Implementation Notes for the Distributed LP Framework

Mohamed Abdlekader

April 2017

1 Introduction

This documents describes some implementation details of the distributed LP framework. This document uses the corresponding *IFAC2017* paper as a reference.

2 Dynamics

$$\mathbf{x}^{+} = \mathbf{x} + (\mathbf{B}_{\text{in}} - \mathbf{B}_{\text{out}})\mathbf{u} \tag{1}$$

Also,

$$\mathbf{B} = \mathbf{B}_{\text{in}} - \mathbf{B}_{\text{out}} \in \mathcal{R}^{n_s \times n_u}$$

Where $\mathbf{u}$ is

$$\mathbf{u} = [\mathbf{u}_{s_1}^T \ \mathbf{u}_{s_2}^T \dots \mathbf{u}_{s_{n_s}}^T]^T \in \mathcal{R}^{n_u}.$$

and $\mathbf{u}_{s_i}$ is

$$\mathbf{u}_{s_i} = [u_{s_i \to s_1} \dots u_{s_i \to s_{i-1}} u_{s_i \to s_i} u_{s_i \to s_{i+1}} \dots u_{s_i \to s_{n_s}}]^T. \tag{2}$$

Notice that Eq. 2 is different from the definition in the paper. That is because $u_{s_i \to s_i}$ is added to the $\mathbf{u}_{s_i}$ vector. *However, it is always multiplied by zero.* It is added in order to make the implementation easier later on. Therefore, $n_u = n_s^2$ in this implementation.

Dynamics constraints : dynamics over prediction time horizon T_p,

$$\mathbf{X} = \mathbf{T}_u \mathbf{U} + \mathbf{T}_{x_0} \mathbf{x}[0] \tag{3}$$

where $\mathbf{T}_u$ is,

$$\mathbf{T}_u = \begin{bmatrix} \mathbf{B} & \cdots & \mathbf{0} & \\ \mathbf{B} & \mathbf{B} & \cdots & \mathbf{0} \\ \dots & \vdots & \dots & \\ \mathbf{B} & \mathbf{B} & \cdots & \mathbf{B} \end{bmatrix}_{(n_s T_p) \times (n_u T_p)}$$

and,

$$\mathbf{T}_{x_0} \mathbf{x}[0] = \begin{bmatrix} \mathbf{x}[0] \\ \vdots \\ \mathbf{x}[0] \end{bmatrix}_{n_s T_p \times 1}$$

Assuming that the optimization vector is,

$$\hat{\mathbf{X}} = \begin{bmatrix} \mathbf{X} \\ \mathbf{U} \end{bmatrix}_{(n_s + n_u) T_p \times 1} \tag{4}$$

Eq 3 can be written as,

$$\begin{bmatrix}\mathbf{I}_{n_sT_p} & -\mathbf{T}_u\end{bmatrix}\hat{\mathbf{X}} = \mathbf{T}_{x_0}\mathbf{x}[0] \tag{5}$$

It can be converted to inequalities as follows,

$$\begin{bmatrix}\mathbf{I}_{n_sT_p} & -\mathbf{T}_u \\ -\mathbf{I}_{n_sT_p} & \mathbf{T}_u\end{bmatrix}_{2n_sT_p\times(n_s+n_u)T_p}\hat{\mathbf{X}} \leq \begin{bmatrix}\mathbf{T}_{x_0}\mathbf{x}[0] \\ -\mathbf{T}_{x_0}\mathbf{x}[0]\end{bmatrix}_{2n_sT_p\times 1} \tag{6}$$

Which can be re-written as

$$\mathbf{A}_{\text{dynamics}}\hat{\mathbf{X}} \leq \mathbf{b}_{\text{dynamics}}$$

3 Flow Constraints

Flow constraints over prediction time horizon are described by,

$$\mathbf{T}_{u,c}\mathbf{U} \leq \mathbf{T}_{x_0,c}\mathbf{x}[0] \tag{7}$$

where,

$$\mathbf{T}_{u,c} = \begin{bmatrix}\mathbf{B}_{\text{out}} & \mathbf{0} & \cdots & \cdots & \mathbf{0} \\ -\mathbf{B} & \mathbf{B}_{\text{out}} & \mathbf{0} & \cdots & \mathbf{0} \\ -\mathbf{B} & -\mathbf{B} & \mathbf{B}_{\text{out}} & \cdots & \mathbf{0} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ -\mathbf{B} & -\mathbf{B} & -\mathbf{B} & -\mathbf{B} & \mathbf{B}_{\text{out}}\end{bmatrix}$$

$\mathbf{T}_{u,c} \in \mathcal{R}^{n_sT_p\times n_uT_p}$, and

$$\mathbf{T}_{x_0,c}\mathbf{x}[0] = \begin{bmatrix}\mathbf{x}[0] \\ \vdots \\ \mathbf{x}[0]\end{bmatrix}_{n_sT_p\times 1}$$

Using the same optimization vector in (4), constraints (7) can be written as,

$$\begin{bmatrix}\mathbf{0}_{n_sT_p\times n_sT_p} & \mathbf{T}_{u,c}\end{bmatrix}_{n_sT_p\times(n_s+n_u)T_p}\hat{\mathbf{X}} \leq \mathbf{T}_{x_0,c}\mathbf{x}[0] \tag{8}$$

which can be written as,

$$\mathbf{A}_{\text{flow}}\hat{\mathbf{X}} \leq \mathbf{b}_{\text{flow}}$$

4 Boundary Constraints

Constraints on optimization vector $\hat{\mathbf{X}}$ are,

$$\mathbf{0} \leq \hat{\mathbf{X}} \leq \mathbf{1} \tag{9}$$

or,

$$\hat{\mathbf{X}} \leq \mathbf{1}_{(n_s+n_u)T_p}$$
$$-\hat{\mathbf{X}} \leq \mathbf{0}_{(n_s+n_u)T_p}$$

which can be written as,

$$\begin{bmatrix} \mathbf{I} \\ -\mathbf{I} \end{bmatrix}_{2(n_s+n_u)T_p \times (n_s+n_u)T_p} \hat{\mathbf{X}} \leq \begin{bmatrix} \mathbf{1} \\ \mathbf{0} \end{bmatrix} \tag{10}$$

which can be written as

$$\mathbf{A}_{\text{boundary}} \hat{\mathbf{X}} \leq \mathbf{b}_{\text{boundary}}$$

5 Compact Form of Constraints

Using inequality (6), (8), and (10), the LP problem can be written in compact from as,

$$\min_{\hat{\mathbf{X}}} \mathbf{C}^T \hat{\mathbf{X}}$$

s.t.

$$\begin{bmatrix} \mathbf{I}_{n_s T_p} & -\mathbf{T}_u \\ -\mathbf{I}_{n_s T_p} & \mathbf{T}_u \\ \mathbf{0}_{n_s T_p \times n_s T_p} & \mathbf{T}_{u,c} \\ \mathbf{I}_{(n_s+n_u)T_p} & \\ -\mathbf{I}_{(n_s+n_u)T_p} & \end{bmatrix}_{(5n_s+2n_u)Tp \times (n_s+n_u)T_p} \hat{\mathbf{X}} \leq \begin{bmatrix} \mathbf{T}_{x_0}\mathbf{x}[0] \\ -\mathbf{T}_{x_0}\mathbf{x}[0] \\ \mathbf{T}_{x_0,c}\mathbf{x}[0] \\ \mathbf{1}_{(n_s+n_u)T_p} \\ \mathbf{0}_{(n_s+n_u)T_p} \end{bmatrix}_{(5n_s+2n_u)Tp \times 1} \tag{11}$$

which can be written as,

$$\begin{bmatrix} \mathbf{A}_{\text{dynamics}} \\ \mathbf{A}_{\text{flow}} \\ \mathbf{A}_{\text{boundary}} \end{bmatrix} \hat{\mathbf{X}} \leq \begin{bmatrix} \mathbf{b}_{\text{dynamics}} \\ \mathbf{b}_{\text{flow}} \\ \mathbf{b}_{\text{boundary}} \end{bmatrix} \tag{12}$$

6 Notes fot GLPK Use

If `GLPK` is used, boundary constraints in (9) are not considered in constraints matrix (the rows of A). It is entered as separate types of constraints in the defined `glpk` problem.

7 Estimation of Enemy State, $\mathbf{X}^{\text{e}}$

Enemy state trajectory $\mathbf{X}^{\text{e}}$ is used in the LP objective vector $\mathbf{C} = \beta \mathbf{X}^{\text{ref}} + \alpha \mathbf{X}^{\text{e}}$ as follows,

$$\mathcal{J}_{T_p}(\hat{\mathbf{X}}) = min_{\hat{\mathbf{X}}}\, \mathbf{C}^T \hat{\mathbf{X}} \tag{13}$$

In this implementation, the enemy state is computes based on the following linear model,

$$\mathbf{x}^{\mathrm{e}}[t+1] = \mathbf{x}^{\mathrm{e}}[t] + \mathbf{B}\mathbf{u}^{\mathrm{e}}[t] \tag{14}$$

where,

$$\mathbf{u}^{\mathrm{e}}[t] = \mathbf{G}^{\mathrm{e}}\mathbf{x}^{\mathrm{e}}[t] \tag{15}$$

Hence,

$$\mathbf{x}^{\mathrm{e}}[t+1] = (\mathbf{I} + \mathbf{B}\mathbf{G}^{\mathrm{e}})^{t+1}\mathbf{x}_0^{\mathrm{e}} \tag{16}$$

based on (16), the enemy state trajectory can be written as,

$$\mathbf{X}^{\mathrm{e}} = \begin{bmatrix} (\mathbf{I} + \mathbf{B}\mathbf{G}^{\mathrm{e}}) \\ (\mathbf{I} + \mathbf{B}\mathbf{G}^{\mathrm{e}})^2 \\ \vdots \\ (\mathbf{I} + \mathbf{B}\mathbf{G}^{\mathrm{e}})^{T_p} \end{bmatrix}_{n_s T_p \times n_s} \mathbf{x}_0^{\mathrm{e}} \tag{17}$$

or,

$$\mathbf{X}^{\mathrm{e}} = \mathbf{T}_{\mathbf{G}}\, \mathbf{x}_0^{\mathrm{e}}$$

Calculation of $\mathbf{G}^{\mathbf{e}}$:

- This matrix contains the probabilities that an agent moves from one sector s_i to another sector $s_j \in N_{s_i}$.
- $\mathbf{G} \in \mathcal{R}^{nu \times n_s}$
- $g_{s_i \to s_j}$ is the probability that an agent in sector s_i will move to a neighbor sector s_j.
- $u_{s_i \to s_j} = g_{s_i \to s_j} \cdot x_{s_i}$
- Therefore, each row in $\mathbf{G}$ contains only one non-zero element, $g_{s_i \to s_j}$.
- The non-zero elements in matrix $\mathbf{G}$ can be accessed in a `Eigen` matrix as $g_{s_i \to s_j} = \mathbf{G}(s_i n_s - n_s + (s_j - 1), s_i)$. This assumes that the matrix indexing starts from zero.