最適制御問題の固有構造を利用したQP問題の最適化計算（本書[1]付録A.4補足稿）

2024年6月12日

線形MPCの場合や、非線形MPCで逐次二次計画法を用いる場合にはQP問題を扱います。このQP問題には最適制御に固有の構造があり、この固有の構造を利用して最適化計算をさらに高速化することが可能です。例えば、hpipm [2] は、この固有の構造を利用した内点法ベースのQPソルバーです。

本補足稿では、QP問題の内点法ベースの最適化計算において最適制御に固有の構造を利用する方法の概要を紹介します。なお、有効制約法や交互方向乗数法においても最適制御に固有の構造を利用することは可能であり、興味のある読者は [3, 4] などの手法を参照してください。

準備

本補足稿で参照する目的で、本書 [1] 中で説明した式を以下に再掲します。

本書 [1] 式 (6.15)：線形MPCにおいて扱う離散時間最適制御問題の定式

$$\begin{aligned}
\underset{\mathbf{X},\mathbf{U}}{\text{minimize}} \quad & \mathbf{x}_K^T Q_K \mathbf{x}_K + 2q_K^T \mathbf{x}_K \\
& + \sum_{k=0}^{K-1} \left\{ \begin{bmatrix} \mathbf{x}_k \\ \mathbf{u}_k \end{bmatrix}^T \begin{bmatrix} Q_k & S_k \\ S_k^T & R_k \end{bmatrix} \begin{bmatrix} \mathbf{x}_k \\ \mathbf{u}_k \end{bmatrix} + 2 \begin{bmatrix} q_k \\ r_k \end{bmatrix}^T \begin{bmatrix} \mathbf{x}_k \\ \mathbf{u}_k \end{bmatrix} \right\} \\
\text{subject to} \quad & \begin{cases} \mathbf{x}_0 = \bar{\mathbf{x}}_{init} \\ \mathbf{x}_{k+1} = A_k \mathbf{x}_k + B_k \mathbf{u}_k + b_k \\ G_k^x \mathbf{x}_k + G_k^u \mathbf{u}_k \le h_k \\ G_K^x \mathbf{x}_K \le h_K \end{cases}
\end{aligned} \tag{1}$$

本書 [1] 式 (A.9)：QP問題の内点法において扱うKKT行列線型方程式

$$\begin{bmatrix} Q & -C^T & O \\ -C & O & I \\ O & Z^\ell & \Pi^\ell \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta \pi \\ \Delta z \end{bmatrix} = \begin{bmatrix} -\gamma_x^\ell \\ -\gamma_\pi^\ell \\ -Z^\ell \Pi^\ell e + \mu^\ell e \end{bmatrix} \tag{2}$$

1 最適制御における不等式制約付き QP 問題の KKT 行列線形方程式計算の変換

不等式制約がある線形最適制御問題は QP 問題として表現できます（本書 [1]6.3.2 項）。これを内点法で解く際に必要な KKT 行列の線形方程式 (2) を解く部分は、LQ 制御問題に変換することが可能 [5] であり、LQ 制御の効率的解法を使用して高速に解くことができます（後に本補足稿 2 節で説明します）。本節では、そのように高速に解くための前段階として、KKT 行列の線形方程式 (2) を LQ 制御問題に変換することを数式を追いながら確認します。

式 (1) の QP 問題における不等式制約をスラック変数 z_k を用いて表現すると、次式となります。

$$\min_{\mathbf{X},\mathbf{U}} \quad \mathbf{x}_K^T Q_K \mathbf{x}_K + 2q_K^T\mathbf{x}_K + \sum_{k=0}^{K-1}\begin{bmatrix}\mathbf{x}_k\\ \mathbf{u}_k\\ 1\end{bmatrix}^T\begin{bmatrix}Q_k & S_k & q_k\\ S_k^T & R_k & r_k\\ q_k^T & r_k^T & 0\end{bmatrix}\begin{bmatrix}\mathbf{x}_k\\ \mathbf{u}_k\\ 1\end{bmatrix}$$

$$\text{subject to} \quad \begin{cases}\mathbf{x}_0 = \bar{\mathbf{x}}_{init}\\ \mathbf{x}_{k+1} = A_k\mathbf{x}_k + B_k\mathbf{u}_k + b_k\\ G_k^x\mathbf{x}_k + G_k^u\mathbf{u}_k + z_k = h_k\\ G_K^x\mathbf{x}_K + z_K = h_K\\ z_k \geq 0\end{cases}$$

この QP 問題の解が満たすべき最適性の必要条件は、次の KKT 条件です。

$$\begin{aligned}\mathbf{x}_0 &= \bar{\mathbf{x}}_{init}\\ Q_k\mathbf{x}_k + S_k\mathbf{u}_k + A_k^T\lambda_k - \lambda_{k-1} + (G_k^x)^T\pi_k &= -q_k\\ S_k^T\mathbf{x}_k + R_k\mathbf{u}_k + B_k^T\lambda_k + (G_k^u)^T\pi_k &= -r_k\\ A_k\mathbf{x}_k + B_k\mathbf{u}_k - \mathbf{x}_{k+1} &= -b_k\\ G_k^x\mathbf{x}_k + G_k^u\mathbf{u}_k + z_k &= h_k\\ G_K^x\mathbf{x}_k + z_K &= h_K\\ Z_k\Pi_k e &= 0\\ \pi_k &\geq 0\\ z_k &\geq 0\end{aligned}$$

ここで、初期状態等式制約の未定乗数を λ_{-1}、k ステップ目の状態方程式モデルの等式制約に対する未定乗数を λ_k、k ステップ目の不等式制約のスラック変数を用いた等式表現に対する未定乗数を π_k としています。本書 [1] 付録 A.2.1 で述べたように、内点法ではこの KKT 条件を現在の推定解周りで線形化して修正し、得られた KKT 行列の線形方程式を解くことにより更新方向を計算します。

不等式制約には $k=0,\cdots,K-1$ ステップ目の不等式制約 $G_k^x\mathbf{x}_k+G_k^u\mathbf{u}_k<h_k$ と $k=K$ ステップ目の不等式制約 $G_K^x\mathbf{x}_K<h_K$ の 2 種類が存在します。まず、$k=0,\cdots,K-1$ ステップ目の不等式制約を扱う方法を説明します。KKT 行列の線形方程式の中で k ステップ目の変数 $(\mathbf{x}_k,\mathbf{u}_k,\pi_k,z_k,\lambda_k)$ に関する勾配についての最適性の必要条件に対応する行は、関係する成分のみを取り出して表記すると次のように書くことができます。

$$\begin{bmatrix} -I & Q_k & S_k & (G_k^x)^T & O & A_k^T & O \\ O & S_k^T & R_k & (G_k^u)^T & O & B_k^T & O \\ O & G_k^x & G_k^u & O & I & O & O \\ O & O & O & Z_k & \Pi_k & O & O \\ O & A_k & B_k & O & O & O & -I \end{bmatrix} \begin{bmatrix} \Delta\lambda_{k-1} \\ \Delta\mathbf{x}_k \\ \Delta\mathbf{u}_k \\ \Delta\pi_k \\ \Delta z_k \\ \Delta\lambda_k \\ \Delta\mathbf{x}_{k+1} \end{bmatrix} = \begin{bmatrix} -\gamma_{x_k} \\ -\gamma_{u_k} \\ -\gamma_{\pi_k} \\ -\gamma_{z_k} \\ -\gamma_{\lambda_k} \end{bmatrix} \tag{3}$$

ここで、式 (3) の右辺は現在の推定解の残差に内点法による修正を加えたものであり、式 (2) の右辺と同様です。式 (3) の 3 行目と 4 行目より、$\Delta\pi_k$ と Δz_k は次式により得られます。

$$\begin{aligned} \Delta z_k &= -G_K^x\Delta x_k - G_k^u\Delta u_k - \gamma_{\pi_k} \\ \Delta\pi_k &= Z_k^{-1}(-\Pi_k\Delta z_k - \gamma_{z_k}) \\ &= Z_k^{-1}\Pi_k(G_k^x\Delta x_k + G_k^u\Delta u_k) + Z_k^{-1}(\Pi_k\gamma_{\pi_k} - \gamma_{z_k}) \end{aligned}$$

これらを式 (3) の 1 行目に代入すると、次式が得られます。

$$\begin{bmatrix} -I & Q'_k & S'_k & A_k^T & O \\ O & S_k'^T & R'_k & B_k^T & O \\ O & A_k & B_k & O & -I \end{bmatrix} \begin{bmatrix} \Delta\lambda_{k-1} \\ \Delta\mathbf{x}_k \\ \Delta\mathbf{u}_k \\ \Delta\lambda_k \\ \Delta\mathbf{x}_{k+1} \end{bmatrix} = \begin{bmatrix} -\gamma'_{x_k} \\ -\gamma'_{u_k} \\ -\gamma_{\lambda_k} \end{bmatrix} \tag{4}$$

ここで、以下のように省略表記しました。

$$\begin{aligned} Q'_k &= Q_k + (G_k^x)^T Z_k^{-1}\Pi_k G_k^x \\ S'_k &= S_k + (G_k^x)^T Z_k^{-1}\Pi_k G_k^u \\ R'_k &= R_k + (G_k^u)^T Z_k^{-1}\Pi_k G_k^u \\ \gamma'_{x_k} &= \gamma_{x_k} + (G_k^x)^T Z_k^{-1}(\Pi_k\gamma_{\pi_k} - \gamma_{z_k}) \\ \gamma'_{u_k} &= \gamma_{u_k} + (G_k^u)^T Z_k^{-1}(\Pi_k\gamma_{\pi_k} - \gamma_{z_k}) \end{aligned}$$

次に、$k=K$ ステップ目の終端状態の不等式制約を扱う方法を説明します。変数 $\mathbf{x}_K,\pi_K,z_K$ に関する勾配の最適性の必要条件に対応する行を式 (3) と同

様に表記すると、次式となります。

$$\begin{bmatrix} -I & Q_K & (G_K^x)^T & O \\ O & G_K^x & O & I \\ O & O & Z_K & \Pi_K \end{bmatrix} \begin{bmatrix} \Delta\lambda_{K-1} \\ \Delta\mathbf{x}_K \\ \Delta\pi_K \\ \Delta z_K \end{bmatrix} = \begin{bmatrix} -\gamma_{x_K} \\ -\gamma_{\pi_K} \\ -\gamma_{z_K} \end{bmatrix} \tag{5}$$

式 (5) の 2 行目と 3 行目より、$\Delta\pi_K$ と Δz_K はそれぞれ次式となります。

$$\begin{aligned} \Delta z_K &= -G_K^x \Delta x_K - \gamma_{\pi_K} \\ \Delta\pi_K &= Z_K^{-1}(-\Pi_K \Delta z_K - \gamma_{z_K}) = Z_K^{-1}\Pi_K G_K^x \Delta x_K + Z_K^{-1}(\Pi_K \gamma_{\pi_K} - \gamma_{z_K}) \end{aligned}$$

これらを式 (5) の 1 行目に代入すると、次式が得られます。

$$\begin{bmatrix} -I & Q'_K \end{bmatrix} \begin{bmatrix} \Delta\lambda_{K-1} \\ \Delta\mathbf{x}_K \end{bmatrix} = -\gamma'_{x_K} \tag{6}$$

ここで、

$$\begin{aligned} Q'_K &= Q_K + (G_K^x)^T Z_K^{-1} \Pi_K G_K^x \\ \gamma'_{x_K} &= \gamma_{x_K} + (G_K^x)^T Z_K^{-1}(\Pi_K \gamma_{\pi_K} - \gamma_{z_K}) \end{aligned}$$

そのため、最適制御において不等式制約条件を含む QP 問題を解く際の内点法の中で KKT 行列の線形方程式を解いて更新方向を計算する部分は、式 (4) と式 (6) を満たす $(\Delta x_k, \Delta u_k, \Delta\lambda_k)$ を計算することと等価です。

式 (4) と式 (6) は、次の QP 問題の解 $(\Delta x_k, \Delta u_k, \Delta\lambda_k)$ が満たすべき最適性の必要条件として見ることもできます。

$$\begin{aligned} \min_{\Delta\mathbf{X}, \Delta\mathbf{U}} \quad & \Delta\mathbf{x}_K^T Q'_K \Delta\mathbf{x}_K + 2\gamma'^T_{x_K} \Delta\mathbf{x}_K \\ & + \sum_{k=0}^{K-1} \begin{bmatrix} \Delta\mathbf{x}_k \\ \Delta\mathbf{u}_k \\ 1 \end{bmatrix}^T \begin{bmatrix} Q'_k & S'_k & \gamma'_{x_k} \\ S'^T_k & R'_k & \gamma'_{u_k} \\ \gamma'^T_{x_k} & \gamma'^T_{u_k} & 0 \end{bmatrix} \begin{bmatrix} \Delta\mathbf{x}_k \\ \Delta\mathbf{u}_k \\ 1 \end{bmatrix} \\ \text{subject to} \quad & \begin{cases} \Delta\mathbf{x}_0 = \bar{\mathbf{x}}_{init} - \tilde{\mathbf{x}}_0 \\ \Delta\mathbf{x}_{k+1} = A_k \Delta\mathbf{x}_k + B_k \Delta\mathbf{u}_k + \gamma_{\lambda_k} \end{cases} \end{aligned}$$

ここで、$\tilde{\mathbf{x}}_0$ は現在の $\mathbf{x}_0$ の推定解です。この QP 問題は、評価関数に交差項があり状態方程式にバイアス項がある場合の有限ホライズンの離散時間 LQ 制御問題として見ることができます。

以上により、最適制御において不等式制約条件を含む QP 問題を解く際の内点法の中の KKT 行列の線形方程式を解いて更新方向を計算する部分は、有限ホライズンの離散時間 LQ 制御問題の解を計算することに変換することができることを確認しました。

2 リッカチ再帰式を利用したKKT 行列の線形方程式の解法

KKT 行列の線形方程式を解く問題は有限ホライズンの離散時間 LQ 制御問題に変換することができる（本補足稿 1 節）ため、この問題は線形制御の効率的解法であるリッカチ再帰式を利用して解くことができます [5, 6, 7]。以下では、このことを数式を追いながら確認します。

本節では、次の有限ホライズンの離散時間 LQ 制御問題、すなわち、評価関数が二次形式で制約条件が初期状態と線形状態方程式モデルのみである QP 問題を考えます。

$$
\begin{aligned}
\min_{\mathbf{X},\mathbf{U}} \quad & \mathbf{x}_K^T Q_K \mathbf{x}_K + 2q_K^T \mathbf{x}_K + \sum_{k=0}^{K-1} \begin{bmatrix} \mathbf{x}_k \\ \mathbf{u}_k \\ 1 \end{bmatrix}^T \begin{bmatrix} Q_k & S_k & q_k \\ S_k^T & R_k & r_k \\ q_k^T & r_k^T & 0 \end{bmatrix} \begin{bmatrix} \mathbf{x}_k \\ \mathbf{u}_k \\ 1 \end{bmatrix} \\
\text{subject to} \quad & \begin{cases} \mathbf{x}_0 = \bar{\mathbf{x}}_{init} \\ \mathbf{x}_{k+1} = A_k \mathbf{x}_k + B_k \mathbf{u}_k + b_k \end{cases}
\end{aligned}
$$

この QP 問題の最適性の必要条件は、対角に帯状にブロック行列が並んだ KKT 行列の線形方程式となります。

$$
\begin{bmatrix}
\ddots & & & & & & \\
 & & -I & & & & \\
 & -I & Q_{K-1} & S_{K-1} & A_{K-1}^T & & \\
 & & S_{K-1}^T & R_{K-1} & B_{K-1}^T & & \\
 & & A_{K-1} & B_{K-1} & & -I & \\
 & & & & -I & Q_K
\end{bmatrix}
\begin{bmatrix} \vdots \\ \lambda_{K-2} \\ \mathbf{x}_{K-1} \\ \mathbf{u}_{K-1} \\ \lambda_{K-1} \\ \mathbf{x}_K \end{bmatrix}
=
\begin{bmatrix} \vdots \\ -b_{K-1} \\ -q_{K-1} \\ -r_{K-1} \\ -b_{K-1} \\ -q_K \end{bmatrix}
\tag{7}
$$

Q_K を乗じた下から 2 番目の行に一番下の行を加えると、次式が得られます。

$$
\begin{bmatrix}
\ddots & & & & \\
 & & -I & & \\
 & -I & Q_{K-1} & S_{K-1} & A_{K-1}^T \\
 & & S_{K-1}^T & R_{K-1} & B_{K-1}^T \\
 & & Q_K A_{K-1} & Q_K B_{K-1} & -I
\end{bmatrix}
\begin{bmatrix} \vdots \\ \lambda_{K-2} \\ \mathbf{x}_{K-1} \\ \mathbf{u}_{K-1} \\ \lambda_{K-1} \end{bmatrix}
=
\begin{bmatrix} \vdots \\ -b_{K-1} \\ -q_{K-1} \\ -r_{K-1} \\ -(Q_K b_{K-1} + q_K) \end{bmatrix}
$$

下から 3 番目の行に A_{K-1}^T を乗じた一番下の行を加え、下から 2 番目の行に

B_{K-1}^T を乗じた一番下の行を加えると、次式が得られます。

$$\begin{bmatrix} \ddots & & & \\ & & -I & \\ & -I & Q_{K-1} + A_{K-1}^T Q_K A_{K-1} & S_{K-1} + A_{K-1}^T Q_K B_{K-1} \\ & & S_{K-1}^T + B_{K-1}^T Q_K A_{K-1} & R_{K-1} + B_{K-1}^T Q_K B_{K-1} \end{bmatrix} \begin{bmatrix} \vdots \\ \lambda_{K-2} \\ \mathbf{x}_{K-1} \\ \mathbf{u}_{K-1} \end{bmatrix}$$
$$= \begin{bmatrix} \vdots \\ -b_{K-1} \\ -(A_{K-1}^T(Q_K b_{K-1} + q_K) + q_{K-1}) \\ -(B_{K-1}^T(Q_K b_{K-1} + q_K) + r_{K-1}) \end{bmatrix}$$

下から 2 番目の行に $(S_{K-1} + A_{K-1}^T Q_K B_{K-1})(R_{K-1} + B_{K-1}^T Q_K B_{K-1})^{-1}$ を乗じた一番下の行を加えると、次式が得られます。

$$\begin{bmatrix} \ddots & & \\ & & -I \\ & -I & P_{K-1} \end{bmatrix} \begin{bmatrix} \vdots \\ \lambda_{K-2} \\ \mathbf{x}_{K-1} \end{bmatrix} = \begin{bmatrix} \vdots \\ -b_{K-1} \\ -p_{K-1} \end{bmatrix} \tag{8}$$

ここで、P_K と p_k は次式

$$\begin{aligned} P_k =& Q_k + A_k^T P_{k+1} A_k \\ & - (S_k + A_k^T P_{k+1} B_k)(R_k + B_k^T P_{k+1} B_k)^{-1}(S_k^T + B_k^T P_{k+1} A_k) \end{aligned} \tag{9}$$

$$\begin{aligned} p_k =& q_k + A_k^T(P_{k+1} b_k + p_{k+1}) \\ & - (S_k + A_k^T P_{k+1} B_k)(R_k + B_k^T P_{k+1} B_k)^{-1}(B_k^T(P_{k+1} b_k + p_{k+1}) + r_k) \end{aligned} \tag{10}$$

に $k = K-1$、$P_K = Q_K$、$p_K = q_K$ を代入して計算した値です。なお、この式はリッカチ再帰式と呼ばれます。

式 (7) から式 (8) を得る操作は $k = 0$ まで再帰的に繰り返し適用することが可能です。その際、P_k と p_k は、式 (9) と式 (10) により得ることができます。また、$\mathbf{u}_k$ と λ_k は、次式で得ることができます。

$$\begin{aligned} \mathbf{u}_k =& - (R_k + B_k^T P_{k+1} B_k)^{-1}(S_k^T + B_k^T P_{k+1} A_k)\mathbf{x}_k \\ & - (B_k^T(P_{k+1} b_k + p_{k+1}) + r_k) \end{aligned} \tag{11}$$

$$\lambda_k = P_{k+1}\mathbf{x}_{k+1} + p_{k+1} \tag{12}$$

$k = 0$ まで再帰的に繰り返し適用した後、最終的には次式が得られます。

$$\begin{bmatrix} & -I \\ -I & P_0 \end{bmatrix} \begin{bmatrix} \lambda_{-1} \\ \mathbf{x}_0 \end{bmatrix} = \begin{bmatrix} -\bar{\mathbf{x}}_{init} \\ -p_0 \end{bmatrix}$$

したがって、式 (7) の KKT 行列の線形方程式は、大まかには以下の流れで解くことができます。

- (P_k, p_k) を、式 (9) と式 (10) を用いて $k = K, K-1, \cdots, 0$ の順番に計算する。

- $(\mathbf{u}_k, \mathbf{x}_{k+1}, \lambda_{k+1})$ を、状態方程式モデルと式 (11) と式 (12) を用いて $k = 0, 1, \cdots, K-1$ の順番に計算する。

リッカチ再帰式を用いて KKT 行列の線形方程式を解く方法の主要な計算コストは、式 (9)-式 (11) の中の逆行列計算と行列同士の積算であり、状態次元数 n_x あるいは制御次元数 n_u に関して 3 乗のオーダーです。この計算をホライズン長 K の回数分だけ再帰的に実行するため、全体としては K に関しては線形オーダーの計算量です。一方で、式 (7) の KKT 行列の線形方程式を汎用的なコレスキー分解を用いて解く場合には、KKT 行列のサイズが $K(2n_x + n_u) \times (2n_x + n_u)$ であるため計算量は $(K(2n_x + n_u))^3$ のオーダーであり、つまり、K に関して 3 乗のオーダーです。そのため、最適制御問題に固有の構造を利用したリッカチ再帰式を用いて KKT 行列の線形方程式を解く方が効率的に計算を行うことができます。

本節では、リカッチ再帰式を用いて KKT 行列の線形方程式を解くためのアイデアの概要を紹介しました。より実践的なアルゴリズムの実装方法やより詳細な計算量の解析については、[7] を参照してください。

3　partial condensing（状態変数の部分的な消去）

本書 [1] 6.3.2 項では、最適制御における QP 問題の状態変数を消去しない場合と全て消去する場合を考えました。状態変数の全てではなく一部を変数消去するという中間的なアプローチも存在し、condensing を部分的に行うという意味で **partial condensing** と呼ばれます [8, 7]。最適制御における QP 問題を解くための計算コストはホライズン K・状態次元数 n_x・制御次元数 n_u により決まる（本補足稿 2 節参照）ため、partial condensing を用いて状態変数を適当な割合で部分的に消去することにより QP 問題を解くための計算コストを減らすことができます。partial condensing は実践的 MPC ツールである acados [9]（本書 [1] 付録 B で紹介します）においても利用可能なオプションとして用意されていて、より高速な MPC を実装する際には重要な考え方の一つです。

以下では、partial condensing の具体例として、元々の問題のホライズンが 3 の倍数であり $K = 3\tilde{K}$ とした場合に状態 $\mathbf{x}_{k+1}, \mathbf{x}_{k+2}$ を変数消去する例を説明します。$\mathbf{x}_{k+1}, \mathbf{x}_{k+2}$ は、次のように書くことができます。

$$
\begin{aligned}
\mathbf{x}_{k+1} &= A_k \mathbf{x}_k + B_k \mathbf{u}_k + b_k \\
\mathbf{x}_{k+2} &= A_{k+1} \mathbf{x}_{k+1} + B_{k+1} \mathbf{u}_{k+1} + b_{k+1} \\
&= A_{k+1} A_k \mathbf{x}_k + \begin{bmatrix} A_{k+1} B_k & B_{k+1} \end{bmatrix} \begin{bmatrix} \mathbf{u}_k \\ \mathbf{u}_{k+1} \end{bmatrix} + (A_{k+1} b_k + b_{k+1}) \\
\mathbf{x}_{k+3} &= A_{k+2} \mathbf{x}_{k+2} + B_{k+2} \mathbf{u}_{k+2} + b_{k+2} \\
&= \tilde{A}_k \mathbf{x}_k + \tilde{B}_k \begin{bmatrix} \mathbf{u}_k \\ \mathbf{u}_{k+1} \\ \mathbf{u}_{k+2} \end{bmatrix} + \tilde{b}_k
\end{aligned}
$$

ここで、$\tilde{A}_k, \tilde{B}_k, \tilde{b}_k$ は、次のように書くことができます。

$$
\begin{aligned}
\tilde{A}_k &= A_{k+2} A_{k+1} A_k \\
\tilde{B}_k &= \begin{bmatrix} A_{k+2} A_{k+1} B_k & A_{k+2} B_{k+1} & B_{k+2} \end{bmatrix} \\
\tilde{b}_k &= A_{k+2}(A_{k+1}(A_k + b_k) + b_{k+1}) + b_{k+2}
\end{aligned}
$$

新たに変数 $\tilde{\mathbf{x}}_k = \mathbf{x}_{3k}$、$\tilde{\mathbf{u}}_k = \begin{bmatrix} \mathbf{u}_k \\ \mathbf{u}_{k+1} \\ \mathbf{u}_{k+2} \end{bmatrix}$ と定義すると、次のように書くことができます。

$$
\tilde{\mathbf{x}}_{k+1} = \tilde{A}_k \tilde{\mathbf{x}}_k + \tilde{B}_k \tilde{\mathbf{u}}_k + \tilde{b}_k \tag{13}
$$

これは、$\tilde{\mathbf{x}}_k$ と $\tilde{\mathbf{u}}_k$ を状態変数と制御入力とするシステムの状態方程式モデルとして見ることができます。

新たに定義した変数について、次のように書くことができます。

$$\begin{bmatrix}\mathbf{x}_k\\ \mathbf{u}_k\\ 1\end{bmatrix} = \begin{bmatrix} I & O & O & O & 0\\ O & I & O & O & 0\\ O & O & O & O & 1\end{bmatrix}\begin{bmatrix}\mathbf{x}_k\\ \mathbf{u}_k\\ \mathbf{u}_{k+1}\\ \mathbf{u}_{k+2}\\ 1\end{bmatrix} = T_{k,0}\begin{bmatrix}\tilde{\mathbf{x}}_k\\ \tilde{\mathbf{u}}_k\\ 1\end{bmatrix}$$

$$\begin{bmatrix}\mathbf{x}_{k+1}\\ \mathbf{u}_{k+1}\\ 1\end{bmatrix} = \begin{bmatrix} A_k & B_k & O & O & b_k\\ O & O & I & O & 0\\ O & O & O & O & 1\end{bmatrix}\begin{bmatrix}\mathbf{x}_k\\ \mathbf{u}_k\\ \mathbf{u}_{k+1}\\ \mathbf{u}_{k+2}\\ 1\end{bmatrix} = T_{k,1}\begin{bmatrix}\tilde{\mathbf{x}}_k\\ \tilde{\mathbf{u}}_k\\ 1\end{bmatrix}$$

$$\begin{aligned}\begin{bmatrix}\mathbf{x}_{k+2}\\ \mathbf{u}_{k+2}\\ 1\end{bmatrix} &= \begin{bmatrix} A_{k+2}A_k & A_{k+1}B_k & B_{k+1} & O & A_{k+1}b_k + b_{k+1}\\ O & O & O & I & 0\\ O & O & O & O & 1\end{bmatrix}\begin{bmatrix}\mathbf{x}_k\\ \mathbf{u}_k\\ \mathbf{u}_{k+1}\\ \mathbf{u}_{k+2}\\ 1\end{bmatrix}\\ &= T_{k,2}\begin{bmatrix}\tilde{\mathbf{x}}_k\\ \tilde{\mathbf{u}}_k\\ 1\end{bmatrix}\end{aligned}$$

ステージコストの係数行列を次式で定義します。

$$\begin{bmatrix}\tilde{Q}_{k+i} & \tilde{S}_{k+i} & \tilde{q}_{k+i}\\ \tilde{S}^T_{k+i} & \tilde{R}_{k+i} & \tilde{r}_{k+i}\\ \tilde{q}^T_{k+i} & \tilde{r}^T_{k+i} & 0\end{bmatrix} = \sum_{i=0}^{2} T^T_{k,i}\begin{bmatrix}Q_{k+i} & S_{k+i} & q_{k+i}\\ S^T_{k+i} & R_{k+i} & r_{k+i}\\ q^T_{k+i} & r^T_{k+i} & 0\end{bmatrix} T_{k,i}$$

式 (1) の元々の問題の評価関数は、次のように書き変えることができます。

$$\begin{aligned}&\mathbf{x}_K^T Q_k \mathbf{x}_K + 2q_K^T\mathbf{x}_K + \sum_{k=0}^{K-1}\begin{bmatrix}\mathbf{x}_k\\ \mathbf{u}_k\\ 1\end{bmatrix}^T\begin{bmatrix}Q_k & S_k & q_k\\ S_k^T & R_k & r_k\\ q_k^T & r_k^T & 0\end{bmatrix}\begin{bmatrix}\mathbf{x}_k\\ \mathbf{u}_k\\ 1\end{bmatrix}\\ &= \mathbf{x}_{\tilde{K}}^T Q_K \mathbf{x}_{\tilde{K}} + 2q_K^T\tilde{\mathbf{x}}_{\tilde{K}} + \sum_{k=0}^{\tilde{K}-1}\begin{bmatrix}\tilde{\mathbf{x}}_k\\ \tilde{\mathbf{u}}_k\\ 1\end{bmatrix}^T\begin{bmatrix}\tilde{Q}_k & \tilde{S}_k & \tilde{q}_k\\ \tilde{S}_k^T & \tilde{R}_k & \tilde{r}_k\\ \tilde{q}_k^T & \tilde{r}_k^T & 0\end{bmatrix}\begin{bmatrix}\tilde{\mathbf{x}}_k\\ \tilde{\mathbf{u}}_k\\ 1\end{bmatrix} \qquad (14)\end{aligned}$$

これは、$\tilde{\mathbf{x}}_k$ と $\tilde{\mathbf{u}}_k$ を状態変数と制御入力とするシステムの評価関数として見ることができます。

同様にして、制約条件も書き換えることができます。$i = \{0, 1, 2\}$ に対して、式 (1) の元々の問題の不等式

$$G^x_{k+i}\mathbf{x}_{k+i} + G^u_{k+i}\mathbf{u}_{k+i} \le h_{k+i}$$

は次のように書き換えることができます。

$$\begin{bmatrix} G_{k+i}^x & G_{k+i}^u & -h_{k+i} \end{bmatrix} \begin{bmatrix} \mathbf{x}_{k+i} \\ \mathbf{u}_{k+i} \\ 1 \end{bmatrix} = \begin{bmatrix} G_{k+i}^x & G_{k+i}^u & -h_{k+i} \end{bmatrix} T_{k.i} \begin{bmatrix} \tilde{\mathbf{x}}_k \\ \tilde{\mathbf{u}}_k \\ 1 \end{bmatrix} \leq 0$$

係数 $\tilde{G}_k^x, \tilde{G}_k^u, \tilde{h}_k$ を次式で定義します。

$$\begin{bmatrix} \tilde{G}_k^x & \tilde{G}_k^u & -\tilde{h}_k \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} G_k^x & G_k^u & -h_k \end{bmatrix} T_{k,0} \\ \begin{bmatrix} G_{k+1}^x & G_{k+1}^u & -h_{k+1} \end{bmatrix} T_{k,1} \\ \begin{bmatrix} G_{k+2}^x & G_{k+2}^u & -h_{k+2} \end{bmatrix} T_{k,2} \end{bmatrix}$$

$k = 0, \cdots, \tilde{K} - 1$ に対して、

$$\begin{bmatrix} \tilde{G}_k^x & \tilde{G}_k^u & -\tilde{h}_k \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{x}}_k \\ \tilde{\mathbf{u}}_k \\ 1 \end{bmatrix} \leq 0 \tag{15}$$

と書くと、$\tilde{\mathbf{x}}_k$ と $\tilde{\mathbf{u}}_k$ を状態変数と制御入力とするシステムの制約条件として見ることができます。

式 (13)-(15) より、partial condensing により部分的に変数消去をして再定式化をした最適制御問題は、次のように書くことができます。

$$\min_{\tilde{\mathbf{x}}_0, \cdots, \tilde{\mathbf{x}}_{\tilde{K}}, \tilde{\mathbf{u}}_0, \cdots, \tilde{\mathbf{u}}_{\tilde{K}-1}} \quad \tilde{\mathbf{x}}_{\tilde{K}}^T Q_K \tilde{\mathbf{x}}_{\tilde{K}} + 2 q_K^T \mathbf{x}_K + \sum_{k=0}^{\tilde{K}-1} \begin{bmatrix} \tilde{\mathbf{x}}_k \\ \tilde{\mathbf{u}}_k \\ 1 \end{bmatrix}^T \begin{bmatrix} \tilde{Q}_k & \tilde{S}_k & \tilde{q}_k \\ \tilde{S}_k^T & \tilde{R}_k & \tilde{r}_k \\ \tilde{q}_k^T & \tilde{r}_k^T & 0 \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{x}}_k \\ \tilde{\mathbf{u}}_k \\ 1 \end{bmatrix}$$

$$\text{subject to} \quad \begin{cases} \tilde{\mathbf{x}}_0 = \bar{\mathbf{x}}_{init} \\ \tilde{\mathbf{x}}_{k+1} = \tilde{A}_k \tilde{\mathbf{x}}_k + \tilde{B}_k \tilde{\mathbf{u}}_k + \tilde{b}_k \\ \tilde{G}_k^x \tilde{\mathbf{x}}_k + \tilde{G}_k^u \tilde{\mathbf{u}}_k \leq \tilde{h}_k \\ G_K^x \tilde{\mathbf{x}}_{\tilde{K}} \leq h_K \end{cases} \tag{16}$$

再定式化した問題における状態変数の次元は $\tilde{n}_x = n_x$、制御入力の次元は $\tilde{n}_u = 3n_u$、ホライズンは $\tilde{K} = K/3$ です。つまり、partial condensing では、ホライズンを 1/3 倍、制御入力の次元を 3 倍とした等価な最適制御問題を再定式化します。この具体例を一般化して、partial condensing により、ホライズンを $1/m$ 倍、制御入力の次元を m 倍とした等価な最適制御問題を再定式化することができます [8, 7]。ホライズン長を $\tilde{K} = 1$ とした場合は全ての状態変数を消去した場合、$\tilde{K} = K$ とした場合は状態変数を消去しない場合として解釈することができます。また、さらなる一般化として、消去する状態変数を等間隔に選ばないような拡張を考えることもできます。以上のことから、partial condensing は、ホライズンを短くする代わりに制御入力の次元を増やして最適制御問題を再定式化する方法として解釈することができます。

本補足稿 2 節で述べたように、QP 最適化における主要な計算コストである KKT 行列の線型方程式を解く計算量はホライズン K・状態次元数 n_x・制御次元数 n_u により決まるため、partial condensing によってこれらの値を変更してから QP ソルバーを適用することにより、より少ない計算コストで最適化を行うことができます。

最適化ソルバーを適用する前の前処理としての partial condensing の計算コストと比較して、最適化ソルバーを適用して低減化される計算コストの方が大きいため、partial condensing 自体の計算コストを含めてトータルで評価した上でも計算を効率化することができます（詳細は [7] の 13 章数値実験を参照してください）。partial condensing のアルゴリズムについては、[7] の 10 章を参照してください。

参考文献

[1] 深津卓弥, 菱沼徹, and 荒牧大輔. *Python* **と** *CasADi* **で学ぶモデル予測制御**. KS 理工学専門書. 講談社, 2024.

[2] Gianluca Frison and Moritz Diehl. Hpipm: a high-performance quadratic programming framework for model predictive control. *IFAC-PapersOnLine*, 53(2):6563–6569, 2020.

[3] Isak Nielsen and Daniel Axehill. Low-rank modifications of riccati factorizations for model predictive control. *IEEE Transactions on Automatic Control*, 63(3):872–879, 2017.

[4] Leo Emil Sokoler, Gianluca Frison, Martin S Andersen, and John Bagterp Jørgensen. Input-constrained model predictive control via the alternating direction method of multipliers. In *European Control Conference*, pages 115–120. IEEE, 2014.

[5] CV Rao, SJ Wright, and JB Rawlings. Application of interior-point methods to model predictive control. *Journal of Optimization Theory and Applications*, 99:723–757, 1998.

[6] Stephen P Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge University Press, 2004.

[7] Gianluca Frison. *Algorithms and Methods for High-Performance Model Predictive Control*. PhD thesis, Technical University of Denmark, 2015.

[8] Daniel Axehill. Controlling the level of sparsity in mpc. *Systems & Control Letters*, 76:1–7, 2015.

[9] Robin Verschueren, Gianluca Frison, Dimitris Kouzoupis, Jonathan Frey, Niels van Duijkeren, Andrea Zanelli, Branimir Novoselnik, Thivaharan Albin, Rien Quirynen, and Moritz Diehl. acados – a modular open-source framework for fast embedded optimal control. *Mathematical Programming Computation*, 2021.