Performance, Information Pattern Trade-offs and Computational Complexity Analysis of a Consensus Based Distributed Optimization Method

Alireza Farhadi

in collaboration with M. Cantoni and P. M. Dower

Department of Electrical and Electronic Engineering

The University of Melbourne

October 30, 2012

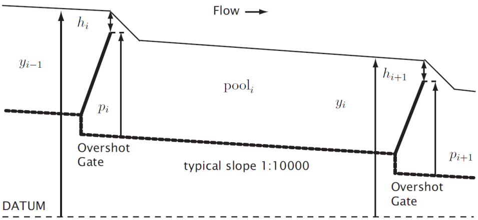


Figure: An irrigation network.

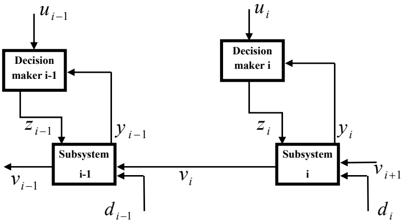


Figure: An automated irrigation network via distributed distant downstream feedback control. $z_i(s) = C_i(s)e_i(s)$, $C_i(s) = \frac{K_i T_i s + K_i}{s(T_i F_i s + T_i)}$, $e_i = u_i - y_i$.

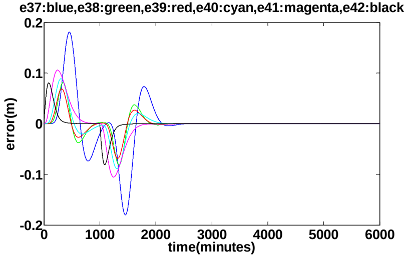


Figure: Downstream errors.

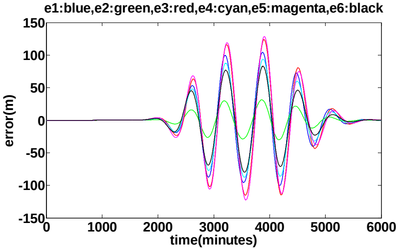


Figure: Upstream errors.

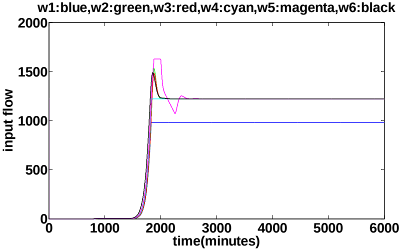


Figure: Upstream input flows.

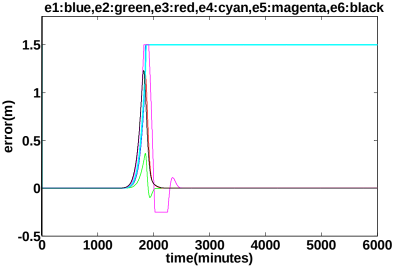


Figure: Upstream errors.

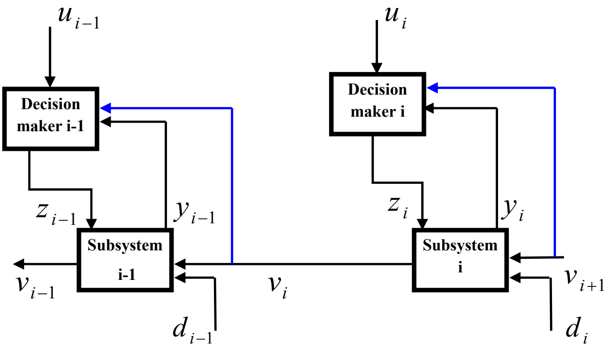


Figure: An automated irrigation network via distributed distant downstream feedback and feedforward control. $z_i(s) = C_i(s)e_i(s) + f_i v_{i+1}$, $C_i(s) = \frac{K_i T_i s + K_i}{s(T_i F_i s + T_i)}$, $e_i = u_i - y_i$.

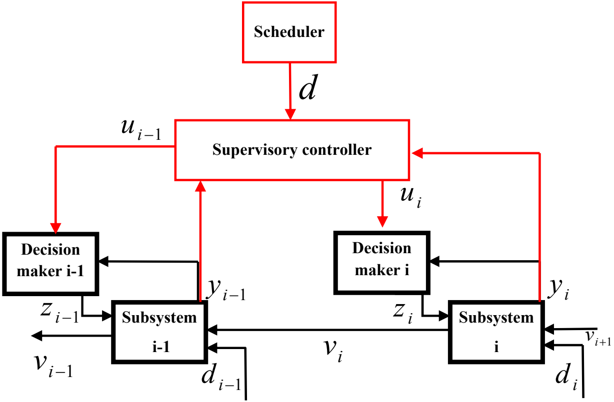


Figure: An automated irrigation network equipped with a supervisory controller.

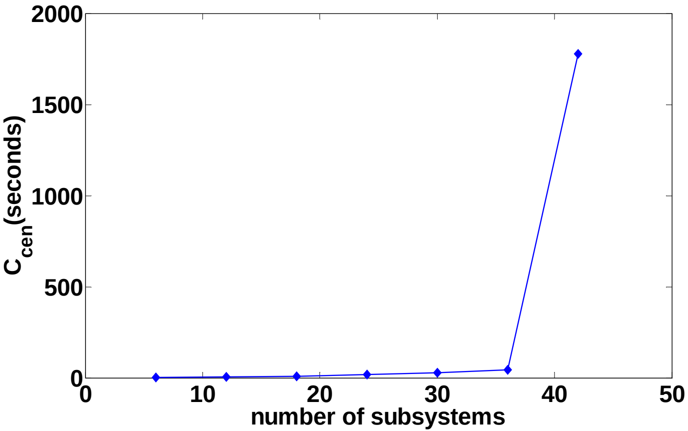


Figure: Computational complexity of the centralized optimization method versus the number of subsystems.

Distributed supervisory control

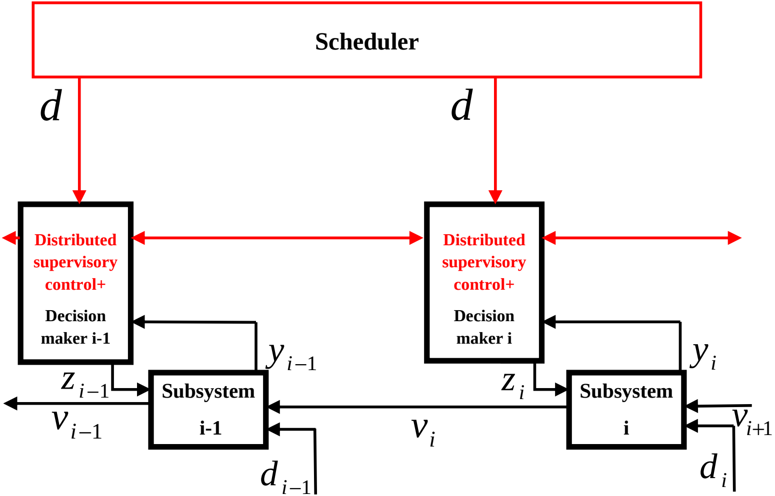


Figure: An automated irrigation network equipped with distributed supervisory controller.

Distributed optimization method (problem formulation)

$$\min_{\mathbf{u}=(\mathbf{u_1},\ldots,\mathbf{u_n})} \{J(u_1,...,u_n), \quad u_i \subset \mathcal{U}_i\}$$

$$\mathcal{U}_i \subset \mathbf{R}^{m_i}, \quad argmin_{u_i} J(u_1,...,u_n) \in \mathbf{R}^{Nm_i}.$$

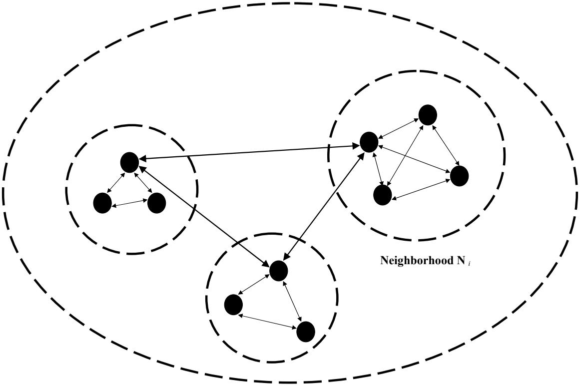


Figure: Two-level architecture for exchanging information between distributed decision makers.

Distributed optimization method (steps[1])

$N_1 = \{S_1, S_2\}, \quad N_2 = \{S_3, S_4\}$

- *Initialization:* The information exchange between neighborhoods at outer iterate t makes it possible for subsystem S_i to initialize its local decision variables as $h_i^0 = u_i^t$, where $u_i^0 \in \mathcal{U}_i$ are chosen arbitrarily at time $t = 0$.

- *Inner Iterate:* Then, subsystem S_i performs $\bar{p}$ inner iterates as follows: For inner iterate $p \in \{0, 1, ..., \bar{p} - 1\}$, it first updates its decision variable via

$$h_i^{p+1} = \pi_i h_i^* + (1 - \pi_i) h_i^p,$$

where

$$\pi_1 + \pi_2 = 1, \quad \pi_3 + \pi_4 = 1$$

and

$$h_1^* = argmin_{h_1 \in \mathcal{U}_1} J(h_1, h_2^p, h_3^0, h_4^0), \quad h_2^* = argmin_{h_2 \in \mathcal{U}_2} J(h_1^p, h_2, h_3^0, h_4^0),$$

$$h_3^* = argmin_{h_3 \in \mathcal{U}_3} J(h_1^0, h_2^0, h_3, h_4^p), \quad h_4^* = argmin_{h_4 \in \mathcal{U}_4} J(h_1^0, h_2^0, h_3^p, h_4).$$

[1] [ACC2010] B. T. Stewart, J. B. Rawlings, and S. J. Wright.

Distributed optimization method (steps)

- *Inner Iterate (continued):* Then, subsystem S_i trades its updated decision variable h_i^{p+1} with all other subsystems within its neighborhood.
- *Outer Iterate:* After $\bar{p}$ inner iterates there is an outer iterate update as follows

$$u_i^{t+1} = \lambda_i h_i^{\bar{p}} + (1 - \lambda_i) u_i^t,$$

where

$$\lambda_1 = \lambda_2, \quad \lambda_3 = \lambda_4, \quad \lambda_1 + \lambda_3 = 1.$$

Then, there is an outer iterate communication, in which the updated decision variables u_i^{t+1} are shared between all neighborhoods and subsequently between all subsystems.

Feasibility, convergence and optimality results [2]

Feasibility: Given any collection of disjoint neighborhoods, above strictly convex finite horizon cost functional J, convex control constraint sets $\mathcal{U}_i$ and a feasible initialization (i.e., $u_i^0 \in \mathcal{U}_i$), the inner and outer iterates are feasible (i.e., $h_i^{p+1}, u_i^{t+1} \in \mathcal{U}_i$).

Convergence: Given any collection of disjoint neighborhoods and a feasible initialization, the strictly convex finite horizon cost functional $J(u_1^t, \ldots, u_n^t)$ is non-increasing at each outer iterate t and converges as $t \longrightarrow \infty$.

Optimality: Given any collection of disjoint neighborhoods, a feasible initialization, strictly convex and quadratic cost J, and closed convex control constraint sets $\mathcal{U}_i$, the cost $J(u_1^t, \ldots, u_n^t)$ converges to the optimal cost $J(u_1^*, \ldots, u_n^*)$, and the iterates $(u_1^t, \ldots, u_n^t)$ converge to the unique optimal solution $(u_1^*, \ldots, u_n^*)$, as $t \rightarrow \infty$.

[2] [AUCC2012]A. Farhadi, M. Cantoni, and P. M. Dower.

Interaction strength decomposition method

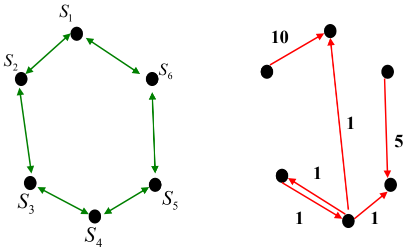


Figure: Left: Communication graph. Right: Interaction strength graph summarizing the effects of decision variables on subsystems.

No hopping is allowed for intra-neighborhood communication $\Rightarrow$ Following the communication graph, the size of each neighborhood must be at most 2:
Option1: $\{S_2, S_3\}$, $\{S_4, S_5\}$, $\{S_6, S_1\}$
Option2: $\{S_1, S_2\}$, $\{S_3, S_4\}$, $\{S_5, S_6\}$

Following interaction strength graph, option 2 is selected.

Interaction strength decomposition method

Dynamic system:

$$S_i: \; x_i[k+1] = A_i x_i[k] + B_i u_i[k] + v_i[k], i = 1, 2, ..., n, k \in \{0, 1, 2, ..., N-1\},$$

where

$$v_i[k] = \sum_{j=1, j\neq i}^{n} M_{ij} x_j[k] + N_{ij} u_j[k].$$

Transfer function from $U(z) = \begin{pmatrix} U_1'(z) & . & . & . & U_n'(z) \end{pmatrix}'$ to state $X(z) = \begin{pmatrix} X_1'(z) & . & . & . & X_n'(z) \end{pmatrix}'$ is given by

$$G(z) = V^{-1}(z) W(z),$$

where $V(z) \doteq [V_{ij}(z)]$ with

$$V_{ij}(z) \doteq \begin{cases} I_{n_i}, & \text{when } i = j \\ -(zI_{n_i} - A_i)^{-1} M_{ij}, & \text{otherwise} \end{cases}$$

and $W(z) \doteq [W_{ij}(z)]$ with

$$W_{ij}(z) \doteq \begin{cases} (zI_{n_i} - A_i)^{-1} B_i, & \text{when } i = j \\ (zI_{n_i} - A_i)^{-1} N_{ij}, & \text{otherwise.} \end{cases}$$

Interaction strength decomposition method

$$G(z)|_{z=1} = \begin{pmatrix} E_1 & E_{12} & . & . & . & E_{1n} \\ E_{21} & E_2 & . & . & . & E_{2n} \\ . & & & & & \\ . & & & & & \\ . & & & & & \\ E_{n1} & E_{n2} & . & . & . & E_n \end{pmatrix}, E_{ij} \in \mathbf{R}^{n_i \times m_j}.$$

Interaction Strength (IS):

$$IS_{ij} \doteq \begin{cases} 0, & \text{if } i = j \\ \frac{\sigma_{max}(E_{ij})}{\sigma_{min}(E_i)}, & \text{if } \sigma_{min}(E_i) \neq 0 \text{ and } i \neq j \\ \frac{\sigma_{max}(E_{ij})}{\gamma}, & \text{if } \sigma_{min}(E_i) = 0 \text{ and } i \neq j \end{cases}$$

Normalized interaction strength:

$$ISN_{ij} \doteq \text{round}\left(\frac{IS_{ij}}{IS_{min}}\right), \; IS_{min} \doteq \min_{\{i,j;IS_{ij}>0\}} IS_{ij}.$$

Interaction strength decomposition method

Example: Consider a system with six interacting scalar subsystems. The aggregated system is described as follows:

$$x[k+1] = Ax[k] + Bu[k],$$

$$x[k] = \begin{pmatrix} x_1[k] & x_2[k] & x_3[k] & x_4[k] & x_5[k] & x_6[k] \end{pmatrix}'$$

$$u[k] = \begin{pmatrix} u_1[k] & u_2[k] & u_3[k] & u_4[k] & u_5[k] & u_6[k] \end{pmatrix}',$$

$$A = \begin{pmatrix} 1.7049 & -0.0049 & -0.9082 & -0.2732 & 0.5496 & -0.2756 \\ 0.2328 & 1.4672 & -0.0213 & -0.4127 & -0.4861 & 0.5709 \\ 0.1213 & -0.1213 & 0.7311 & 0.0955 & 0.5566 & -0.4652 \\ -0.3836 & 0.3836 & 0.1393 & 1.2061 & 0.132 & 0.198 \\ -0.1148 & 0.11.48 & -0.6754 & 0.007 & 2.3762 & -0.4357 \\ -0.5148 & 0.5148 & 0.0246 & -0.143 & 0.4762 & 1.5143 \end{pmatrix},$$

$$B = diag(1.7, -1, 1.5, -1.2, 1.9, 0.86).$$

Interaction strength decomposition method

Interaction strength matrix:

Subsystems	S_1	S_2	S_3	S_4	S_5	S_6
S_1	0	36	226	3	245	82
S_2	37	0	21	29	49	27
S_3	20	12	0	22	182	70
S_4	93	55	63	0	148	39
S_5	53	31	151	13	0	67
S_6	106	62	73	1	185	0

Strength weights $(SW(ij) \doteq ISN_{ij} + ISN_{ji}, \quad i \neq j)$

$(1,2) = 73$	$(1,3) = 246$	$(1,4) = 96$	$(1,5) = 298$
$(1,6) = 188$	$(2,3) = 33$	$(2,4) = 84$	$(2,5) = 80$
$(2,6) = 89$	$(3,4) = 85$	$(3,5) = 333$	$(3,6) = 143$
$(4,5) = 161$	$(4,6) = 40$	$(5,6) = 252$	$(5,6) = 252$

$$N_1 = \{S_3, S_5\}, \qquad N_2 = \{S_1, S_6\}, \qquad N_3 = \{S_2, S_4\}.$$

Interaction strength decomposition method

Strength weights $(SW(ijk) \doteq ISN_{ij} + ISN_{ik} + ISN_{ji} + ISN_{jk} + ISN_{ki} + ISN_{kj},\ \ i \neq j \neq k)$

$(1,2,3) = 352$	$(1,2,4) = 253$	$(1,2,5) = 451$
$(1,2,6) = 350$	$(1,3,4) = 427$	$(1,3,5) = 877$
$(1,3,6) = 577$	$(1,4,5) = 555$	$(1,4,6) = 324$
$(1,5,6) = 738$	$(2,3,4) = 202$	$(2,3,5) = 446$
$(2,3,6) = 265$	$(2,4,5) = 325$	$(2,4,6) = 213$
$(2,5,6) = 421$	$(3,4,5) = 579$	$(3,4,6) = 268$
$(3,5,6) = 728$	$(4,5,6) = 453$	$(4,5,6) = 453$

$$N_1 = \{S_1, S_3, S_5\}, \qquad N_2 = \{S_2, S_4, S_6\}.$$

Performance criteria

Performance Loss: For a given number of outer iterate updates t and $\bar{p}$, the Performance Loss $PL_t(\bar{p})$ (measured in percent) is defined as

$$PL_t(\bar{p}) \doteq 100\left(\frac{J(u_1^t, \dots, u_n^t) - \bar{J}}{\bar{J}}\right),$$

where $\bar{J}$ is the optimal cost.

Total Number of Iterations: For a given $\bar{p}$,

$$T_t \doteq \bar{p} \times t$$

is referred as the total number of iterations up to outer iterate t.

Total Number of Iterations for Convergence: For a given performance loss PL, let $\bar{t}_{PL}$ be the smallest integer such that

$$PL_t(\bar{p}) \leq PL \;\text{ for all } t \geq \bar{t}_{PL}.$$

Then,

$$T_{PL} \doteq \bar{p} \times \; \bar{t}_{PL}$$

is referred as the total number of iterations for convergence.

Illustrative example

Dynamic system:

$$S_i: \ x_i[k+1] = A_i x_i[k] + B_i u_i[k] + v_i[k], i = 1, 2, ..., 6, k \in \{0, 1, 2, 3, 4\},$$

where

$$x_i[0] = 0, \quad v_i[k] = \sum_{j=1, j\neq i}^{6} M_{ij} x_j[k].$$

$$\min_{\mathbf{u}} \left\{ J(\mathbf{x}[0], u_1, ..., u_6), x_i[k] \in \mathcal{X}_i = [-12, 12], u_i[k] \in \mathcal{G}_i = [-6, 6], \forall i, k \right\},$$

$$J(\mathbf{x}[0], u_1, ..., u_6) \doteq \sum_{i=1}^{6} \sum_{k=0}^{4} ||x_i[k] - x_i^d||^2 + ||u_i[k]||^2.$$

$$x_1^d = 1, x_2^d = 2, x_3^d = 3, x_4^d = 4, x_5^d = 5, x_6^d = 6,$$

$$\bar{J} = 9370.89.$$

$\bar{p}$	T_{PL}	$PL_t(\bar{p})$ at $t = T_{PL}/\bar{p}$	Computation time (sec.)
1	453	0.99	77.63
10	820	0.95	142.34
20	1400	0.93	244.93
50	3250	0.98	564.91

Table: Two-neighborhoods case.

$\bar{p}$	T_{PL}	$PL_t(\bar{p})$ at $t = T_{PL}/\bar{p}$	Computation time (sec.)
1	424	0.99	74.23
10	2200	0.99	390.14
20	4320	0.98	755.36
50	10750	0.99	1885.2

Table: Three-neighborhoods case.

$\bar{p}$	T_{PL}	$PL_t(\bar{p})$ at $t = T_{PL}/\bar{p}$	Computation time (sec.)
1	1020	0.99	179.21
10	10200	0.99	1834.3
20	20400	0.99	3569.9
50	51000	0.99	9027.9

Table: Six-neighborhoods case.

Illustrative example

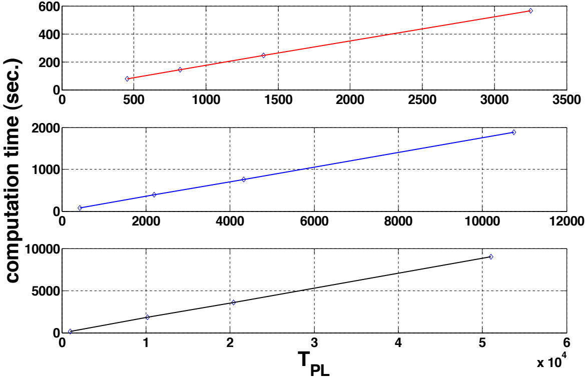


Figure: Computation time versus the total number of iterations for convergence T_{PL} for different decompositions and $PL = 1$ percent. Red: The two-neighborhoods case. Blue: The three-neighborhoods case. **Black**: The six-neighborhoods case.

Computation time equals γT_{PL}, where $\gamma = 0.175$.

Illustrative example

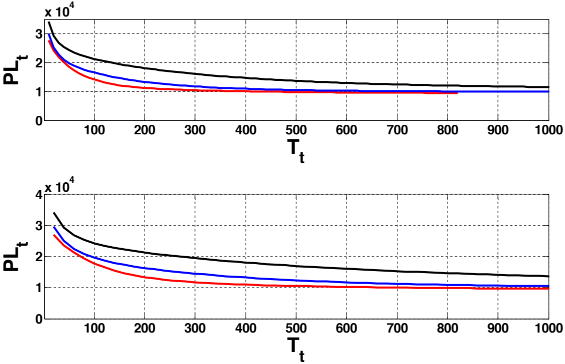


Figure: Trade-offs between $PL_t(\bar{p})$ and T_t for different decompositions and $\bar{p} = 10$ (top figure) and $\bar{p} = 20$ (bottom figure). Red: The two-neighborhoods case. Blue: The three-neighborhoods case. **Black**: The six-neighborhoods case.

Illustrative example

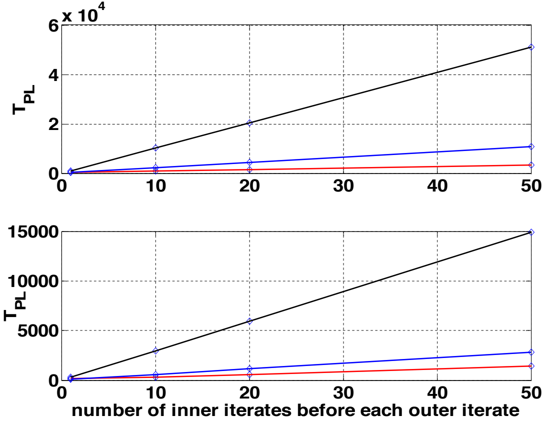


Figure: Trade-offs between the total number of iterations for convergence T_{PL} and $\bar{p}$ for different decompositions and $PL = 1$ percent (top figure) and $PL = 10$ percent (bottom figure). Red: The two-neighborhoods case. Blue: The three-neighborhoods case. **Black**: The six-neighborhoods case.

Example:

Inner iterate communication overhead: 1 second

Outer iterate communication overhead: 10 seconds

For the system decomposed into 3 neighborhoods with $\bar{p} = 10$:

Total communication overhead equals $(220 \times 10 + 2200 \times 1 =)4400$ seconds

Total computation time for producing the optimal inputs equals $(390.14 + 4400 =)4790.14$ seconds.

Without decomposition and inner iterates:

Total communication overhead equals $(950 \times 10 =)9500$ seconds

Total computation time for producing the optimal inputs equals $(174.126 + 9500 =)9674.126$ seconds.

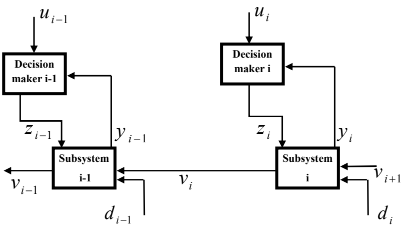


Figure: An automated irrigation network via distributed distant downstream feedback control. $z_i(s) = C_i(s)e_i(s),\ C_i(s) = \frac{K_i T_i s + K_i}{s(T_i F_i s + T_i)},\ e_i = u_i - y_i$.

Automated irrigation network model:

$$S_i:\ x_i[k+1] = A_i x_i[k] + B_i u_i[k] + F_i d_i[k] + v_i[k], \quad v_i[k] = M_i x_{i+1}[k],$$

$$y_i[k] = C_i x_i[k],$$

$$z_i[k] = D_i x_i[k],$$

$$i = 1, 2, ..., n,\ k \in \{0, 1, 2, ..., N-1\}.$$

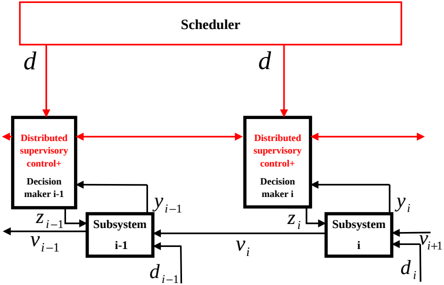


Figure: An automated irrigation network with distributed supervisory controller.

Cost functional:

$$\min_{\mathbf{u}=(\mathbf{u_1},\ldots,\mathbf{u_n})} \left\{ J(\mathbf{x}[0], \mathbf{d}, \mathbf{y}_d, u_1, \ldots, u_n), L_i \leq y_i[k], u_i[k] \leq H_i, 0 \leq z_i[k] \leq Z_i, \ \forall i, k \right\},$$

$$J(\mathbf{x}[0], \mathbf{d}, \mathbf{y}_d, u_1, \ldots, u_n) \doteq \sum_{i=1}^{n} \sum_{k=0}^{N-1} ||y_i[k] - y_i^d||_Q^2 + ||z_i[k]||_P^2 + ||u_i[k] - u_i[k-1]||_R^2.$$

Centralized technique (active set method)

Number of decision variables: n_d

Number of inequality constraints: n_c

$$\mathcal{C}_{cen}(n_d) \sim \mathcal{O}(n_d^3), \quad \text{(for a given } n_c\text{)}^3$$
$$\mathcal{C}_{cen}(n_c) \sim \mathcal{O}(n_c^3), \quad \text{(for a given } n_d\text{)}^4$$

$$\mathcal{C}_{cen}(n_d, n_c) \sim \mathcal{O}(n_d^3 \times n_c^3) \quad ^5$$

For automated irrigation networks: $n_d = nN$, $n_c = 6nN$

$$\boxed{\mathcal{C}_{cen}(n) \sim \mathcal{O}(n_d^3 \times n_c^3) \sim \mathcal{O}(n^6)}$$

---

[3] [ECC2009] M. S. K. Lau, S. P. Yue, K. V. Ling and J. M. Maciejowski.
[4] [TCST2010] Y. Wang and S. Boyd.
[5] [ECC2009],[TCST2010].

Distributed technique

For synchronized communication:

$$\mathcal{C}_{dis}(n) = \sum_{j=1}^{T_{PL}(n)} \mathcal{C}_j(n),$$

$T_{PL}(n)$: Total number of iterations for convergence

$\mathcal{C}_j(n)$: Maximum computation time of the decision maker with the dominating computational complexity

Assumption: Distributed decision makers also use active set method for their smaller QPs.

Number of decision variables of each decision maker: N

Number of inequality constraints of the dominating decision maker:

$$\begin{cases} N(4n+1), & \text{if } n \leq \frac{N}{2} \\ N(4\left\lfloor \frac{N}{2} \right\rfloor + 2), & \text{otherwise} \end{cases} .$$

Distributed technique

For a given n, the dominating decision maker remains constant for all iterations, whereby the dominating computational complexity $\mathcal{C}_j(n)$ also remains constant for all $j > 1$

$$\mathcal{C}_j(n) \doteq \mathcal{C}(n), \quad \forall j > 1.$$

For $j = 1$, it takes some time that variables to be placed into the cache memory

$$\mathcal{C}_1(n) \geq \mathcal{C}_j(n) = \mathcal{C}(n), \quad \forall j \geq 1.$$

$$\boxed{\mathcal{C}_{dis}(n) = \sum_{j=1}^{T_{PL}(n)} \mathcal{C}_j(n) = \mathcal{C}_1(n) + (T_{PL}(n) - 1)\mathcal{C}(n)}$$

Distributed technique

Number of inequality constraints of the dominating decision maker:

$$\begin{cases} N(4n+1), & \text{if } n \leq \frac{N}{2} \\ N(4\left\lfloor \frac{N}{2} \right\rfloor + 2), & \text{otherwise} \end{cases}.$$

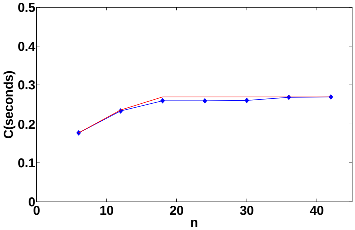
$$\Rightarrow$$

$$\mathcal{C}(n) \sim \begin{cases} \mathcal{O}(n), & \text{if } n \leq \frac{N}{2} \\ \alpha, & \text{otherwise} \end{cases}.$$

$$\mathcal{C}_1(n) = \eta, \qquad T_{PL}(n) = \beta n$$

$$\mathcal{C}_{dis}(n) = \mathcal{C}_1(n) + (T_{PL}(n) - 1)\mathcal{C}(n) \sim \begin{cases} \mathcal{O}(n^2), & \text{if } n \leq \frac{N}{2} \\ \mathcal{O}(n), & \text{otherwise} \end{cases}.$$

Simulation results



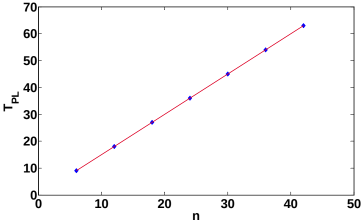


Figure: Left: $\mathcal{C}(n)$. Right: $T_{PL}(n)$.

$$\mathcal{C}(n) \approx \begin{cases} 0.00983n + 0.118 \sim \mathcal{O}(n), \text{if } n \leq 12 \\ 0.269, \qquad \text{otherwise} \end{cases} . \qquad T_{PL}(n) = 1.5n, \qquad \mathcal{C}_1(n) \approx \mathcal{C}_1 = 1.36.$$

Simulation results

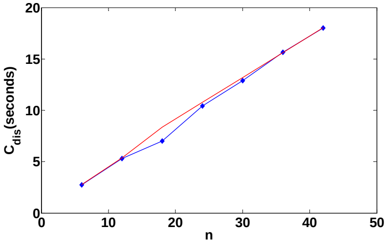


Figure: $\mathcal{C}_{dis}(n)$ versus n.

$$\mathcal{C}_{dis}(n) = \mathcal{C}_1(n) + (T_{PL}(n) - 1)\mathcal{C}(n) \quad (1)$$

$$\mathcal{C}(n) \approx \begin{cases} 0.00983n + 0.118 \sim \mathcal{O}(n), \text{if } n \leq 12 \\ 0.269, \quad \text{otherwise} \end{cases}. \qquad T_{PL}(n) = 1.5n, \qquad \mathcal{C}_1(n) \approx \mathcal{C}_1 = 1.36.$$

$$\mathcal{C}_{dis}(n) \approx \begin{cases} 0.0147n^2 + 0.167n + 1.242 \sim \mathcal{O}(n^2) & \text{if } n \leq 12 \\ 0.403n + 1.091 \sim \mathcal{O}(n), & \text{otherwise} \end{cases}. \quad (2)$$

Simulation result

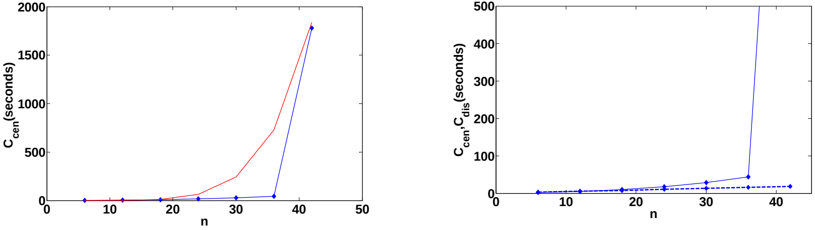


Figure: Left: $\mathcal{C}_{cen}(n)$. Right: $\mathcal{C}_{cen}(n)$: solid line, $\mathcal{C}_{dis}(n)$: dashed line.

$$\mathcal{C}_{cen} \approx (\frac{n}{12})^6 \sim \mathcal{O}(n^6). \tag{3}$$

Finding an analytical expression for T_{PL} (and therefore $\mathcal{C}_{dis} = \sum_{j=1}^{T_{PL}} \mathcal{C}_j$)

$$T_{PL} = F(\lambda_{m,l}, \pi_{m,l}, PL, \bar{p}, q, l).$$

Finding an analytical expression for communication overhead: Com

$$Com = G(\bar{p}, q, l).$$

Balancing interactions between control,computation,communication, and scalability to have the best possible performance: good quality control inputs with minimum overall computation time

$$\min_{\lambda_{m,l}, \pi_{m,l}, PL, \bar{p}, q, l} \left\{ \mathcal{C}_{dis} + Com, \quad \text{subject to constraints on } \lambda_{m,l},\ \pi_{m,l},\ PL \right\}$$

PL: Quality of control

$\lambda_{m,l}, \pi_{m,l}$: Convergence rate, quality of distributed computation

$\bar{p}$: Communication pattern

q,l: Scalability architecture

[ACC2010]B. T. Stewart, J. B. Rawlings, and S. J. Wright, Hierarchical cooperative distributed model predictive control, *2010 American Control Conference*, pp. 3963-3968, 2010.

[AUCC2012] A. Farhadi, M. Cantoni, and P. M. Dower, Performance and information pattern trade-offs in a consensus based distributed optimization method, *2012 Australian Control Conference*, 2012.

[ECC2009] M. S. K. Lau, S. P. Yue, K. V. Ling and J. M. Maciejowski, A Comparison of interior point and active set methods for FPGA implementation of model predictive control, *Proceedings of the European Control Conference*, pp. 156-161, August 2009.

[TCST2010] Y. Wang ans S. Boyd, Fast model predictive control using online optimization, *IEEE Transactions on Control Systems Technology*, 18(2), pp. 267 - 278, 2010.