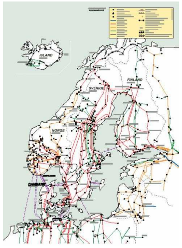
Distributed Procedures for Control Synthesis

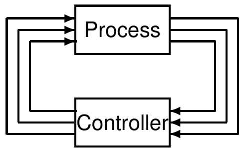
Anders Rantzer

with contributions by Pontus Giselsson, Karl Mårtensson

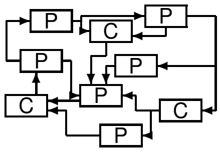
Lund University, Sweden

Building theoretical foundations for distributed control





⇓



We need methodology for

- Decentralized specifications
- Decentralized design
- Verification of global behavior

Outline

- **Introduction**
- Dynamic dual decomposition
- Distributed Model Predictive Control
- Distributed Iterative Feedback Tuning

50 year old idea: Dual decomposition

$$\min_{z_i}[V_1(z_1,z_2)+V_2(z_2)+V_3(z_3,z_2)]$$

$$=\max_{p_i}\min_{z_i,v_i}\Big[V_1(z_1,v_1)+V_2(z_2)+V_3(z_3,v_3)+p_1(z_2-v_1)+p_3(z_2-v_3)\Big]$$

The optimum is a Nash equilibrium of the following game:

The three computers try to minimize their respecive costs

Computer 1: $\min_{z_1,v_1}\big[V_1(z_1,v_1)-p_1v_1\big]$

Computer 2: $\min_{z_2}\big[V_2(z_2)+(p_1+p_3)z_2\big]$

Computer 3: $\min_{z_3,v_3}\big[V_3(z_3,v_3)-p_3v_3\big]$

while the ”market makers” try to maximize their payoffs

Between computer 1 and 2: $\max_{p_1}[p_1(z_2-v_1)]$

Between computer 2 and 3: $\max_{p_3}[p_3(z_2-v_3)]$

The saddle point algorithm

Update in gradient direction:

Computer 1: $\begin{cases}\dot z_1 = -\partial V_1/\partial z_1\\ \dot v_1 = -\partial V_1/\partial z_2 + p_1\end{cases}$

Computer 1 and 2: $\dot p_1 = z_2 - v_1$

Computer 2: $\dot z_2 = -\partial V_2/\partial z_2 - p_1 - p_3$

Computer 2 and 3: $\dot p_3 = z_2 - v_3$

Computer 3: $\begin{cases}\dot z_3 = -\partial V_3/\partial z_3\\ \dot v_3 = -\partial V_3/\partial z_2 + p_3\end{cases}$

Globally convergent if V_i are convex!
[Arrow, Hurwicz, Usawa 1958]

Decentralized Bounds on Suboptimality

Given any $p_1, p_3, \bar z_1, \bar z_2, \bar z_3$, the distributed test

$$V_1(\bar z_1,\bar z_2)-p_1\bar z_2 \le \alpha\min_{z_1,v_1}[V_1(z_1,v_1)-p_1v_1]$$

$$V_2(\bar z_2)+(p_1+p_3)\bar z_2 \le \alpha\min_{z_2}[V_2(z_2)+(p_1+p_3)z_2]$$

$$V_3(\bar z_3,\bar z_2)-p_3\bar z_2 \le \alpha\min_{z_3,v_3}[V_3(z_3,v_3)-p_3v_3]$$

implies that the globally optimal cost J^* is bounded as

$$J^* \le V_1(\bar z_1,\bar z_2)+V_2(\bar z_2)+V_3(\bar z_3,\bar z_2) \le \alpha J^*$$

Proof: Add both sides up!

Important Aspects of Dual Decomposition

- Very weak assumptions on graph
- No need for central coordination
- Natural learning procedure is globally convergent
- Unique Nash equilibrium corresponds to global optimum

Conclusion: Ideal for control synthesis by prescriptive games

A long history

The saddle algorithm:
Arrow, Hurwicz, Usawa 1958

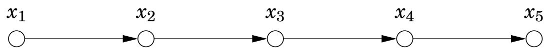
Books on control and coordination in hierarchical systems:
Mesarovic, Macko, Takahara 1970
Singh, Titli 1978
Findeisen 1980

Major application to water supply network:
Carpentier and Cohen, Automatica 1993

Outline

- Introduction
- **Dynamic dual decomposition**
- Distributed Model Predictive Control
- Distributed Iterative Feedback Tuning

Example 1: A vehicle formation

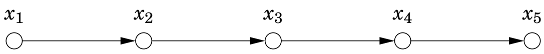


Each vehicle obeys the independent dynamics

$$\begin{bmatrix} x_1(t+1) \\ x_2(t+1) \\ x_3(t+1) \\ x_4(t+1) \end{bmatrix} = \begin{bmatrix} * & 0 & 0 & 0 \\ 0 & * & 0 & 0 \\ 0 & 0 & * & 0 \\ 0 & 0 & 0 & * \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \\ x_4(t) \end{bmatrix} + \begin{bmatrix} B_1u_1(t)+w_1(t) \\ B_2u_2(t)+w_2(t) \\ B_3u_3(t)+w_3(t) \\ B_4u_4(t)+w_4(t) \end{bmatrix}$$

The objective is to make $\mathbf{E}|Cx_{i+1} - Cx_i|^2$ small for $i = 1, \dots, 4$.

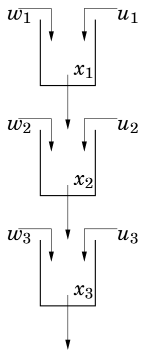
Example 2: A supply chain for fresh products



Fresh products degrade with time:

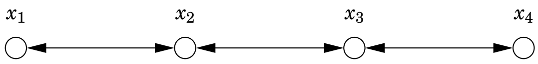
$$\begin{bmatrix} x_1(t+1) \\ x_2(t+1) \\ x_3(t+1) \\ x_4(t+1) \end{bmatrix} = \begin{bmatrix} * & 0 & 0 & 0 \\ 0 & * & 0 & 0 \\ 0 & 0 & * & 0 \\ 0 & 0 & 0 & * \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \\ x_4(t) \end{bmatrix} + \begin{bmatrix} -u_1(t)+w_1(t) \\ u_1(t)-u_2(t) \\ u_2(t)-u_3(t) \\ u_3(t)+w_4(t) \end{bmatrix}$$

Example 3: Water distribution systems



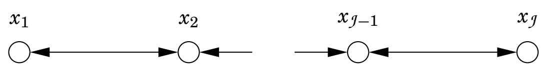
$$\begin{bmatrix} x_1(t+1) \\ x_2(t+1) \\ x_3(t+1) \\ x_4(t+1) \end{bmatrix} = \begin{bmatrix} * & 0 & 0 & 0 \\ * & * & 0 & 0 \\ 0 & * & * & 0 \\ 0 & 0 & * & * \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \\ x_4(t) \end{bmatrix} + \begin{bmatrix} B_1u_1+w_1 \\ B_2u_2+w_2 \\ B_3u_3+w_3 \\ B_4u_4+w_4 \end{bmatrix}$$

Example 4: Wind farms



$$\begin{bmatrix} x_1(t+1) \\ x_2(t+1) \\ x_3(t+1) \\ x_4(t+1) \end{bmatrix} = \begin{bmatrix} * & * & 0 & 0 \\ * & * & * & 0 \\ 0 & * & * & * \\ 0 & 0 & * & * \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \\ x_4(t) \end{bmatrix} + \begin{bmatrix} B_1u_1(t)+w_1(t) \\ B_2u_2(t)+w_2(t) \\ B_3u_3(t)+w_3(t) \\ B_4u_4(t)+w_4(t) \end{bmatrix}$$

A control problem with graph structure



$$\begin{bmatrix} x_1(\tau+1) \\ x_2(\tau+1) \\ \vdots \\ x_{\mathcal{J}}(\tau+1) \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} & & 0 \\ A_{21} & \ddots & \ddots & \\ & \ddots & \ddots & A_{(\mathcal{J}-1)\mathcal{J}} \\ 0 & & A_{\mathcal{J}(\mathcal{J}-1)} & A_{\mathcal{J}\mathcal{J}} \end{bmatrix} \begin{bmatrix} x_1(\tau) \\ x_2(\tau) \\ \vdots \\ x_{\mathcal{J}}(\tau) \end{bmatrix} + \begin{bmatrix} u_1(\tau) \\ u_2(\tau) \\ \vdots \\ u_{\mathcal{J}}(\tau) \end{bmatrix}$$

Minimize the convex objective $\sum_{t=0}^N \underbrace{\sum_{i=1}^{\mathcal{J}} \ell_i(x_i(\tau), u_i(\tau))}_{\ell(x(\tau),u(\tau))}$

with convex constraints $x_i(\tau) \in X_i$, $u_i(\tau) \in U_i$ and $x(0) = \bar{x}$.

Decomposing the problem

Minimize $\sum_{t=0}^N \ell(x(\tau), u(\tau))$

subject to

$$\begin{bmatrix} x_1(\tau+1) \\ x_2(\tau+1) \\ \vdots \\ x_{\mathcal{J}}(\tau+1) \end{bmatrix} = \begin{bmatrix} A_{11}x_1(\tau) \\ A_{22}x_2(\tau) \\ \vdots \\ A_{\mathcal{J}\mathcal{J}}x_{\mathcal{J}}(\tau) \end{bmatrix} + \begin{bmatrix} v_1(\tau) \\ v_2(\tau) \\ \vdots \\ v_{\mathcal{J}}(\tau) \end{bmatrix} + \begin{bmatrix} u_1(\tau) \\ u_2(\tau) \\ \vdots \\ u_{\mathcal{J}}(\tau) \end{bmatrix}$$

where $x(0) = \bar{x}$ and

$$v_i = \sum_{j \neq i} A_{ij} x_j$$

holds for all i.

Decomposing the Cost Function

$$\max_p \min_{u,v,x} \sum_{\tau=0}^N \sum_{i=1}^{\mathcal{J}} \left[\ell_i(x_i, u_i) + p_i^T \left(v_i - \sum_{j \neq i} A_{ij} x_j \right) \right]$$

$$= \max_p \sum_i \min_{u_i, x_i} \sum_{\tau=0}^N \underbrace{\left[\ell_i(x_i, u_i) + p_i^T v_i - x_i^T \left(\sum_{j \neq i} A_{ji}^T p_j \right) \right]}_{\ell_i^p(x_i, u_i, v_i)}$$

so, given the sequences $\{p_j(t)\}_{t=0}^N$, agent i should minimize

$$\underbrace{\sum_{\tau=0}^N \ell_i(x_i, u_i) +}_{\text{local cost}} \overbrace{\sum_{\tau=0}^N p_i^T v_i}^{\text{what he expects others to charge him}} - \underbrace{\sum_{\tau=0}^N x_i^T \left(\sum_{j \neq i} A_{ji}^T p_j \right)}_{\text{what he is payed by others}}$$

subject to $x_i(t+1) = A_{ii}x_i(t) + v_i(t) + u_i(t)$ and $x_i(0) = \bar{x}_i$.

Distributed Optimization Procedure

Local optimizations in each node

$$V_i^{N,p}(\bar{x}_i) = \min_{u_i, x_i} \sum_{\tau=0}^{N} \ell_i^p(x_i(\tau), u_i(\tau), v_i(\tau))$$

can be coordinated by (local) gradient updates of the prices

$$p_i^{k+1}(\tau) = p_i^k(\tau) + \gamma_i^k \left[v_i^k(\tau) - \sum_{j \neq i} A_{ij} x_j^k(\tau) \right]$$

Future prices included in negotiation for first control input!

Convergence guaranteed under different types of assumptions on the step size sequence γ_i^k.

Outline

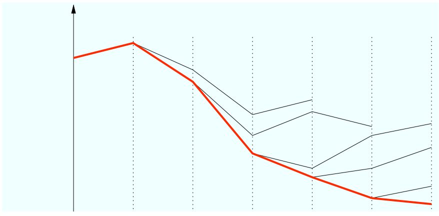
- Introduction
- Dynamic dual decomposition
- **Distributed Model Predictive Control**
- Distributed Iterative Feedback Tuning

Idea of Distributed Model Predicitve Control

Replace the original problem by iterative online solutions of the decentralized finite horizon problem

$$\min_{x_i, u_i} \sum_{t=0}^{N} l_i^p(x_i(t), u_i(t), v_i(t))$$

Two sources of error: Finite horizon and non-optimal prices



A Distributed MPC Algorithm

At time t:

1. Measure the states $x_i(t)$ locally.
2. Use gradient iterations to generate
   - price prediction sequences $\{p_i(t,\tau)\}_{\tau=0}^N$
   - state prediction sequences $\{x_i(t,\tau)\}_{\tau=1}^N$
   - input prediction sequences $\{u_i(t,\tau)\}_{\tau=1}^N$

   warm-starting from predictions at time $t-1$.
3. Apply the inputs $u_i(t) = u_i(t,0)$.

Important parameters: Prediction horizons N_i, number of gradient iterations K_i and gradient step sizes γ_i.

Fixed or flexible parameters N_i, K_i, γ_i?

Fixed parameters

- Simpler implementation
- Gives distributed LTI controllers
- Can be analyzed off-line or on-line

Flexible parameters

- Useful to handle hard state constraints
- Can speed up on-line computations
- Can slow down on-line computations

"Wind Farm" Revisited

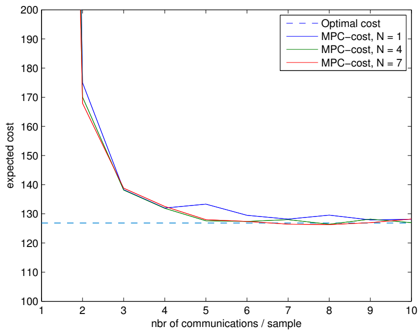
Minimize $V = \mathbf{E} \sum_{i=1}^n \left(|x_i|^2 + |u_i|^2\right)$ subject to

$$\begin{bmatrix} x_1(t+1) \\ x_2(t+1) \\ \vdots \\ x_n(t+1) \end{bmatrix} = \begin{bmatrix} 0.6 & 0.1 & & 0 \\ 0.3 & \ddots & \ddots & \\ & \ddots & \ddots & 0.1 \\ 0 & & 0.3 & 0.6 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \\ \vdots \\ x_n(t) \end{bmatrix} + \begin{bmatrix} u_1(t) + w_1(t) \\ u_2(t) + w_2(t) \\ \vdots \\ u_n(t) + w_n(t) \end{bmatrix}$$

We will solve this by "distributed MPC". For every t, the agents measure their local state $x_i(t)$. The vector of future prices is then updated by a few gradient iterations starting from the prices computed at $t-1$ for a time horizon of length N.

Re-negotiation of future prices at every time step!
This is the key to dynamic dual decomposition.

Performance Versus Number of Gradient Iterations



A distributed controller with 100 agents, using only local data.
Low order local dynamics, so short prediction horizons are OK

Challenges for theory

- What prediction horizon is needed?
- How many gradient iterations for the prices?

References:
Grüne and Rantzer, IEEE TAC October 2008.
Pannek, PhD thesis 2009
Giselsson and Rantzer, submission for ACC 2010.

Theorem on accuracy of distributed MPC

Suppose all local finite horizon costs
$V_i^{N,p}(\bar{x}_i) = \min_{u_i,x_i} \sum_{\tau=0}^{N} \ell_i^p(x_i(\tau), u_i(\tau), v_i(\tau)) \geq 0$
satisfy

$V_i^{N,p(t,\cdot)}(x_i(t)) \geq V_i^{N,p(t+1,\cdot)}(x_i(t+1)) + \alpha \ell_i^{p(t,\cdot)}(x_i(t), u_i(t), \sum_{j\neq i} A_{ij} x_j(t))$

for all i and $t \geq 0$. Then

$$\alpha \sum_{t=0}^{\infty} \ell(x(t), u(t)) \leq V^{\infty}(\bar{x})$$

Notice: Failure of inequality hints on update of N_i or K_i!

Conclusions on Distributed MPC

We have synthesized a game that solves optimal control problems via independent decision-makers in every node, acting in their own interest!

- Optimal strategies independent of global graph structure!
- States are measured only locally
- Linearly complexity (given horizon and iteration scheme)
- Distributed bounds on distance to optimality

Outline

- Introduction
- Dynamic dual decomposition
- Distributed Model Predictive Control
- **Distributed Iterative Feedback Tuning**

Tuning a tri-diagonal controller for the "Wind Farm"

Minimize $V = \mathbf{E} \sum_{i=1}^{n} (|x_i|^2 + |u_i|^2)$

$$\begin{bmatrix} x_1(t+1) \\ x_2(t+1) \\ \vdots \\ x_n(t+1) \end{bmatrix} = \begin{bmatrix} 0.6 & 0.1 & & 0 \\ 0.3 & \ddots & \ddots & \\ & \ddots & \ddots & 0.1 \\ 0 & & 0.3 & 0.6 \end{bmatrix} \begin{bmatrix} x_1(t+1) \\ x_2(t+1) \\ \vdots \\ x_n(t+1) \end{bmatrix} + \begin{bmatrix} u_1(t) + w_1(t) \\ u_2(t) + w_2(t) \\ \vdots \\ u_n(t) + w_n(t) \end{bmatrix}$$

We will optimize a tri-diagonal control structure

$$\bar{L} = \begin{bmatrix} * & * & & 0 \\ * & & \ddots & \\ & \ddots & & * \\ 0 & & * & * \end{bmatrix}$$

Computing the closed loop control performance

We are applying the control law $u = -Lx$ to the system

$$x(t+1) = Ax(t) + Bu(t) + w(t)$$

where w is white noise with variance W. Define

$$J(L) = \mathbf{E}\left(|x|_Q^2 + |u|_R^2\right)$$

Then the gradient with respect to a particular element L_{ij} is

$$(\nabla_L J)_{ij} = 2RL\mathbf{E}\,[x_i x_j^T] + 2B^T \mathbf{E}\,[p_i x_j^T]$$

where $p(t)$ is the stationary solution of the adjoint equation

$$p(t-1) = (A - BL)^T p(t) - (Q + L^T RL)x(t)$$

A distributed synthesis procedure

1. Measure the states $x_i(t)$ for $t = t_k, \ldots, t_k + N$
2. Simulate the adjoint equation
$$p_i(t-1) = \sum_{j\in E_i} (A - BL)_{ji}^T p_j(t) - 2(Q_i x_i(t) - \sum_{j\in E_i} L_{ji}^T R_j u_j(t))$$
for $t = t_k, \ldots, t_k + N$ by communicating states between nodes.
3. Calculate the estimates of $\mathbf{E}\, u_i x_j^T$ and $\mathbf{E}\, p_i x_j^T$ by
$$(\mathbf{E}\, u_i x_j^T)_{\text{est}} = \tfrac{1}{N+1} \sum_{t=t_k}^{t_k+N} u_i(t) x_j(t)^T \quad (\mathbf{E}\, p_i x_j^T)_{\text{est}} = \tfrac{1}{N+1} \sum_{t=t_k}^{t_k+N} p_i(t) x_j(t)^T$$
4. For fixed step length $\gamma > 0$, update
$L_{ij}^{(k+1)} = L_{ij}^{(k)} + 2\gamma R_i \left(\mathbf{E}\, u_i x_j^T\right)_{\text{est}} + B_i^T \left(\mathbf{E}\, p_i x_j^T\right)_{\text{est}}$.
Let $t_{k+1} = t_k + N$ and start over.

Gradient iteration for the wind park

```
cost =

    14.9887


L =

     0     0     0     0     0
     0     0     0     0     0
     0     0     0     0     0
     0     0     0     0     0
     0     0     0     0     0
```

Gradient iteration for the wind park

```
cost =

    10.5429


L =

    0.0327    0.0400         0         0         0
   -0.0007    0.0560    0.0527         0         0
         0   -0.0069    0.0434    0.0315         0
         0         0   -0.0207    0.0131    0.0437
         0         0         0   -0.0033    0.0373
```

Gradient iteration for the wind park

cost =

7.8184

L =

0.0310	0.0595	0	0	0
-0.0168	0.1002	0.1151	0	0
0	0.0345	0.1357	0.0986	0
0	0	0.0636	0.0831	0.1351
0	0	0	0.0102	0.1295

Gradient iteration for the wind park

cost =

7.6192

L =

0.0404	0.0685	0	0	0
-0.0086	0.1076	0.1193	0	0
0	0.0382	0.1421	0.1094	0
0	0	0.0593	0.0991	0.1449
0	0	0	0.0131	0.1348

Gradient iteration for the wind park

cost =

7.4004

L =

0.0576	0.0583	0	0	0
0.0115	0.1224	0.1381	0	0
0	0.0373	0.1500	0.1153	0
0	0	0.0546	0.1068	0.1566
0	0	0	0.0168	0.1594

Gradient iteration for the wind park

cost =

7.2493

L =

0.0712	0.0654	0	0	0
0.0061	0.1224	0.1443	0	0
0	0.0341	0.1550	0.1166	0
0	0	0.0773	0.1409	0.1580
0	0	0	0.0418	0.1601

Gradient iteration for the wind park

cost =

6.9736

L =

0.0936	0.1056	0	0	0
0.0331	0.1775	0.1341	0	0
0	0.0563	0.1500	0.1215	0
0	0	0.0700	0.1564	0.1567
0	0	0	0.0567	0.1646

Gradient iteration for the wind park

cost =

6.8211

L =

0.1390	0.1070	0	0	0
0.0357	0.1821	0.1549	0	0
0	0.0668	0.1797	0.1098	0
0	0	0.0633	0.1685	0.1413
0	0	0	0.0589	0.1754

Gradient iteration for the wind park

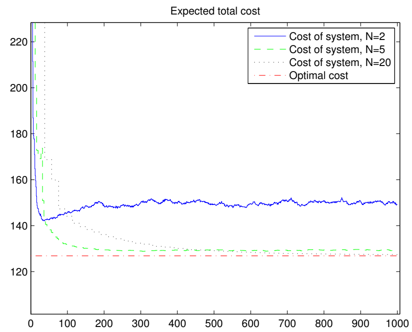
cost =

6.7464

L =

0.1438	0.1208	0	0	0
0.0470	0.2031	0.1632	0	0
0	0.0749	0.1909	0.1046	0
0	0	0.0779	0.1843	0.1388
0	0	0	0.0445	0.1732

Performance Versus Number of Gradient Iterations



A distributed controller with 100 agents, using only local data. Fewer gradient iterations gives faster convergence, but worse stationary performance.

Summary

- Introduction
- Dynamic dual decomposition
- Distributed Model Predictive Control
- Distributed Iterative Feedback Tuning

Theme Semester Spring 2010

Invited world-leading researchers from Control, Computer Science, Economics, Communication, Mathematics, . . .

- Multi-agent coordination and estimation (Jan 18 - Feb 19, 2010)
- Distributed decisions via games and price mechanisms (Feb 22 - Mar 26, 2010)
- Adaptation and learning in autonomous systems (Apr 6 - 30, 2010)
- Distributed model predictive control and supply chains (May 3 - 28, 2010)

See www.lccc.lth.se and announcements.

Case study: A water supply network in Paris

[Carpentier and Cohen, Automatica 1993]

- Network serving about 1 million inhabitants
- 20 main water reservoirs
- 30 pumping stations
- 13 peripheral subnetworks

Challenges for control

- Minimize cost for pumping
- Bounds on reservoirs
- Bounds and delays in pumping power
- Prediction of consumption

Optimal control using dual decomposition and saddle algorithm
Subnetworks separated by two variables: Water flow and price