Chordality for Model Predictive Control

Anders Hansson
Division of Automatic Control
Linköping University

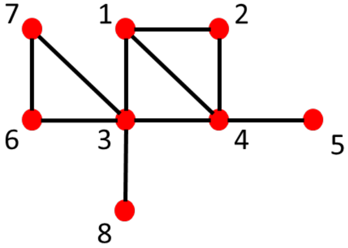
November 16, 2023

Outline

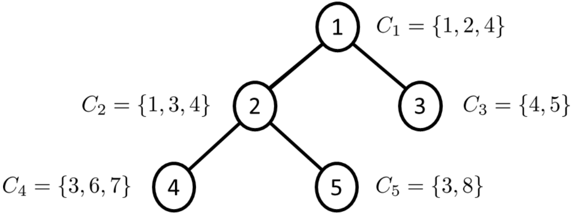
Simple Example

$$\underset{x}{\text{minimize}} \quad \bar{F}_1(x_1, x_3) + \bar{F}_2(x_1, x_2, x_4) + \bar{F}_3(x_4, x_5) + \bar{F}_4(x_3, x_4) + \bar{F}_5(x_3, x_6, x_7) + \bar{F}_6(x_3, x_8). \quad (1)$$

Has sparsity graph (edge between vertexes if components in same term)



Clique Tree for Sparsity Graph



We now assign one computational agent for each clique, and we may assign $\bar{F}_i$ to an agent if and only if the indexes of its variables belong to the corresponding clique. Hence we can assign $\bar{F}_1 + \bar{F}_4$ to C_2, $\bar{F}_2$ to C_1, $\bar{F}_3$ to C_3, $\bar{F}_5$ to C_4 and $\bar{F}_6$ to C_5. (Not unique assignment)

Message Passing or Dynamic Programming over Trees

Start with the leaves and compute for agents 3, 4, and 5

$$m_{31}(x_4) = \min_{x_5} \left\{ \bar{F}_3(x_4, x_5) \right\} \tag{2}$$

$$m_{42}(x_3) = \min_{x_6, x_7} \left\{ \bar{F}_5(x_3, x_6, x_7) \right\} \tag{3}$$

$$m_{52}(x_3) = \min_{x_8} \left\{ \bar{F}_6(x_3, x_8) \right\} \tag{4}$$

Then add the results from agents 4 and 5 to the functions of Agent 2 and compute

$$m_{21}(x_1, x_4) = \min_{x_3} \left\{ \bar{F}_1(x_1, x_3) + \bar{F}_4(x_3, x_4) + m_{42}(x_3) + m_{52}(x_3) \right\} \tag{5}$$

Finally add the results from agents 2 and 3 to the functions of Agent 1 and compute

$$\min_{x_1, x_2, x_4} \left\{ \bar{F}_2(x_1, x_2, x_4) + m_{31}(x_4) + m_{21}(x_1, x_4) \right\}$$

Comments

- Not easy in general to compute messages or value functions $m_{i,j}$.
- For linearly constrained convex quadratic problems the messages are convex quadratic functions with equality constraints.
- The dual variables can also be recovered.
- In fact results in a *multi-frontal factorization technique* for the KKT saddle point problem.
- Can be used to compute search directions in most optimization methods.
- All other computations in many optimization methods also distribute over the clique tree.
- In total 6 upward and 6 downward passes through the clique tree, of which only one pass involves significant computations, for each iteration in an IP algorithm

Interior-Point Methods

Consider the QP

$$\underset{z}{\text{minimize}}\ \frac{1}{2}z^T \mathcal{Q} z + q^T z \tag{6}$$

$$\text{subj. to } \mathcal{A}z = b \tag{7}$$

$$\mathcal{D}z \leq e \tag{8}$$

where $\mathcal{Q} \succeq 0$, and $\mathcal{A}$ has full row rank.

KKT optimality conditions:

$$\begin{bmatrix} \mathcal{Q} & \mathcal{A}^T & \mathcal{D}^T & \\ \mathcal{A} & & & \\ \mathcal{D} & & & I \\ & & & M \end{bmatrix} \begin{bmatrix} z \\ \lambda \\ \mu \\ s \end{bmatrix} = \begin{bmatrix} -q \\ b \\ e \\ 0 \end{bmatrix} \tag{9}$$

and $(\mu, s) \geq 0$, where $M = \text{diag}(\mu)$.

Search Directions

Linearize:

$$\begin{bmatrix} \mathcal{Q} & \mathcal{A}^T & \mathcal{D}^T & \\ \mathcal{A} & & & \\ \mathcal{D} & & & I \\ & & S & M \end{bmatrix} \begin{bmatrix} \Delta z \\ \Delta \lambda \\ \Delta \mu \\ \Delta s \end{bmatrix} = \begin{bmatrix} r_z \\ r_\lambda \\ r_\mu \\ r_s \end{bmatrix} \tag{10}$$

where $S = \mathrm{diag}(s)$, and where $r = (r_z, r_\lambda, r_\mu, r_s)$ is residual vector.

Reduced KKT system

Equivalently $\Delta s = r_\mu - \mathcal{D}\Delta z$, $\Delta\mu = S^{-1}(r_s - M\Delta s)$ and

$$\begin{bmatrix} \mathcal{Q} + \mathcal{D}^T S^{-1} M \mathcal{D} & \mathcal{A}^T \\ \mathcal{A} & \end{bmatrix} \begin{bmatrix} \Delta z \\ \Delta\lambda \end{bmatrix} = \begin{bmatrix} r_z - \mathcal{D}^T S^{-1}(r_s - M r_\mu) \\ r_\lambda \end{bmatrix}. \tag{11}$$

Unique solution iff

$$\mathcal{Q}_s = \mathcal{Q} + \mathcal{D}^T S^{-1} M \mathcal{D} \tag{12}$$

is positive definite on the null-space of $\mathcal{A}$.

Parametric QPs

Consider

$$\underset{z}{\text{minimize}}\ \frac{1}{2} z^T M z + m^T z \tag{13}$$

$$\text{subj. to } Cz = d \tag{14}$$

with C full row rank and $M \succeq 0$.

KKT conditions:

$$\begin{bmatrix} M & C^T \\ C & \end{bmatrix} \begin{bmatrix} z \\ \lambda \end{bmatrix} = \begin{bmatrix} -m \\ d \end{bmatrix}.$$

with unique solution if and only if $M + C^T C \succ 0$.

Partitioned Problem

Let

$$M = \begin{bmatrix} Q & S \\ S^T & R \end{bmatrix}; \quad C = \begin{bmatrix} A & B \\ & D \end{bmatrix}; \quad d = \begin{bmatrix} e \\ f \end{bmatrix}; \quad m = \begin{bmatrix} q \\ r \end{bmatrix}; \quad z = \begin{bmatrix} x \\ y \end{bmatrix}$$

with A full row rank.

Solve

$$\underset{x}{\text{minimize}} \ \frac{1}{2} \begin{bmatrix} x \\ y \end{bmatrix}^T \begin{bmatrix} Q & S \\ S^T & \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + q^T x \tag{15}$$

$$\text{subj. to } Ax + By = e \tag{16}$$

parametrically with respect to all y.

KKT Conditions for Parametric Problem

$$\begin{bmatrix} Q & A^T \\ A & \end{bmatrix} \begin{bmatrix} x \\ \mu \end{bmatrix} = \begin{bmatrix} -q - Sy \\ e - By \end{bmatrix}.$$

- Solution x will be affine in y
- Results in a quadratic message in y.
- The 1,1-block of $M + C^T C$ is $Q + A^T A$, which by the Schur complement formula is positive definite, which implies unique solution

Rank Condition

In case A does not have full row rank, perform a rank-revealing factorization

$$\begin{bmatrix} \bar{A}_1 \\ 0 \end{bmatrix} x + \begin{bmatrix} \bar{B}_1 \\ \bar{B}_2 \end{bmatrix} y = \begin{bmatrix} \bar{e}_1 \\ \bar{e}_2 \end{bmatrix}$$

and append the constraint $\bar{B}_2 y = \bar{e}_2$ to belong to

$$Dy = f$$

- Step-lenght computations also distribute over clique tree.
- Generalizes to Augmented Lagrangian (AL) methods and Levenberg Marquardt (LM) method.

Model Predictive Control (MPC)

$$\underset{x,u}{\text{minimize}} \frac{1}{2} \sum_{k=0}^{N-1} \begin{bmatrix} x_k \\ u_k \end{bmatrix}^T Q \begin{bmatrix} x_k \\ u_k \end{bmatrix} + \frac{1}{2} x_N^T S x_N \tag{17}$$

$$\text{subj. to } x_{k+1} = Ax_k + Bu_k, \quad x_0 = \bar{x} \tag{18}$$

where $Q \succeq 0$ and $S \succeq 0$

Let $\mathcal{I}_{\mathcal{C}_k}(x_k, u_k, x_{k+1})$ be indicator function for

$$\mathcal{C}_k = \{(x_k, u_k, x_{k+1}) \mid x_{k+1} = Ax_k + Bu_k\}$$

and $\mathcal{I}_{\mathcal{D}}(x_0)$ indicator function for

$$\mathcal{D} = \{x_0 \mid x_0 = \bar{x}\}$$

Equivalent Formulation

$$\underset{x,u}{\text{minimize}} \quad \bar{F}_1(x_0, u_0, x_1) + \cdots + \bar{F}_N(x_{N-1}, u_{N-1}, x_N), \qquad (19)$$

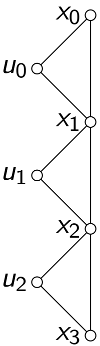
where

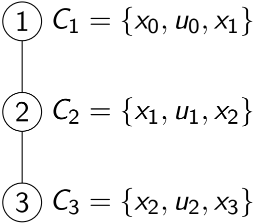
$$\bar{F}_1(x_0, u_0, x_1) = \mathcal{I}_{\mathcal{D}}(x_0) + \frac{1}{2}\begin{bmatrix} x_0 \\ u_0 \end{bmatrix}^T Q \begin{bmatrix} x_0 \\ u_0 \end{bmatrix} + \mathcal{I}_{\mathcal{C}_0}(x_0, u_0, x_1)$$

$$\bar{F}_{k+1}(x_k, u_k, x_{k+1}) = \frac{1}{2}\begin{bmatrix} x_k \\ u_k \end{bmatrix}^T Q \begin{bmatrix} x_k \\ u_k \end{bmatrix} + \mathcal{I}_{\mathcal{C}_k}(x_k, u_k, x_{k+1})$$

$$\bar{F}_N(x_{N-1}, u_{N-1}, x_N) = \frac{1}{2}\begin{bmatrix} x_{N-1} \\ u_{N-1} \end{bmatrix}^T Q \begin{bmatrix} x_{N-1} \\ u_{N-1} \end{bmatrix} + \mathcal{I}_{\mathcal{C}_{N-1}}(x_{N-1}, u_{N-1}, x_N) + \frac{1}{2} x_N^T S x_N$$

Sparsity Graph and Clique Tree

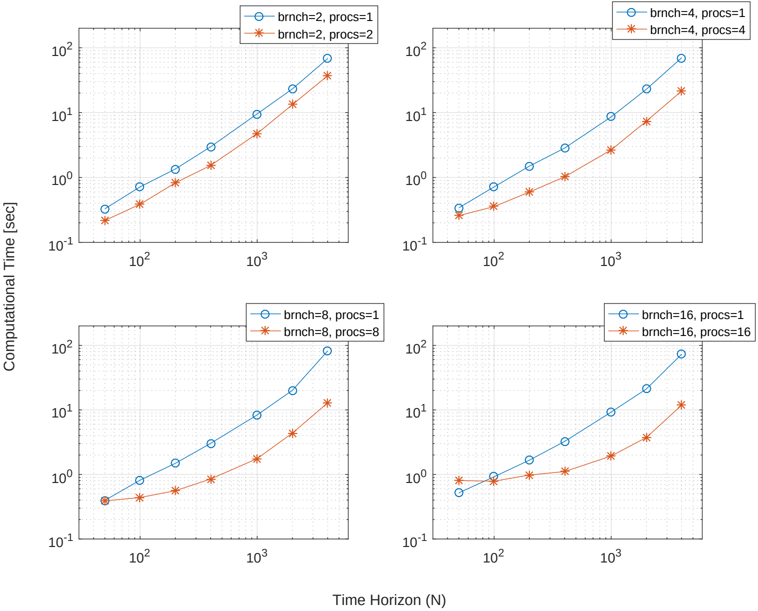




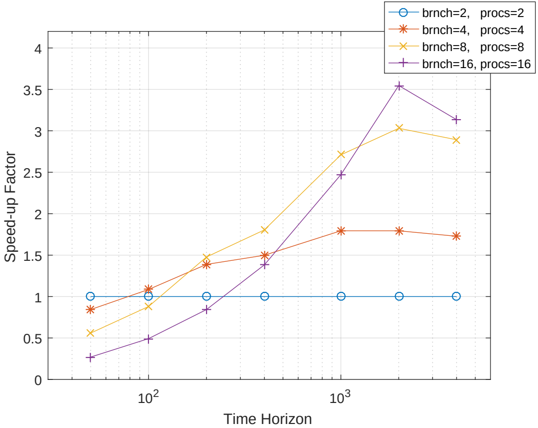
Assign $\bar{F}_k$ to C_k.

Can just as well take C_2 or C_3 as root! Possible to do even more parallelization. (details omitted)

Julia Implementation for Parallel Computations



Speed-up Factor



Stochastic MPC

$$\underset{x,u}{\text{minimize}} \sum_{j=1}^{M} \omega_j \left(\frac{1}{2} \sum_{k=0}^{N-1} \begin{bmatrix} x_k^j \\ u_k^j \end{bmatrix}^T Q \begin{bmatrix} x_k^j \\ u_k^j \end{bmatrix} + \frac{1}{2} (x_N^j)^T S x_N^j \right) \quad (20)$$

$$\text{subj. to } x_{k+1}^j = A_k^j x_k^j + B_k^j u_k^j + v_k^j, \quad x_0^j = \bar{x} \quad (21)$$

$$\bar{C} u = 0 \quad (22)$$

where $u = (u^1, u^2, \dots, u^M)$ with $u^j = (u_0^j, u_1^j, \dots, u_{N-1}^j)$, and

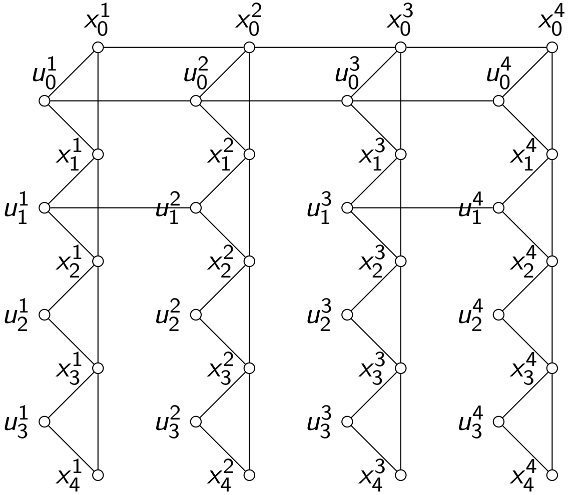
$$\bar{C} = \begin{bmatrix} C_{1,2} & -C_{1,2} & & & \\ & C_{2,3} & -C_{2,3} & & \\ & & \ddots & \ddots & \\ & & & C_{M-1,M} & -C_{M-1,M} \end{bmatrix}$$

with

$$C_{j,j+1} = \begin{bmatrix} I & 0 \end{bmatrix}$$

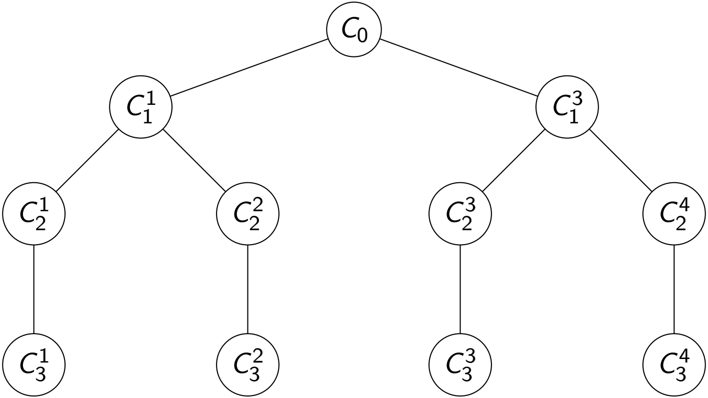
The constraint $\bar{C} u = 0$ is the non-ancipativity constraint.

Sparsity Graph



Make chordal embedding.

Clique Tree



Robust MPC can be done similarly for a QCQP.

Distributed MPC

$$\underset{x,u}{\text{minimize}} \sum_{i=1}^{m} \left(\sum_{k=1}^{N} h_i(x_i(k), u_i(k)) + h_i^f(x_i(N+1) \right) \tag{23}$$

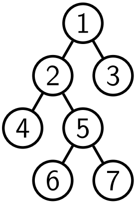
$$\text{subj. to } x_i(k+1) = f_i(x_i(k), u_i(k)) + \sum_{j \in \mathcal{N}(i)} g_j(x_j(k), u_j(k))$$

$$x_i(1) = \bar{x}_i, \quad k = 1, \ldots, N, \quad i = 1, \ldots, m \tag{24}$$

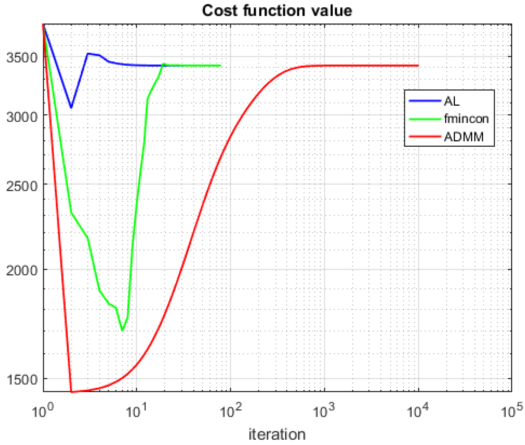
Example

$$
\begin{aligned}
\underset{x,u}{\text{minimize}} \quad & \sum_{i=1}^{7}\Big(\sum_{k=1}^{N} r_x x_i(k)^2 + r_u u_i(k)^2\Big) + r_x x_i(N+1)^2 && (25)\\
\text{subj. to} \quad & x_1(k+1) = \alpha_1 x_1(k)^2 + \beta_1 u_1(k) + x_2(k) + x_3(k)\\
& x_2(k+1) = \alpha_2 x_2(k)^2 + \beta_2 u_2(k) + x_4(k) + x_5(k)\\
& x_3(k+1) = \alpha_3 x_3(k)^2 + \beta_3 u_3(k)\\
& x_4(k+1) = \alpha_4 x_4(k)^2 + \beta_4 u_4(k)\\
& x_5(k+1) = \alpha_5 x_5(k)^2 + \beta_5 u_5(k) + x_6(k) + x_7(k)\\
& x_6(k+1) = \alpha_6 x_6(k)^2 + \beta_6 u_6(k)\\
& x_7(k+1) = \alpha_7 x_7(k)^2 + \beta_7 u_7(k), \quad k = 1, \dots, N\\
& x_i(1) = \bar{x}_i, \quad i = 1, \dots, 7 && (26)
\end{aligned}
$$

Clique Tree

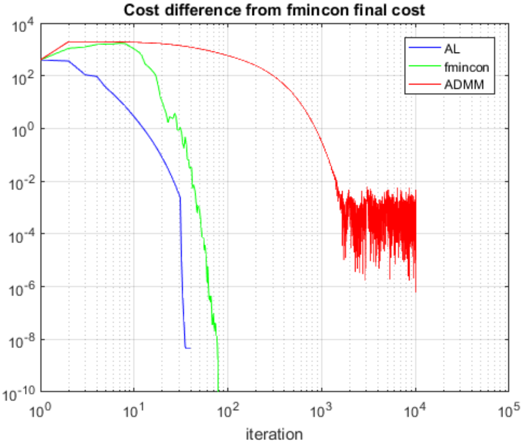


Convergence

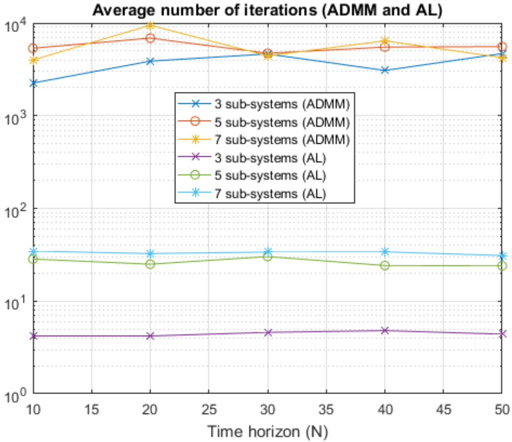


Augmented Lagrangian (AL) method

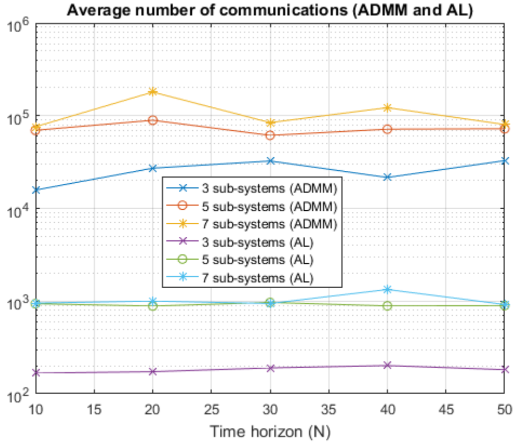
Convergence ctd.



Average Iterations Versus Time Horizon



Average Communications Versus Time Horizon



Variable Horizon (VH) MPC

$$
\begin{array}{lll}
\text{minimize}_{u,\xi,N} & J_N(\xi,u) + cN & \\
\text{subject to} & \xi_{k+1} = F\xi_k + Gu_k & \text{for } k = 0,\ldots,N-1 \\
 & c_k \leq C\xi_k + Du_k \leq d_k & \text{for } k = 0,\ldots,N-1 \\
 & c_N \leq C_N\xi_N \leq d_N, &
\end{array}
\tag{27}
$$

where

$$
J_N(\xi,u) = \frac{1}{2}\sum_{k=0}^{N-1} \begin{bmatrix} \xi_k \\ u_k \end{bmatrix}^T \begin{bmatrix} Q_1 & Q_{12} \\ Q_{12}^T & Q_2 \end{bmatrix} \begin{bmatrix} \xi_k \\ u_k \end{bmatrix} + \frac{1}{2}\xi_N^T Q_N \xi_N
$$

Equalivalent Formulation

Inner problem:

$$
\begin{aligned}
\text{minimize}_{\xi,u} \quad & J_N(\xi_0, u) \\
\text{subj. to} \quad & \xi_{k+1} = F\xi_k + Gu_k & \text{for } k = 0, \ldots, N-1 \\
& c_k \leq C\xi_k + Du_k \leq d_k & \text{for } k = 0, \ldots, N-1 \\
& c_N \leq C_N \xi_N \leq d_N, &
\end{aligned}
\tag{28}
$$

Denote the solution of this problem by $(\xi^\star, u_N^\star)$.

Outer problem:

$$
\text{minimize}_N \quad J_N(\xi^\star, u_N^\star) + cN \tag{29}
$$

Inner Problem

KKT equation for OSQP:

$$\begin{bmatrix} P + I\sigma & A^T \\ A & -\rho^{-1} I \end{bmatrix} \begin{bmatrix} \tilde{x}^{k+1} \\ \tilde{\nu}^{k+1} \end{bmatrix} = \begin{bmatrix} \sigma x^k - q \\ z^k - \rho^{-1} y^k \end{bmatrix}, \tag{30}$$

- Need to solve for many different values of N.
- Time for permutation and factorization of matrix comparable to time for iterative ADMM steps.
- Forward recursion over N.

Forward recursion

KKT matrix can be written as

$$P_0^T K P_0 = LDL^T$$

with matrices

$$L = \begin{bmatrix} L_0 & & & & & \\ \bar{Y}_{01}^T L_0 & L_1 & & & & \\ & \bar{Y}_{11}^T L_1 & \ddots & & & \\ & & \ddots & L_{2N} & & \\ & & & Y_{2N,1}^T L_{2N} & L_{2N+1} \end{bmatrix},$$

$$D = \begin{bmatrix} D_0 & & & & \\ & -D_1 & & & \\ & & \ddots & & \\ & & & D_{2N} & \\ & & & & -D_{2N+1} \end{bmatrix}.$$

where P_0 is a permutation matrix.

Parallel Computations

Let

$$P_0^T K P_0 = \begin{bmatrix} X & U & \\ U^T & Y & V \\ & V^T & Z \end{bmatrix}$$

and

$$P^T X P = LDL^T, \qquad S^T Z S = MEM^T, \tag{31}$$

Then

$$P_2^T P_1^T P_0^T K P_0 P_1 P_2 = \begin{bmatrix} L & & \\ 0 & M & \\ U^T P L^{-T} D^{-1} & V S M^{-T} E^{-1} & I \end{bmatrix} \begin{bmatrix} D & & \\ & E & \\ & & \hat{Y} \end{bmatrix}$$
$$\times \begin{bmatrix} L^T & 0 & D^{-1} L^{-1} P^T U \\ & M^T & E^{-1} M^{-1} S^T V^T \\ & & I \end{bmatrix}$$

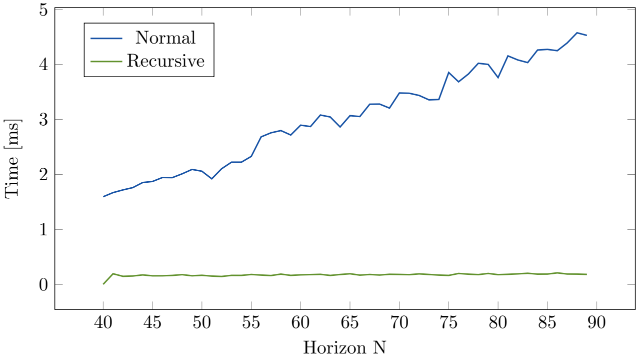
where $\hat{Y} = Y - \bar{U} D^{-1} \bar{U}^T - \bar{V} E^{-1} \bar{V}^T$.

Implementation

- Outer problem implemented in C++ using heuristic search rules.
- Inner problem implemented directly in OSQP to maximize efficiency.
- Code available on GitHub[1].
- All matrix data are saved in the Compressed Sparse Column (CSC) matrix format.
- The CSC format allows to cheaply add or remove columns at the end of the matrix, so updating the A, P, and P_0 matrices is straightforward.
- Only the factorization step in the OSQP implementation is changed.

[1] https://github.com/laperss/osqp-recursive-ldl

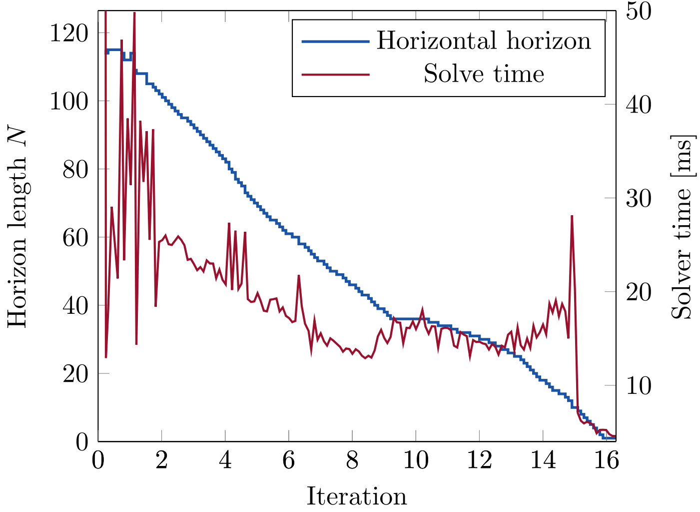
Comparison of Computational Time for increasing N



Outdoor Flight Experiments

- DJI Matrice 100 drone
- NUC 7i7BNB flight computer
- Virtual boat simulated on a separate ground laptop.
- A vertical and a horizontal controller, running at 10 Hz.
- Moderate wind conditions.
- Landing is performed while the boat travels.

Horizon and Solve Time



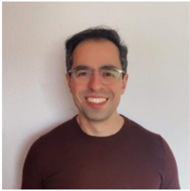
Wind Gust at $t = 9$ s.

Scale on x-axis should be time in s.

Summary

- Optimization methods over trees based on dynamic programming or message passing to compute search directions.
- Needs less communication than other distributed algorithms
- Model predictive control (MPC)
- Parallel MPC
- Stochastic MPC
- Variable horizon MPC
- Distributed localization (not covered)
- Distributed robustness analysis (not covered)

Acknowledgements

Collaboration with *Sina Khoshfetrat Pakazad*, *Shervin Parvini Ahmadi*, *Linnea Persson* and *Bo Wahlberg*

Publications

A. Hansson and S. Khoshfetrat Pakazad. "Exploiting Chordality in Optimization Algorithms for Model Predictive Control", Large-scale and distributed optimization, Lecture Notes in Mathematics, No. 2227, 11-32, 2018.

S. P. Ahmadi, A. Hansson, "Parallel Exploitation for Tree-Structured Coupled Quadratic Programming in Julia", Proceedings of the 22nd International Conference on System Theory, Control and Computing (ICSTCC), 597-602, 2018.

S. P. Ahmadi, A. Hansson, "Distributed optimal control of nonlinear systems using a second-order augmented Lagrangian method", European Journal of Control, 70, 2023.

L. Persson, A. Hansson, B. Wahlberg. "An Optimization Algorithm based on Forward Recursion with Applications to Variable Horizon MPC", European Journal of Control, 2023