Distributed MPC: Data Generation and Manipulations

João Mota

February 15, 2013

This document describes the model we used for MPC, how the data was generated, and how we solved the problem with D-ADMMp.

1 Introduction

Model Predictive Control (MPC) is a popular and widely used control strategy for discrete-time systems. To introduce it, consider a system modeled by the state-space equation $x[t+1] = f^t(x[t], u[t])$, where $x[t]$ and $u[t]$ are, respectively, the state and control at time t. We assume the state is measured, possibly with inaccuracies, at each time instant. The function f^t models the system at time t and, many times, it is inaccurate and/or contains random factors (e.g., modeling noise). The goal of MPC is to drive the state from an initial value $x[0]$ to other predetermined value at time horizon T. Since there may exist many ways to achieve this goal, one might wish to do it while minimizing some function of the state and input, $g^t(x[t], u[t])$. Such function can model, for example, the energy consumed at time t. All this can be formalized with the optimization problem

$$\begin{array}{ll} \underset{\{x[t]\}_{t=0}^{T}, \{u[t]\}_{t=0}^{T-1}}{\text{minimize}} & h(x[T]) + \sum_{t=0}^{T-1} g^t(x[t], u[t]) \\ \text{subject to} & x[t+1] = f^t(x[t], u[t]), \quad t = 0, 1, \ldots, T-1 \\ & x[0] = x^0, \end{array} \tag{1}$$

where the function $h(\cdot)$ expresses the goal of achieving the desired state at time T, and x^0 is the measured state at time $t = 0$. In words, (1) finds a sequence of inputs $u[0]$, $u[1]$, $\ldots$, $u[T-1]$ and the corresponding sequence of (induced) states $x[0]$, $x[1]$, $\ldots$, $x[T]$ that minimize the objective of (1) while satisfying the system dynamics. The objective of (1) is a tradeoff between minimizing $g_t(x[t], u[t])$ at each instant and achieving the goal (final state). The MPC strategy consists of solving (1), applying $u[0]$ to the system, measuring $x[1]$, setting $t = 0$, and repeating the process over again. That is, at each time instant, we compute all the future inputs and all the future states, although we will only use $u[0]$. The more accurate the system model is, the more similar the predicted states and the measured states will be. MPC is thus a conservative way of dealing with model inaccuracies and noise, and it consists of solving a sequence of problems with the format of (1). Next we present our assumptions on problem (1) and subsequent simplifications.

2 Our model

Our goal is to solve (1) with D-ADMMp, an algorithm that solves problems with the format

$$\underset{z \in \mathbb{R}^K}{\text{minimize}} \quad f_1(z_{S_1}) + f_2(z_{S_2}) + \cdots + f_P(z_{S_P}), \tag{2}$$

where $S_p \subseteq \{1, 2, \ldots, K\}$ is the set of indices of the variable $z \in \mathbb{R}^K$ node p depends on. Problem (2) is associated to a network like the one in Figure 1. There are P nodes and E edges, and node p, besides knowing only the function f_p, can only communicate with its neighbors.

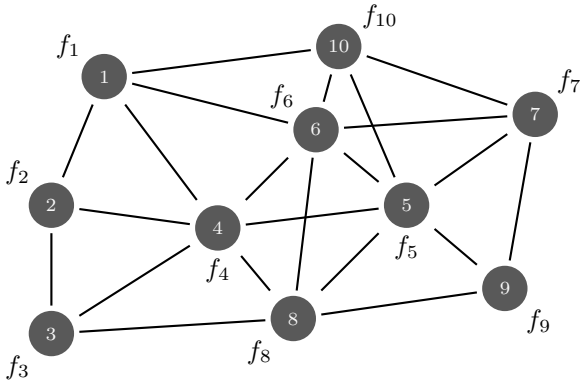

Figure 1: Example of a communication network. Node p only knows f_p, but cooperates with its neighbors in order to solve the global problem (2).

First level of decomposition. Problem (1) is too generic to be recast as (2). The following decomposition, for example, enables recasting (1) as (2):

$$\begin{aligned}
x[t] &= \Big(x_1[t], x_2[t], \dots, x_P[t]\Big), \qquad t = 0, 1, \dots, T \\
u[t] &= \Big(u_1[t], u_2[t], \dots, u_P[t]\Big), \qquad t = 0, 1, \dots, T-1 \\
x_p[t+1] &= f_p^t\Big(\{x_j[t], u_j[t]\}_{j\in\Omega_p}\Big), \qquad t = 0, 1, \dots, T-1 \\
h(x[T]) &= \sum_{p=1}^{P} h_p\Big(\{x_j[T]\}_{j\in\Omega_p}\Big) \\
g^t(x[t], u[t]) &= \sum_{p=1}^{P} g_p^t\Big(\{x_j[t], u_j[t]\}_{j\in\Omega_p}\Big), \qquad t = 0, 1, \dots, T-1\,,
\end{aligned}$$

where $x_p[t] \in \mathbb{R}^{n_p}$ and $u_p[t] \in \mathbb{R}^{m_p}$ are, respectively, the state and input of the subsystem located at node p, and $\Omega_p \subseteq \{1, \dots, P\}$ is the set of neighbors node p interacts with, either in terms of dynamics or objective. With this decomposition, problem (1) can be written as

$$\begin{array}{ll}
\underset{\{x_p[t]\}_{t=0,p=1}^{T,P},\{u_p[t]\}_{t=0,p=1}^{T-1,P}}{\text{minimize}} & \sum_{p=1}^{P}\Big(h_p\Big(\{x_j[T]\}_{j\in\Omega_p}\Big) + \sum_{t=0}^{T-1} g_p^t\Big(\{x_j[t], u_j[t]\}_{j\in\Omega_p}\Big)\Big) \\
\text{subject to} & x_p[t+1] = f_p^t\Big(\{x_j[t], u_j[t]\}_{j\in\Omega_p}\Big), \qquad t = 0, 1, \dots, T-1\,, \quad p = 1, \dots, P \\
& x_p[0] = x_p^0\,, \qquad p = 1, \dots, P\,.
\end{array} \tag{3}$$

Problem (3) can be promptly solved with D-ADMMp, for any choice of functions that makes the overall problem convex, with the objective function closed and convex. In the next level of decomposition, we select specific functions and make further assumptions.

Second level of decomposition. We simplify the previous decomposition by making some assumptions commonly found in distributed settings, namely linear time-invariant subsystems and quadratic functions. Furthermore, we assume that there is coupling only through the dynamics, i.e., through the function f_p^t. In particular we assume, for $p = 1, \dots, P$,

$$x_p[t+1] = A_{pp}x_p[t] + B_{pp}u_p[t] + \sum_{j\in\Omega_p'} B_{pj}u_j[t]\,, \qquad t = 0, \dots, T-1 \tag{4}$$

$$\begin{aligned}
h_p(\{x_j[T]\}_{j\in\Omega_p}) &= x_p[T]^\top \bar{Q}_p^f\, x_p[T] \\
g_p^t(\{x_j[t], u_j[t]\}_{j\in\Omega_p}) &= x_p[t]^\top \bar{Q}_p\, x_p[t] + u_p[t]^\top \bar{R}_p\, u_p[t]\,, \qquad t = 0, \dots, T-1\,,
\end{aligned}$$

where we decomposed Ω_p into $\Omega_p = \{p\} \cup \Omega'_p$. With these assumptions, problem (3) becomes

$$\begin{array}{ll} \underset{\substack{x_1,\dots,x_P \\ u_1,\dots,u_P}}{\text{minimize}} & \sum_{p=1}^{P} u_p^\top R_p u_p + x_p^\top Q_p x_p \\ \text{subject to} & x_p = C_p\{u_j\}_{j\in\mathcal{S}_p} + D_p^0\,, \qquad p = 1,\dots,P\,, \end{array} \tag{5}$$

where, for each p,

$$\begin{aligned} x_p &= (x_p[0], x_p[1], \dots, x_p[T-1], x_p[T]) \in \mathbb{R}^{Tn_p} \\ u_p &= (u_p[0], u_p[1], \dots, u_p[T-1]) \in \mathbb{R}^{Tm_p} \\ R_p &= I_T \otimes \bar{R}_p \\ Q_p &= \begin{bmatrix} I_T \otimes \bar{Q}_p & 0 \\ 0 & \bar{Q}_p^f \end{bmatrix}, \end{aligned}$$

and I_T is the identity matrix in $\mathbb{R}^T$. To arrive at the constraint in (5), we just iterated (4):

$$\begin{aligned} x_p[0] &= x_p^0 \\ x_p[1] &= A_{pp}x_p[0] + \sum_{j\in\Omega_p} B_{pj}u_j[0] \\ x_p[2] &= A_{pp}x_p[1] + \sum_{j\in\Omega_p} B_{pj}u_j[1] \\ &= A_{pp}^2 x_p[0] + A_{pp}\sum_{j\in\Omega_p} B_{pj}u_j[0] + \sum_{j\in\Omega_p} B_{pj}u_j[1] \\ x_p[3] &= A_{pp}x_p[2] + \sum_{j\in\Omega_p} B_{pj}u_j[2] \\ &= A_{pp}^3 x_p[0] + A_{pp}^2\sum_{j\in\Omega_p} B_{pj}u_j[0] + A_{pp}\sum_{j\in\Omega_p} B_{pj}u_j[1] + \sum_{j\in\Omega_p} B_{pj}u_j[2] \\ &\vdots \end{aligned}$$

from which

$$x_p = C_p\{u_j\}_{j\in\Omega_p} + D_p^0\,,$$

where

$$C_p := \begin{bmatrix} 0 & 0 & 0 & 0 & \cdots & 0 \\ B_p & 0 & 0 & 0 & \cdots & 0 \\ A_{pp}B_p & B_p & 0 & 0 & \cdots & 0 \\ A_{pp}^2B_p & A_{pp}B_p & B_p & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & & \vdots \\ A_{pp}^{T-2}B_p & A_{pp}^{T-3}B_p & A_{pp}^{T-4}B_p & & \ddots & 0 \\ A_{pp}^{T-1}B_p & A_{pp}^{T-2}B_p & A_{pp}^{T-3}B_p & A_{pp}^{T-4}B_p & \cdots & B_p \end{bmatrix}, \qquad D_p^0 := \begin{bmatrix} I \\ A_{pp} \\ A_{pp}^2 \\ A_{pp}^3 \\ \vdots \\ A_{pp}^{T-1} \\ A_{pp}^T \end{bmatrix} x_p^0\,, \tag{6}$$

and $B_p := \begin{bmatrix} B_{pj} \end{bmatrix}_{j\in\Omega_p}$, where the operation $[\cdot]$ denotes horizontal concatenation.

Problem (5) can clearly be simplified by eliminating x. Namely, its objective becomes

$$\begin{aligned} &\sum_{p=1}^{P} u_p^\top R_p u_p + \Big(C_p\{u_j\}_{j\in\Omega_p} + D_p^0\Big)^\top Q_p \Big(C_p\{u_j\}_{j\in\Omega_p} + D_p^0\Big) \\ &= \sum_{p=1}^{P} u_p^\top R_p u_p + \{u_j\}_{j\in S_p}^\top C_p^\top Q_p C_p\{u_j\}_{j\in S_p} + 2D_p^{0\top} Q_p C_p\{u_j\}_{j\in S_p} + D_p^{0\top} Q_p D_p^0 \\ &= \sum_{p=1}^{P} \{u_j\}_{j\in S_p}^\top E_p\{u_j\}_{j\in S_p} + 2(C_p^\top Q_p D_p^0)^\top\{u_j\}_{j\in S_p} + D_p^{0\top} Q_p D_p^0\,, \end{aligned} \tag{7}$$

where E_p is a matrix obtained by summing R_p with $C_p^\top Q_p C_p$ in the correct entries; this depends on how we define the variable. Problem (5) becomes

$$\underset{u_1,\ldots,u_P}{\text{minimize}} \ \sum_{p=1}^{P} \{u_j\}_{j\in S_p}^\top E_p \{u_j\}_{j\in S_p} + w_p^\top \{u_j\}_{j\in S_p}\,, \tag{8}$$

where

$$w_p := 2(C_p^\top Q_p D_p^0)\,. \tag{9}$$

Problem (8) clearly has the format of (2).

3 Algorithm

Algorithm 1 shows the application of D-ADMMp to solve (8). The variable is

$$u = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_P \end{bmatrix} \in \mathbb{R}^{PTm_p}\,.$$

At each iteration, node p has to solve

$$\begin{aligned} &\underset{\{u_j\}_{j\in S_p}}{\text{minimize}} \ \{u_j\}_{j\in S_p}^\top E_p \{u_j\}_{j\in S_p} + w_p^\top \{u_j\}_{j\in S_p} + v^\top \{u_j\}_{j\in S_p} + \{u_j\}_{j\in S_p}^\top \Phi \{u_j\}_{j\in S_p} \\ \iff &\underset{\{u_j\}_{j\in S_p}}{\text{minimize}} \ \{u_j\}_{j\in S_p}^\top (E_p + \Phi) \{u_j\}_{j\in S_p} + (w_p + v)^\top \{u_j\}_{j\in S_p}\,, \end{aligned}$$

where Φ is a diagonal matrix; both Φ and v are provided by D-ADMMp. The solution to this problem can be computed in closed-form:

$$\{u_j^\star\}_{j\in S_p} = -\frac{1}{2}(E_p + \Phi)^{-1}(w_p + v)\,. \tag{10}$$

Algorithm 1 D-ADMMp for MPC

Input (for node p):

Matrices $\bar{R}_p \in \mathbb{R}^{m_p \times m_p}$, $\bar{Q}_p, \bar{Q}_p^f, A_{pp} \in \mathbb{R}^{n_p \times n_p}$, $B_{pp} \in \mathbb{R}^{n_p \times m_p}$; for the set of nodes $j \in \Omega_p'$ that influences node p, B_{pj};

Time horizon T;

Measurement of the state $x_p^0 \in \mathbb{R}^{n_p}$

Initialization (for node p):

Form $R_p = I_T \otimes \bar{R}_p$ and $Q_p = \mathrm{Diag}(I_T \otimes \bar{Q}_p, \bar{Q}_p^f)$, and also C_p, D_p^0, as in (6);

Form E_p and in (7) and w_p as in (9);

Let S_p denote the set of indices of u node p depends on, and let $u^{(p)}$ be the copy of the components of z held at node p; let $\mathcal{V}_l$ the set of nodes depending on the lth component of z; also, let $\mathcal{C}(p)$ denote the color of node p;

Set $\gamma_l^{(p),1} = u_l^{(p),1} = 0$, for $l \in S_p$, and $k = 1$

Algorithm:

repeat

  for $c = 1, \ldots, C$ **do**

    for all $p \in \mathcal{C}_c$ [in parallel] **do**

      for all $l \in S_p$ **do**

$$v_l^{(p),k} = \gamma_l^{(p),k} - \rho \sum_{\substack{j \in \Omega_p' \cap \mathcal{V}_l \\ C(j) < c}} u_l^{(j),k+1} - \rho \sum_{\substack{j \in \Omega_p' \cap \mathcal{V}_l \\ C(j) > c}} u_l^{(j),k}$$

      end for

      Solve the linear system (10), $\{u_j^{(p),k+1}\}_{j \in S_p} = -\frac{1}{2}(E_p + \Phi)^{-1}(w_p + v^{(p),k})$

      For each component $l \in S_p$, send $u_l^{(p),k+1}$ to $\Omega_p' \cap \mathcal{V}_l$

    end for

  end for

  for all $p \in \mathcal{V}$ and $l \in S_p$ [in parallel] **do**

$$\gamma_l^{(p),k+1} = \gamma_l^{(p),k} + \rho \sum_{j \in \Omega_p' \cap \mathcal{V}_l} (u_l^{(p),k+1} - u_l^{(j),k+1})$$

  end for

  $k \leftarrow k + 1$

until some stopping criterion is met