

ホワイトペーパー

# モデル予測制御 (MPC)

実時間最適制御により制御系のさらなる高性能化を目指す

## はじめに

近年、製品やサービスの価値の追求が急速に進む中で、システムに求められる機能・性能は、高度化かつ複雑化の一途を辿っています。例えば、センサーデータに基づく状況把握や予測、あるいは、次の動き方を支援する行動計画や判断といった高度な技術の開発に向けた挑戦が活発に行われてきました。これらを実現させるためにセンサー情報処理、データ解析、あるいは、人工知能などの分野における様々なアルゴリズムの適用が進められています。制御系の分野においても、厳しい性能要求、システムの複雑化、センサー数の増加といった流れの中で、今後ますます高度な制御技術の必要性が高まっています。

モデル予測制御 (MPC: Model Predictive Control) は、各時刻で未来の応答を予測しながら最適化を行う制御手法です。オンラインで高速に最適化問題を計算しながらフィードバック制御を行います。複雑な系に対して、より高性能な制御を実現することが期待されており、幅広い産業分野で高い注目を集めています。モデル予測制御は、実時間最適制御やReceding Horizon制御と呼ばれる方もされます。本稿では、モデル予測制御の概要について紹介します。基本的な考え方、利点、ならびに、応用先などについて説明し、例題やモデル予測制御の設計ツールについても簡単に紹介します。

## モデル予測制御の考え方

図1は、制御対象とモデル予測制御から構成されるフィードバック制御系のブロック線図を表しています。モデル予測制御の鍵となる技術は「予測」と「最適化」です。モデル予測制御は、コントローラの内部に予測モデル (言い換えると、制御対象モデル) を持つことで、制御対象の現時刻からある有限区間に渡る未来の振る舞いを予測します。制御を行うためには、制御対象の動的な特性、つまり、ダイナミクスを適切に捉えて、モデルとして表現する必要があります。予測モデルの代表例として、ステップ応答モデルやインパルス応答モデル、伝達関数モデル、あるいは、状態方程式を挙げることができます。

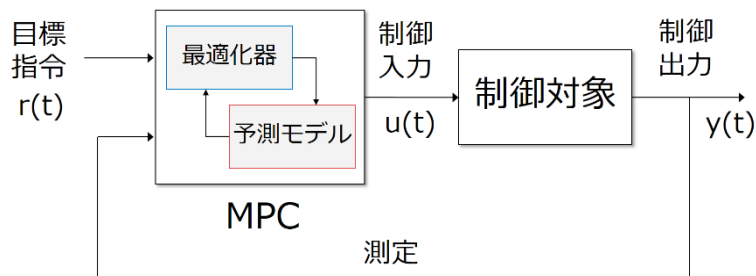


図1: フィードバック制御系のブロック線図

実際に制御を行うためには、予測した結果を利用し、制御対象に与える制御入力を決定しなければなりません。そこで、モデル予測制御では、各サンプリング時間で最適化問題を解くことにより制御入力を一意に決定します。図2に示すように、例えば、制御対象の出力をある目標値に追従させたい場合、現時刻から未来の予測区間において追従誤差の面積を最小化するような制御入力の時系列を探索します。これはある種の最適化問題となるため、数値最適化のアルゴリズムを使用して、解を得ることができます。最終的に計算された時系列の一番目の要素を、実際の制御入力として制御対象に適用します。そして、次のサンプリング時間に移行し、再び最適化問題を解いて制御入力を決定する作業を逐次繰り返してゆきます。フィードバック制御となっているのは、各サンプリング時間において、毎回予測を行う際に、現時刻の状態量をセンサーから計測された値で補正するためです。すなわち、予測の初期値を毎回修正するということです。

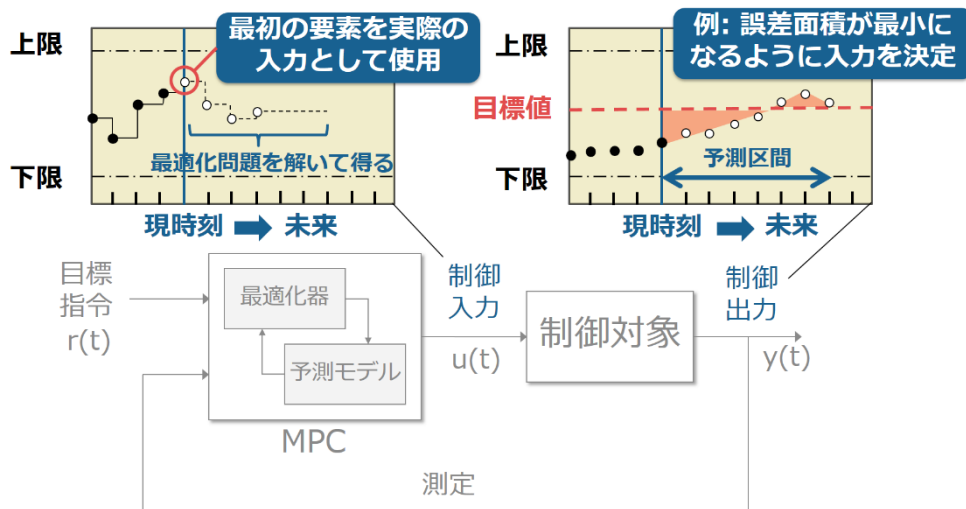


図2: モデル予測制御の仕組み

モデル予測制御では、制御目標・性能を達成するように最適化問題の目的関数と制約条件を定義する必要があります。例えば、小さくしたい(複数の)制御出力の追従誤差、エネルギーコスト、あるいは、制御入力の変化率などを目的関数として定式化します。また、最適化問題では制約条件を扱うことができます。制御対象の入出力が、ある上下限リミットの範囲で動作しなければならない場合、それを不等式制約として定式化し、制約付きの最適化問題を解くことによって、動作範囲を守るような制御入力を探求することができます。このような未来に向かって逐次的に最適化問題を解くという方法が、モデル予測制御の特徴や利点を生み出します。

### モデル予測制御の利点

モデル予測制御は、制御系の性能を最大限に引き出す制御手法として期待されていますが、その背景には次のような利点が考えられます。

1. 物理・性能・安全などの制約を守りながら制御してくれる
2. 多入出力系のような複雑な制御対象にそのまま拡張できる
3. 最適制御の考え方にフィードバック制御の補正効果が加わる
4. 汎用的な手法のため、幅広い応用先が考えられる

前節で述べたように、モデル予測制御の大きな特徴として、制約条件を陽に考慮できる点があります。すなわち、コントローラ自身が制約を理解し、制約を守りながら制御を行う能力を持ちます。これは未来を先読みしながら最適化問題を解くという性質上、制約条件を満足する制御入力を前もって見つけることができるからです。制約の例としては、アクチュエータの出力トルク、加速度、流量などの上下限リミットを上げることができます。また、入出力が多い複雑な制御系(多変数、マルチループ、分散制御など)に対しても同じ考え方をそのまま適用することが可能です。例えば、化学プラントのような大規模な多入出力系、自動車におけるアクセル、ブレーキ、ステアリングの協調制

御、スマートグリッドのエネルギー・マネジメントなど、汎用的な制御手法であるため、様々な複雑なシステムへの応用が考えられます。

一方、課題の一つとして、マイコンなどの計算機への実装が挙げられます。数値最適化を行う場合、PID制御等と比べて、演算量が多く、メモリー (ROM/RAM) も多く消費します。サンプリング時間が非常に高速なアプリケーションや実装する計算機の仕様に厳しい制約があるケースでは適用が困難な面もあります。しかしながら、近年の計算機の目覚ましい進化と数値計算アルゴリズムの高速化や工夫に伴い、徐々にその適用可能な範囲は広がりつつあり、モデル予測制御の性能実証や将来へ向けた新たな制御技術の獲得といった動きが活発に進められています。

## 最適化問題

ここではモデル予測制御における最適化問題がどのように定式化されるのかについて一例を示します。そこで制御対象の予測モデルが、次のような離散時間の線形時不変系として与えられる場合を考えます。

$$\begin{aligned} x(t+1) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) \end{aligned}$$

ただし、 $x$  は状態、 $u$  は制御入力、 $y$  は制御出力、そして、 $t$  は現在の時間を表します。制御入力  $u$  は、各サンプリング時間内で、次の制約付きの最適化問題を解くことで決定します。

$$\min_{u_t, u_{t+1}, \dots, u_{t+p-1}} J = \sum_{k=1}^p (r_{t+k} - y_{t+k})^T Q (r_{t+k} - y_{t+k}) + \Delta u_{t+k}^T R_{\Delta u} \Delta u_{t+k} + u_{t+k}^T R_u u_{t+k}$$

subject to

$$y_{min} \leq y_{t+k} \leq y_{max}$$

$$u_{min} \leq u_{t+k} \leq u_{max}$$

$$\Delta u_{min} \leq \Delta u_{t+k} \leq \Delta u_{max}$$

$p$  は予測ホライズン (予測ステップ数)、 $\Delta u$  は制御入力の前回値からの差分 (変化率) を表します。また、各入出力変数  $u$ 、 $y$ 、 $\Delta u$  の添え字  $\min$ 、 $\max$  は、それぞれ信号の下限値、上限値を意味します。

目的関数  $J$  は、「追従誤差」、「制御入力の変化率」、および、「制御入力の大さき」の3つの項から構成されます。 $Q$ 、 $R_{\Delta u}$ 、 $R_u$  は、各項の重み行列です。各項は入出力変数に関する2次形式で表されており、これは物理的にパワーにあたると解釈することができます。目的関数は、そのパワーを予測区間に渡り時間積分することで得られます。すなわち、上記の最適化問題は、系のエネルギーを最小化する問題であると捉えることができます。各項の重み行列の大きさや予測ホライズンの長さは、制御パラメータとなり、これらの値を変えることによって系の応答を調整します。

では、最適化問題の解き方について考えます。上記の線形時不変系の予測モデルを使用すると、未来の応答  $y_{t+k}$  ( $k = 1, \dots, p$ ) に関する予測式を導出することができます。1ステップ先の予測式は  $y_{t+1} = CAx_t + CBu_t$  となり、 $k$  ステップ先まで状態  $x$  の代入を繰り返すことで予測式  $y_{t+k} = CA^k x_t + \sum_{i=0}^{k-1} CA^i B u_{t+k-1-i}$  を得ることができます。この予測式を目的関数に代入して  $y_{t+k}$  を消去し、整理すると、最終的に次の形式に帰着することができます。

$$\min_U J = U^T H U + 2 F U$$

subject to

$$G U \leq W$$

これは一般的に二次計画問題(QP: Quadratic Programming)と呼ばれます。有効制約法や内点法など、様々な数値解法が存在します。上式において  $U$  は未来の制御入力列

$U = [u_t, u_{t+1}, \dots, u_{t+p-1}]^T$  を表します。各係数行列  $F$  や  $W$  は、元々の目的関数の重み行列に加え、制御対象のシステム行列 ( $A, B, C$ ) やセンサーの測定値  $x(t)$ 、あるいは、目標値にも依存します。すなわち、係数行列が時間によって刻々と変化することを意味します。最終的に数値計算で得られた最適解を  $U^* = [u_t^*, u_{t+1}^*, \dots, u_{t+p-1}^*]^T$  とすると、一番目の要素  $u_t^*$  のみを実際の制御入力として取り出して制御対象に適用します。

## 必要な要素技術

モデル予測制御を構成するために求められる技術要素を改めて以下でまとめます。

### 1. 予測モデリング技術

コントローラに予測モデルが必要となりため、制御対象の支配的なダイナミクスを抽出し、モデリングする術が求められます。一般的に、物理モデリング、統計モデリング、あるいは、その組み合わせによるアプローチでモデル化を行い、伝達関数や状態方程式を導出します。

### 2. 数値最適化技術

最適化問題をリアルタイムで解くための高速な数値最適化ソルバーが必要となります。最適化の問題設定や定式化によって、数値解法のアプローチやアルゴリズムは様々です。例えば、最適化問題が二次計画問題の場合、有効制約法、内点法、あるいは、共役勾配法など様々な解法が存在します。

### 3. 状態観測器 (状態オブザーバ) 設計技術

モデル予測制御はフィードバック制御のため、センサーから測定される情報が必要となります。しかしながら、全ての物理量が測定可能とは限りません。そこで、状態観測器を設計し、観測可能な情報と制御対象のモデルをうまく利用し、未知の状態量を推定します。状態観測器の代表例として、カルマンフィルタを挙げることができます。

### 4. 数値シミュレーション技術

実機による閉ループ系の試験を行う前工程で、PCによるシミュレーション検証は欠かせません。制御対象モデルとMPCモデルを結合した閉ループ系に対して、様々なシナリオや条件を想定した仮想試験を行うことができる数値シミュレーション環境が必要となります。

## モデル予測制御の応用先

モデル予測制御は、1970年代後半に開発され、主に石油精製や化学プロセス等、プロセス制御系を中心に適用されてきましたが、近年の計算機（ハードウェア）の性能向上やアルゴリズム（ソフトウェア）の進化によって、自動車、航空機、産業機械、エネルギー管理システムなどその適用範囲は広がっています。ここでは、自動車、および、エネルギー・インフラ系の分野を例として取り上げます。

自動車の分野においては、様々な応用先が考えられます。エンジン制御の領域では、ディーゼルエンジンのエアパス制御（EGR-VGT）の例を挙げることができます。吸気側に戻す排気ガス流量の制御とターボチャージャーによるブースト圧力の制御を同時に行います。これらを制御するために、EGR弁の開度とタービンハウジング内の排気ガス通路面積を操作する必要がありますが、弁開度や通路面積には物理的な上下制限が存在するため、制約を考慮しながら制御を行わなくてはなりません。別の例として、ハイブリッド自動車を取り上げます。燃費の改善が一つの大きなテーマとなりますが、燃料消費量を抑制するように、エンジン、モーター/ジェネレータ、バッテリー等の各ユニット間でうまく協調させてエネルギーのやりとりを行うことが課題となります。このようなエネルギー管理もモデル予測制御の適用先として考えることができます。また、近年、先進運転支援システム（ADAS: Advanced Driving Assistant System）の開発が急速に進んでいます。この領域においても、例えば、アダプティブクルーズコントロール（ACC）や障害物回避といった車両運動制御への適用が期待されています。

エネルギー管理に対するモデル予測制御は、上述のような自動車の内部のみにとどまらず、家庭、ビル、工場、地域など、エネルギー・インフラ系において幅広く適用先が想定されます。ここでは家庭用エネルギー管理システム（HEMS: Home Energy Management System）について考えます。一般的に家庭におけるエネルギーの流れは、様々な機器を介して行われます。太陽光発電、家電、家庭用/車載用蓄電池、系統連携インバータ、あるいは、給湯器など複数の要素から構成され、エネルギー形態も電気、熱、ガスと様々です。このような機器間で複雑なエネルギーのやりとりを行いながら、いかに上手にエネルギーを発電し、蓄え、賢く使うかを自動化することがエネルギー管理に強く求められます。例えば、家庭における電気料金を抑制するという機能を開発する場合、モデル予測制御が一つの実現手段として考えられます。発電電力や消費電力を数時間や数日先まで予測し、電気料金を最小化するように家庭用蓄電池の充放電パターンを計画するというアプローチはまさにモデル予測制御となります。また、現実には売買電力や機器容量の制限、あるいは、逆潮流防止など、様々な制約が存在します。モデル予測制御では、それらを制約条件として陽に考慮して扱うことができます。

## 例題

DCモーターの位置決め制御を題材に、モデル予測制御がどのような性能を発揮するのかを見てゆきましょう。制御対象は、図3に示すような1入出力のDCモーターです。この系はモーター、ギヤボックス、シャフト、および、機械的負荷から構成されます。入力はモーター電圧、出力は負荷の回転角度です。モーター自身の角度は観測不可と仮定します。

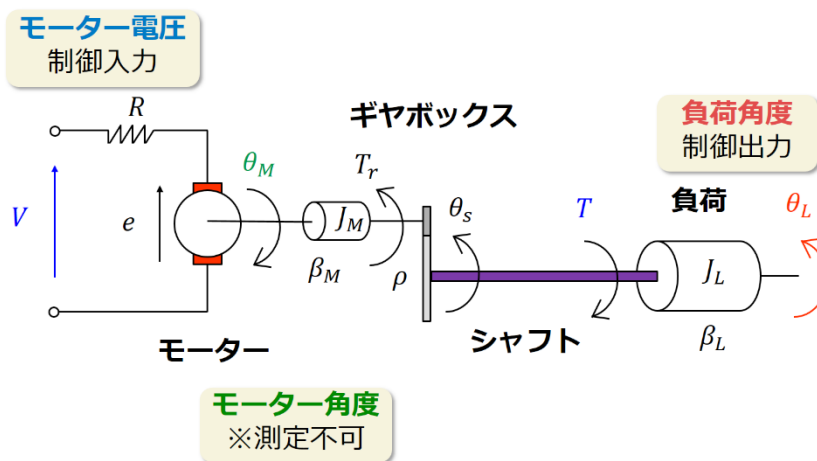


図3:制御対象 (入力:電圧、出力:負荷角度)

次のような問題を設定します。

**目標:**モーター電圧  $V$  を操作し、負荷角度  $\theta_L$  を目標値に追従

**制約:**モーター電圧の上下限制約  $-220 \leq V \leq 220$  [V]、シャフトトルクの上下限制約  $-78.5 \leq T \leq 78.5$  [Nm]

モーター電圧とシャフトに加わるねじれトルクが、上下限リミットを超えないように、位置決め制御を行う問題となります。

予測モデルは、物理法則に基づき運動方程式と回路方程式から線形なモデルを導出します。電気系の応答は、機械系と比べて十分に速いと仮定し、電機子電流のダイナミクスを無視すると、次の微分方程式を得ることができます。

$$\dot{\theta}_L = \omega_L$$

$$\dot{\omega}_L = -\frac{k_\theta}{J_L} \left( \theta_L - \frac{\theta_M}{\rho} \right) - \frac{\beta_\theta}{J_L} \omega_L$$

$$\dot{\theta}_M = \omega_M$$

$$\dot{\omega}_M = -\frac{k_T}{J_M} \left( \frac{V - k_T \omega_M}{R} \right) - \frac{\beta_M \omega_M}{J_M} + \frac{k_\theta}{\rho J_M} \left( \theta_L - \frac{\theta_M}{\rho} \right)$$

$\theta_L$ 、 $\omega_L$  は負荷側の角度、角速度、 $\theta_M$ 、 $\omega_M$  はモーター側の角度、角速度です。各パラメータの意味と値を表1に示します。

記号	パラメータ値 (SI単位系)	定義
$k_\theta$	1280.2	ねじり剛性
$k_T$	10	モーター定数
$J_M$	0.5	モーターイナーシャ
$J_L$	50J_M	公称負荷イナーシャ
$\rho$	20	ギア比
$\beta_M$	0.1	モーター粘性摩擦係数
$\beta_L$	25	負荷粘性摩擦係数
$R$	20	電機子抵抗

表1:DCモーターパラメータ

状態を  $x = [\theta_L \ \omega_L \ \theta_M \ \omega_M]^T$  とおくと、次の状態空間モデルを導出することができます。

$$\dot{x} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -\frac{k_\theta}{J_L} & -\frac{\beta_\theta}{J_L} & \frac{k_\theta}{\rho J_L} & 0 \\ 0 & 0 & 0 & 1 \\ \frac{k_\theta}{\rho J_M} & 0 & -\frac{k_\theta}{\rho^2 J_M} & \frac{\beta_M + \frac{k_T^2}{R}}{J_M} \end{bmatrix} x + \begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{k_T}{R J_M} \end{bmatrix} V \quad \dots \quad \text{状態方程式 } \dot{x} = Ax + Bu$$

$$\theta_L = [1 \ 0 \ 0 \ 0]x \quad \dots \quad \text{出力方程式 } y = Cx$$

ただし、これは連続時間系のモデルであるため、コントローラに適用する際には、制御サンプリング時間で離散時間系に変換する必要があります（※その過程は省略）。また、シャフトに印加されるトルクは次式となります。

$$T = \left[ k_\theta \ 0 \ -\frac{k_\theta}{\rho} \ 0 \right] x$$

これは負荷側とギアボックス側の回転角度の差に比例（ねじり剛性を掛け算）したトルクが発生することを意味しています。



モデル予測制御の最適化問題として、例えば、次式のように定式化することができます。

$$\min J = \sum \{q(\theta_L - \theta_{ref})^2 + r\Delta V^2\}$$

subject to

$$-220 \leq V \leq 220$$

$$-78.5 \leq T \leq 78.5$$

上の目的関数では、負荷回転角度の追従誤差とモーター電圧の変化率を評価しています。チューニングをする際、重みの  $q$ 、 $r$  の大きさや予測ホライズンの長さなどをパラメータとして変更します。直感的に  $q$  を相対的に大きくとると、追従誤差の評価が支配的となり、速い追従応答を得ることができます。一方、 $r$  を相対的に大きくとると、モーター電圧の変化を抑制する効果が強まり、追従が遅くなることが予想されます。

図4 にモデル予測制御を適用した場合の数値シミュレーション結果を示します。上から負荷回転角度、モーター電圧、シャフトトルクの時間応答を表しています。負荷回転角度の目標値は  $\theta_{ref}=\pi$  [rad] (=180 [deg])、制御サンプリング時間は0.1sec、目的関数の重みは  $q=10$ 、 $r=0.05$ 、予測ホライズンは  $p=10$  ステップ (すなわち、 $0.1 \times 10=1$  sec) です。本例ではモーター側の角度と角速度は観測不可のため、カルマンフィルタを使用し状態推定を行っています。図4より、負荷回転角度が目標値に追従しているのと同時に、モーター電圧およびシャフトトルクが設定した制約の範囲内で動作していることが分かります。特に、立ち上げ時に、シャフトトルクが下限値-78.5Nmを越えないように、駆動電圧をゆっくりと上昇させています。そして、駆動電圧を上限220V一杯に張り付けて、できるだけ速く追従するように制御していることが分かります。

一方、比較対象として、PI制御を適用した場合の数値シミュレーション結果を図5に示します。これはモデル予測制御と同等の応答速度で追従するように調整を試みた結果です ( $K_p=180$ 、 $K_i=2$ )。モーター電圧とシャフトトルクの制約を同時に満たすことが困難であったため、MPCコントローラ出力 (電圧指令) に強制的に飽和リミッターをかけています。しかしながら、それでもMPCと同等の速応性を保ちながらシャフトトルクの制約を守るのは困難です。そのため、制約を満たすためには、比例ゲインを落として速応性を犠牲にせざるを得ません。このように、制約の範囲内で動作しながら高い性能を達成するには、通常の線形なフィードバック制御器のみではなかなか難しいケースもあります。

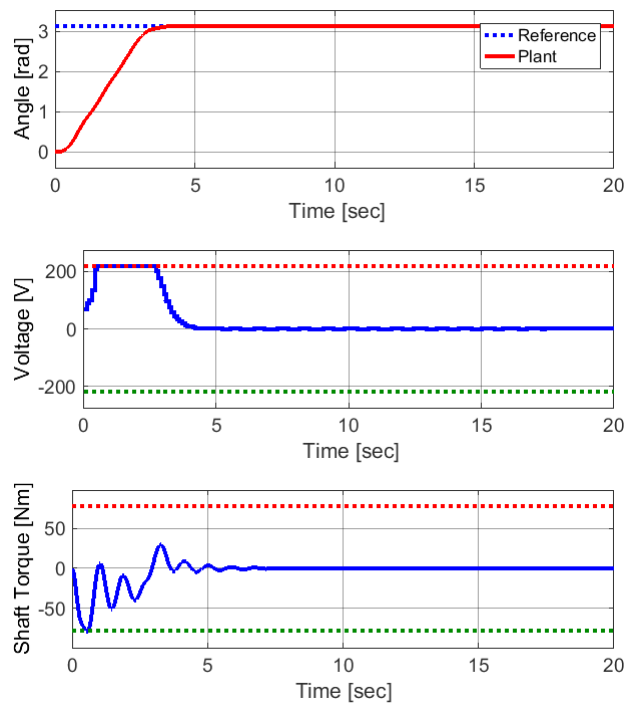


図4: MPC 数値シミュレーション結果  
(上から、負荷回転角度、モーター電圧、シャフトトルクの時間応答)

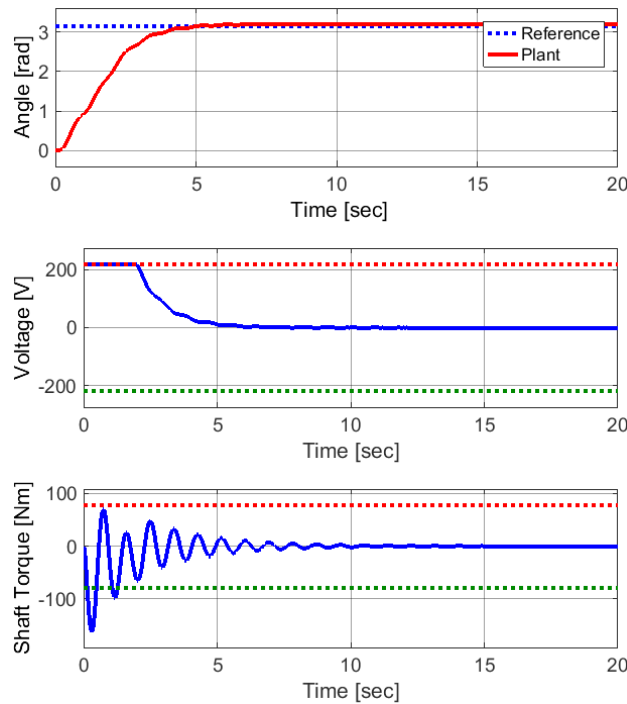


図5: PID 数値シミュレーション結果  
(上から、負荷回転角度、モーター電圧、シャフトトルクの時間応答)

## モデル予測制御設計ツール

MATLAB®は、行列計算にはじまり制御系設計・解析のツールとして長い歴史を持ち、学术界および産業界で幅広く利用されています。モデル予測制御の設計やシミュレーションに関しては、Model Predictive Control Toolbox™という専用の拡張パッケージが用意されています。

Model Predictive Control Toolboxは、20年以上の長い歴史を持ちます。当初から著名な研究者 (Prof. M. Morari, Prof. N. Ricker, and Prof. A. Bemporad) と共同開発されたツールです。

Model Predictive Control Toolboxは、次のような機能を提供しています。

- 専用の数値最適化 (QP) ソルバー
- 設計・シミュレーション用のユーザーインターフェース
- Simulink® 用の MPC コントローラブロック
- コントローラブロックの自動 C コード生成対応

最適化問題の定式化やソルバーはツール側で準備されているため、予測モデルを設計し、必要なパラメータ(予測区間、目的関数の重み、制約条件など)を設定すれば、すぐにシミュレーション実行が可能です。モデル予測制御の性能評価にいち早く到達できるのがModel Predictive Control Toolboxの大きな利点です。

Model Predictive Control Toolbox を中心としたMPC設計のワークフローを図6に示します。「予測モデリング」、「MPC設計」、「シミュレーション」、および、「実装・試験」の4つのステップに分けています。

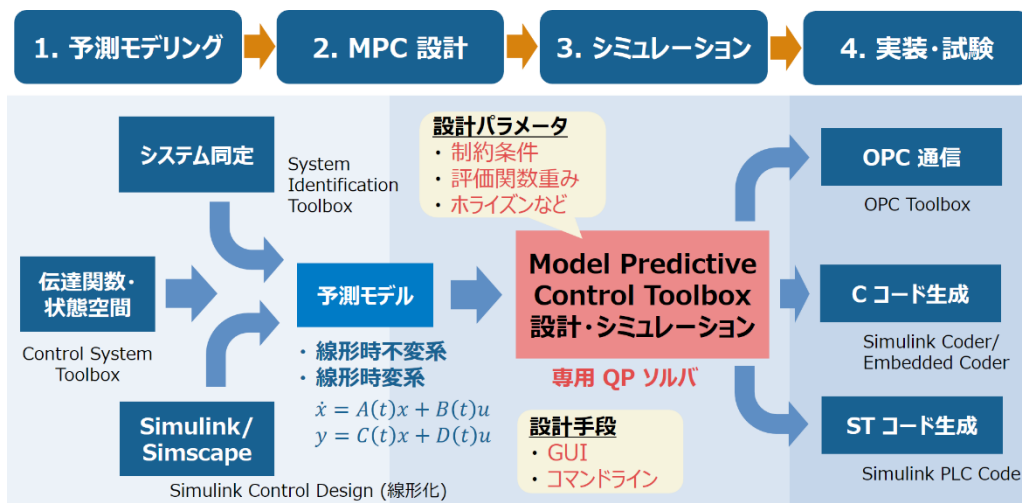


図6: MPC設計のワークフロー

予測モデリングのステップでは、制御系設計ツールを併用しながら、物理モデルや統計モデルを構築します。上述のDCモーターの例のように、物理方程式から伝達関数や状態空間モデルを作成する方法や、あるいは、制御対象の入出力測定データからシステム同定を通じてモデル化する方法などいくつかのアプローチが考えられます。

MPC設計とシミュレーションのステップでは、設計した予測モデルを取り込み、MPCコントローラに必要なパラメータを設定します。Model Predictive Control Toolboxでは、MATLAB言語によるプログラミング、あるいは、グラフィカルユーザーインターフェース (GUI) により必要なパラメータを設定し、MPCコントローラを作成することが可能です。そして、Simulinkによる閉ループ系の数値シミュレーションを通じて設計の妥当性を確認します。

実装・試験のステップでは、設計したMPCコントローラモデルから、自動コード生成ツールを使用してC言語やST言語に変換し、ハードウェア実装して制御試験を行います。他にも、プロセス制御や計装の世界で利用されているOPC通信経由でMATLABから制御を行う手段も用意されています。

※ ST言語: IEC 61131-3 Structured Text (構造化テキスト)、PLC用言語

## 参考情報

### 制御系設計Web ビデオ

- [モデル予測制御 \(MPC\)](#) (1:03:56)
- [Model Predictive Control of Diesel Engine Airpath \(英語\)](#) (28:19)
- [PID制御をSimulinkでより簡単に](#) (59:07)
- [非線形制御モデル化・実践 \(スライディングモード制御編\)](#) (49:23)

### Model Predictive Control Toolbox (英語)

- [製品概要ページ](#)
- [製品付属例題](#)