Instructor: Spring Berman
spring.berman@asu.edu

Assistant Professor, Mechanical and Aerospace Engineering
Autonomous Collective Systems Laboratory
http://faculty.engineering.asu.edu/acs/

Lecture 7
Controller Synthesis using the Macroscopic Model

Optimization

Compute the $k_{ij}$ that minimize a measure of the model’s convergence time to $x^d$

Convex optimization approaches:

- Multi-affine model

Relaxation times: time to return to equilibrium after perturbation

[Heinrich and Schuster, *The Regulation of Cellular Systems*, 1996]

- Estimated by linearizing the model around $x^d$

- Linear model

Eigenvalues of $K$ govern rate of convergence (ROC)

- Tradeoff between fast ROC and few task transitions at equilibrium
Hybrid System Macroscopic Models

Controller Synthesis: Vector Fields on Polytopes

\[ \dot{x} = f(x) + Bu \quad P_S = \text{polytope in} \quad \mathbb{R}^S \]

Compute \( u \) that steers \( x \) to a facet of \( P_S \) in finite time

[Habets, van Schuppen, Automatica ’04, Belta et al., CDC’02]
Swarm Robotic Assembly System

[Loic Matthey, Spring Berman, and Vijay Kumar. “Stochastic Strategies for a Swarm Robotic Assembly System.” *ICRA 2009.*]

Design a reconfigurable manufacturing system that quickly assembles target amounts of products from a supply of heterogeneous parts.
Approach

ODEs are functions of probabilities of assembly and disassembly:
Optimize for fast assembly of target amounts of products

Robots start assemblies and perform disassemblies according to optimized probabilities

\[ \dot{x} = MKy(x) \]
Decisions Modeled as Chemical Reactions

\[ X_R + X^u_i \xrightarrow{e_i} X^c_i \quad i = 1, \ldots, 8 \]

\[ X^c_i + X^c_m \xrightarrow{k_j^+} X^c_n + X_R \]

\[ X^c_n \xrightarrow{k_j^-} X^c_i + X^u_m \]

\[ e_i = A(p^e), \quad k_j^+ = A(p^e)p^a_j\, p^+_j, \quad k_j^- = p^-_j \]

\[ p^e = \text{prob. that a robot encounters a part or another robot} \approx \frac{v_{robot}TW_{comm}}{A} \]

\[ A = \text{arena area} \]

[Correll and Martinoli, Coll. Beh. Workshop, ICRA 2007]
Decisions Modeled as Chemical Reactions

\[ X_R + X_{i}^{u} \xrightarrow{e_i} X_{i}^{c} \quad i = 1, \ldots, 8 \]

\[ X_{l}^{c} + X_{m}^{c} \xrightarrow{k_{j}^{+}} X_{n}^{c} + X_{R} \]

\[ X_{n}^{c} \xrightarrow{k_{j}^{-}} X_{l}^{c} + X_{m}^{u} \]

\[ e_{i} = A \, p^{e}, \quad k_{j}^{+} = A \, p^{e} \, p_{j}^{a} \, p_{j}^{+}, \quad k_{j}^{-} = p_{j}^{-} \]

\[ p_{j}^{a} = \text{prob. of two robots successfully completing assembly process } j \]

(measured from simulations)
Decisions Modeled as Chemical Reactions

\[ X_R + X_i^u \xrightarrow{e_i} X_i^c \quad i = 1, \ldots, 8 \]

\[ X_i^c + X_m^c \xrightarrow{k_j^+} X_n^c + X_R \]

\[ X_n^c \xrightarrow{k_j^-} X_i^c + X_m^u \]

\[ e_i = A \, p^e \quad , \quad k_j^+ = A \, p^e \, p_j^a \, p_j^+ \quad , \quad k_j^- = p_j^- \]

Tunable:

\[ p_j^+ = \text{prob. of two robots starting assembly process } j \]

\[ p_j^- = \text{prob. per unit time of a robot performing disassembly process } j \]
Mapping \( p_i^+, p_i^- \) onto the Robot Controllers

\[ \Delta t = \text{simulation timestep (32 ms)} \]

\[ u = \text{random number uniformly distributed over [0,1]} \]

Robot computes \( u \) at each \( \Delta t \), disassembles the part if

\[ u < p_i^- \Delta t \]

Robot computes \( u \), executes assembly if

\[ u < p_i^+ \]
Reduced Macroscopic Model

Lower-dimensional model (abstract away robots):

\[
\begin{align*}
X_1 + X_2 & \xrightleftharpoons[k_1^-]{k_1^+} X_5 \\
X_3 + X_4 & \xrightleftharpoons[k_2^-]{k_2^+} X_6 \\
X_5 + X_6 & \xrightleftharpoons[k_3^-]{k_3^+} X_7 \\
X_2 + X_7 & \xrightleftharpoons[k_4^-]{k_4^+} X_{F1} \\
X_2 + X_5 & \xrightleftharpoons[k_5^-]{k_5^+} X_8 \\
X_6 + X_8 & \xrightleftharpoons[k_6^-]{k_6^+} X_{F2}
\end{align*}
\]

Vector of complexes: \[ y(x) = [x_1 x_2 \ x_5 \ x_3 x_4 \ x_6 \ x_2 x_7 \ x_{F1} \ x_5 x_6 \ x_7 \ x_2 x_5 \ x_8 \ x_6 \ x_8 \ x_{F2}]^T \]

\[
\dot{x} = MKy(x)
\]

We also define a matrix \( M \in \mathbb{R}^{10 \times 12} \) in which each entry \( M_{ji} \), \( j = 1, \ldots, 10 \), of column \( m_i \) is the coefficient of part type \( j \) in complex \( i \) (0 if absent). We relabel the rate associated with reaction \( (i, j) \in \mathcal{E} \) as \( k_{ij} \) and define a matrix \( K \in \mathbb{R}^{12 \times 12} \) with entries

\[
K_{ij} = \begin{cases} 
  k_{ji} & \text{if } i \neq j, \ (j, i) \in \mathcal{E}, \\
  0 & \text{if } i \neq j, \ (j, i) \notin \mathcal{E}, \\
  -\sum_{(i, l) \in \mathcal{E}} k_{il} & \text{if } i = j.
\end{cases}
\]

Conservation constraints:

\[
\begin{align*}
x_3 - x_4 &= N_1 \\
x_1 + x_5 + x_7 + x_8 + x_{F1} + x_{F2} &= N_2 \\
x_2 + x_5 + x_7 + 2(x_8 + x_{F1} + x_{F2}) &= N_3 \\
x_3 + x_6 + x_7 + x_{F1} + x_{F2} &= N_4
\end{align*}
\]
Reduced Macroscopic Model

\[ \dot{x} = MKy(x) \]

\[
\begin{align*}
x_3 - x_4 &= N_1 \\
x_1 + x_5 + x_7 + x_8 + x_{F1} + x_{F2} &= N_2 \\
x_2 + x_5 + x_7 + 2(x_8 + x_{F1} + x_{F2}) &= N_3 \\
x_3 + x_6 + x_7 + x_{F1} + x_{F2} &= N_4
\end{align*}
\]

The system has a unique, positive, globally asymptotically stable equilibrium.

Proof: Reaction network is deficiency zero and weakly reversible, does not admit equilibria with some \( x_i = 0 \)

\[ \rightarrow \] We can design \( K \) such that the system always converges to a target equilibrium, \( x^d > 0 \)
Design of Optimal $p_i^+, p_i^-$

\[ \dot{x} = MKy(x) \]

\[ \begin{align*}
  x_3 - x_4 &= N_1 \\
  x_1 + x_5 + x_7 + x_8 + x_{F1} + x_{F2} &= N_2 \\
  x_2 + x_5 + x_7 + 2(x_8 + x_{F1} + x_{F2}) &= N_3 \\
  x_3 + x_6 + x_7 + x_{F1} + x_{F2} &= N_4
\end{align*} \]

Recall that $K$ is a function of $p$, the vector of $p_j^+, p_j^-$

\[ k_j^+ = A \, p^e \, p_j^a \, p_j^+ , \quad k_j^- = p_j^- \]

Select $x^d$ that satisfies conservation constraints

Compute $p$ that minimizes the system convergence time to $x^d$ subject to constraints:

\[ MK(p)y(x^d) = 0, \quad 0 \leq p \leq 1 \]
Optimization Problems

I. Linear Program

Objective: Maximize the average inverse relaxation time $\tau_j$

- $\tau_j = \text{time for system mode to return to equilibrium after perturbation}$
- Estimated by linearizing the ODE model around $x^d$

[Heinrich and Schuster, The Regulation of Cellular Systems, 1996]

For reaction $X_k + X_l \rightarrow_{\frac{k_j^+}{k_j^-}} X_m$ : $\tau_j^{-1} = k_j^+(x_k^d + x_l^d) + k_j^-$

II. Monte Carlo Method

Objective: Minimize time for system to reach $0.1\left\| x^0 - x^d \right\|_2$
Optimization Improves Convergence Rate

- 15 robots, 15 basic parts
- Simulations averaged over 30 runs

\[ x_{F1} / (x_{F1}^d + x_{F2}^d) \]
\[ x_{F2} / (x_{F1}^d + x_{F2}^d) \]

\[ \alpha = 0.5 \]
\[ (1 - \alpha) = 0.5 \]

Randomly selected \( p_i^+, p_i^- \)

\( p_i^+, p_i^- \) from Linear Program

\( p_i^+, p_i^- \) from Monte Carlo

ASU MAE 598 Multi-Robot Systems  Berman
Linearization is most effective for $\alpha \approx 0.2 - 0.5$

- For all $\alpha$, linear program only changes rates of disassembling F1, F2

- Monte Carlo $p_i^+, p_i^-$ yield fastest convergence but takes $\sim 10$ hrs to compute (in 2009), vs. $<1$ s using the linear program (2 GHz laptop)

\[ \alpha = \frac{x_{F1}^d}{(x_{F1}^d + x_{F2}^d)} \]

Time for reduced model to reach $0.1 \| x^0 - x^d \|_2$

- 300 basic parts

$p_i^+, p_i^-$ from:

- Random selection (Average of 100 values)
- Linear Program
- Monte Carlo Method
Linearization is most effective for $\alpha \approx 0.2 – 0.5$

- 15 robots, 15 basic parts
- Simulations averaged over 30 runs

\begin{align*}
  x_{F1} / (x_{F1}^d + x_{F2}^d) \\
  x_{F2} / (x_{F1}^d + x_{F2}^d)
\end{align*}

Randomly selected $p_i^+, p_i^-$

\[ p_i^+, p_i^- \text{ from Linear Program} \]

\[ p_i^+, p_i^- \text{ from Monte Carlo} \]
Linearization is most effective for $\alpha \approx 0.2 – 0.5$

- 50 robots, 50 basic parts
- Simulations averaged over 20 runs

\[
\begin{align*}
\frac{x_{F1}}{x_{F1}^d + x_{F2}^d} & \quad \text{Simulation} \\
\frac{x_{F2}}{x_{F1}^d + x_{F2}^d} & \quad \text{Model} \\
\alpha & = 0.5 \\
(1-\alpha) & = 0.5
\end{align*}
\]

Randomly selected $p_i^+, p_i^-$

$p_i^+, p_i^-$ from Linear Program

$p_i^+, p_i^-$ from Monte Carlo

Simulations averaged over 20 runs.

\[
\begin{align*}
\frac{F_1}{x_{F1}^d + x_{F2}^d} & \quad \text{Simulation} \\
\frac{F_2}{x_{F1}^d + x_{F2}^d} & \quad \text{Model} \\
\alpha & = 0.5 \\
(1-\alpha) & = 0.5
\end{align*}
\]
Swarm Multi-Site Deployment

Swarm Multi-Site Deployment

- Model interconnection topology of sites as a directed graph
  \[ G = (\mathcal{V}, \mathcal{E}) \quad \mathcal{V} = \text{set of sites} \quad \mathcal{E} = \{ (i, j) \in \mathcal{V} \times \mathcal{V} \mid i \sim j \} \]
  - Assume that \( G \) is strongly connected (directed path btwn. each pair of sites)
  - Choose for rapid, efficient redistribution

- Assume that each robot:
  - knows \( G \), all \( k_{ij} \), task at each site
  - can navigate between sites
  - can sense neighboring robots

\( k_{ij} \) = Transition probability per unit time for one robot at site \( i \) to travel to site \( j \)
Approach

- Ordinary differential equations in terms of $k_{ij}$ and the fraction of robots $x_i$ at each site $i$

- $N$ robots, $M$ behavior states: \{Doing task at site 1, Doing task at site 2, ..., Doing task at site $M$\}

- Could also include states that represent travel between pairs of sites

\[ \dot{x} = -Kx \]

Abstraction


\[ N \rightarrow \infty \]
Approach

- Analysis and optimization tools to choose $k_{ij}$

"Top-down" controller synthesis approach is computationally inexpensive and gives guarantees on performance

- Switch according to $k_{ij}$; motion control for tasks at sites, navigation
Macroscopic Model

\[ x_i(t) = \text{Fraction of robots at site } i \text{ at time } t \]

\[ \dot{x}_i(t) = \sum_{j \neq i} k_{ji} x_j(t) - \sum_{i \neq j} k_{ij} x_i(t) \]

\[ \dot{x} = -Kx \]

(a) \( K^T 1 = 0 \),

(b) \( K_{ij} \leq 0 \quad \forall (i, j) \in E \)

Conservation constraint: \( 1^T x = 1 \)
Macroscopic Model

\[
\dot{x} = -Kx \quad 1^T x = 1
\]

(a) \( K^T 1 = 0 \), (b) \( K_{ij} \leq 0 \ \forall (i, j) \in \mathcal{E} \)

- There is a unique, stable equilibrium [Proof uses Perron-Frobenius Theorem]

\[ x_i^d = \text{Target fraction of robots at site } i \]

\[ x^d = [x_1^d \ldots x_M^d]^T \]

→ If \( k_{ij} \) are chosen so that (c) \( Kx^d = 0 \), the system always converges to the target distribution
Macroscopic Model

\[ \dot{x} = -Kx \quad \text{and} \quad 1^T x = 1 \]

(a) \( K^T 1 = 0 \),  (b) \( K_{ij} \leq 0 \quad \forall (i, j) \in E \),  (c) \( Kx^d = 0 \)

- Real parts of eigenvalues of \( K \) govern rate of convergence to \( x^d \)
  - \( \rightarrow \) High \( k_{ij} \) for fast redistribution
- Probability that a robot at \( i \) starts moving to \( j \) in a time step is proportional to \( k_{ij} \)
  - \( \rightarrow \) Low \( k_{ij} \) for few idle trips between sites at equilibrium

Optimal \( K \) maximizes convergence rate of system subject to a constraint on inter-site traffic at equilibrium
Macroscopic Model

\[ \dot{x} = -Kx \quad 1^T x = 1 \]

(a) \( K^T 1 = 0 \) , (b) \( K_{ij} \leq 0 \) \( \forall (i, j) \in \mathcal{E} \) , (c) \( Kx^d = 0 \)

Traffic along edge \( (i, j) = k_{ij} x_i \) (fraction of robots per unit time exiting \( i \) to go to \( j \))

Possible constraints on inter-site traffic at equilibrium:

(1) Total limit: \( \sum_{(i, j) \in \mathcal{E}} k_{ij} x_i^d \leq c_{tot} \) or

(2) Edge limits: \( k_{ij} x_i^d \leq c_{ij} , \ (i, j) \in \mathcal{E} \)
Design of Optimal $K$ Matrix

- Maximize a measure of the convergence rate of model
  $$\dot{x} = -Kx$$
  subject to one of the 2 constraints on equilibrium traffic

- Measure the degree of convergence to $x^d$ in terms of the fraction of misplaced robots,
  $$\mu_n(x) = \|x - x^d\|_n \quad n = 1 \text{ or } 2$$

- One problem minimizes convergence time directly using a Monte Carlo method; the others maximize functions of the eigenvalues of $K$ using convex optimization
# K Matrix Optimization Problems

<table>
<thead>
<tr>
<th>Prob.</th>
<th>Objective</th>
<th>FC</th>
<th>DB</th>
<th>$x^0$</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>P1a</strong></td>
<td>Maximize asymptotic ROC</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>P1b</strong></td>
<td>Maximize asymptotic ROC</td>
<td></td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td><strong>P2</strong></td>
<td>Maximize overall ROC</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>P3</strong></td>
<td>Minimize time to reach $0.1\mu_2(x^0)$</td>
<td></td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td><strong>P4</strong></td>
<td>Maximize ROC along $x^d - x^0$</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
</tbody>
</table>

$FC = \text{fully connected (each site accessible from all other sites)}$

$DB = \text{detailed balance condition holds}$

$x^0 = \text{initial distribution known}$

$ROC = \text{rate of convergence}$
Optimal K Comparison

<table>
<thead>
<tr>
<th>Graph A</th>
<th>Graph B</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total limit</td>
<td>Total limit</td>
</tr>
<tr>
<td>[</td>
<td>x - x^d</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>P1a</td>
<td>P2</td>
</tr>
<tr>
<td>0.34</td>
<td>0.44</td>
</tr>
<tr>
<td>P3</td>
<td>P3</td>
</tr>
<tr>
<td>0.33</td>
<td>0.46</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Graph A</th>
<th>Graph B</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total limit</td>
<td>Total limit</td>
</tr>
<tr>
<td>[</td>
<td>x - x^d</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>P1b</td>
<td>P4</td>
</tr>
<tr>
<td>0.49</td>
<td>0.50</td>
</tr>
<tr>
<td>P4</td>
<td>P4</td>
</tr>
<tr>
<td>0.46</td>
<td>0.38</td>
</tr>
</tbody>
</table>

Time (sec) | Time (sec)
---|---
0 | 0
500 | 1000
1500 | 2000
2500 | 3000
Optimal K Comparison

\[ \sum_{(i,j) \in \mathcal{E}} k_{ij} x_i^d \leq c_{tot} \]

Edge limits

\[ k_{ij} x_i^d \leq c_{ij}, \quad (i, j) \in \mathcal{E} \]
• Tradeoff between convergence rate, equilibrium traffic

Equilibrium traveler fraction averaged over 6000 data points
• Tradeoff between convergence rate, equilibrium traffic
• Faster convergence with increased site connectivity
• Limits on edge traffic eliminate advantage of knowing $x^0$
- Monte Carlo runs are consistently optimal but computationally slow.