

# モデル予測制御に基づいた 高度プロセス制御ソリューション

## An Advanced Process Control Solution Based on Model Predictive Control

アズビル株式会社  
技術開発本部

田原 鉄也  
Tetsuya Tabaru

アズビル株式会社  
技術開発本部

藤江 真也  
Shinya Fujie

### キーワード

プロセス制御, 高度制御, 多変数モデル予測制御, 経済最適化, 多装置間最適化, SORTiA

本論文では、開発したプロセス向け高度制御システムで使われている制御・最適化技術について報告する。コアとなるモデル予測制御コントローラはチューニングが容易である。これは、3自由度構成、参照軌道と自由応答を介した制御機構、制御演算の自動設定、正規化された調整パラメータによるものである。また、定常目標値の最適化機能を持ち、積分要素を持つプロセスにも対応する。本稿ではその手法についても述べる。本ソリューションは多装置間最適化機能も持つ。本機能により過渡状態を含めたプラント全体最適化ができることを数値例で示す。

This article is concerned with an advanced process control system, and in particular, with the control and optimization technology used in it. A model predictive controller plays a central role in this system. The controller is designed to reduce tuning efforts, which is accomplished by a three-degree-of-freedom structure, a control mechanism using a reference trajectory and free response, automatic setup of controller output calculation, and normalized adjustment parameters. Target values for the controller are provided from a steady state optimizer that handles integrating processes by means of a novel method described in this article. In addition, our solution includes a plant-wide optimization module, which is intended to cooperate with the model predictive controllers. A numerical example illustrates that this cooperation enables optimization of the entire plant, including both steady state and transitional behavior.

## 1. はじめに

モデル予測制御(以下、MPC (Model Predictive Control) と表記)は、装置や製品仕様の制約を明示的に扱えるという特長があり、プロセス産業、特に石油精製や石油化学産業で普及している<sup>(1),(2)</sup>。これに応じ、多変数モデル予測制御ソリューションが各社から提供されている<sup>(1)</sup>。ユーザにとってこれらのソリューションの導入目的は、装置運転上の制約を確保することによる装置の安定化、さらには、それによって生じた余裕を利用した経済最適化にある<sup>(3)</sup>。この経済最適化により、製品品質をスペック内に維持しながらエネルギー消費量を最小化する、エネルギー原単位を低減するといった価値が生まれる。

当社は高度プロセス制御ソリューションとしてSORTiAシ

リーズを自社開発し、お客さまに提供している。中核であるSORTiA-MPCは経済最適化を含んだ多変数MPCコントローラである。また、複数のMPCの上位に位置し、多装置間最適化を実現するSORTiA-LQPも開発している。

SORTiA-MPCの特長の1つはチューニングが容易ということである。この特長は、高度制御プロジェクトのエンジニアリング作業効率化につながる。また、開発にはプロセス制御に長年取組んできたエンジニアが参加しており、現場の知見や経験が製品に反映されている。例えば、干渉が強いプロセスの取扱い、応答時間が大きく異なる制御変数が混在する場合の制御、積分要素を持つプロセスの制御・最適化など、一般に制御が難しいとされる制御対象を考慮した仕様となっている。

本論文では、チューニングが容易という特長を実現するた

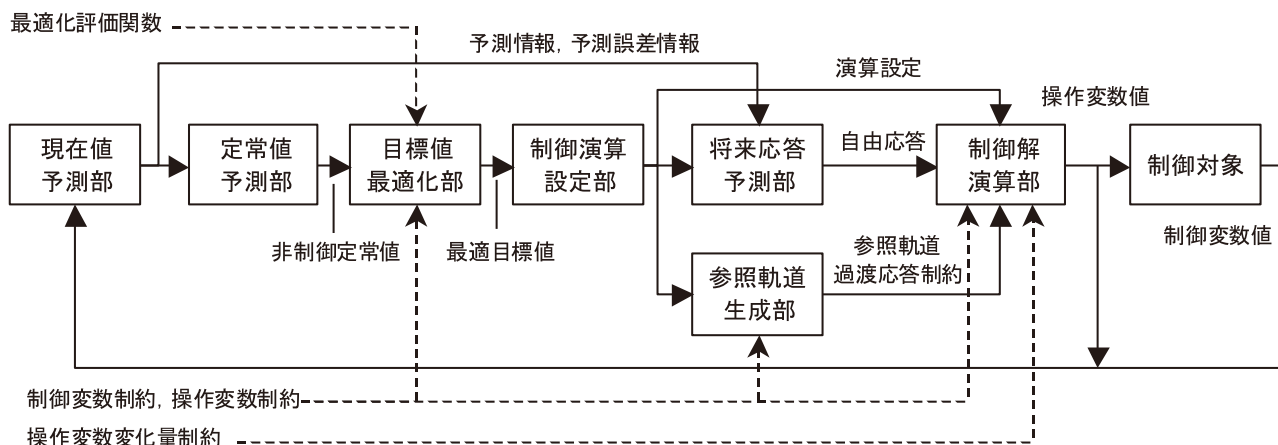


図1 制御・最適化ロジック全体図

めのMPCコントローラ設計(3章)と、積分要素を持つプロセスの定常目標値最適化手法(4章)を中心に、SORTiA-MPCで使われている技術について説明する。また、SORTiA-LQPと連携した多装置間最適化についても述べる(5章)。

## 2. 制御・最適化ロジックの概要

本題に入る前に、SORTiA-MPCの制御・最適化ロジックの全体構成と各構成要素について概説する。

図1は制御・最適化ロジックの全体構成図である。以下、各構成要素について簡単に説明する。

**現在値予測部:** 制御変数の現在値を制御対象の伝達関数モデルを用いて推定する。また、予測誤差も算出する。

**定常値予測部:** 現在の状態が継続したと仮定して、制御変数が収束する値(以下、非制御定常値と呼ぶ)を推定する。

**目標値最適化部:** LP (Linear Programming), またはQP (Quadratic Programming)により、評価関数を最小にする最適定常目標値を算出する。これにより、目的の1つである経済最適化を実現する。

**制御演算設定部:** 制御対象のモデルと最適化の結果を用いて、制御解演算の設定を決定する。

**将来応答予測部:** 制御演算設定部が決めた区間の終端まで反復演算を行い、操作変数を変えない場合の制御変数の将来応答(自由応答)を算出する。

**参照軌道生成部:** 最適定常目標値へと漸近する参照軌道を生成する。また、制御変数の過渡応答に対する制約(以下、過渡応答制約と呼ぶ)も算出する。

**制御解演算部:** 制御変数や操作変数の応答に課せられた制約を守りつつ、変数の応答が参照軌道に近づくような操作変数時系列を求める。得られた時系列の最初の値を操作変数の制御解として出力する。

## 3. チューニングが容易な MPC コントローラの設計

本章では、制御・最適化ロジックのうち、目標値へと動的制御を行う部分について説明する。1章でも述べたように、SORTiA-MPCの特長の1つはチューニングが容易とい

うことである。本章ではその容易さを実現する方法を中心に説明する。

後述するように、制御演算のベースは参照軌道を用いた線形モデル予測制御であり<sup>2)</sup>、特別な技術ではない。だが、制御器の3自由度構成、自由応答や参照軌道を介した挙動の制御、制御解演算設定の自動設定、調整パラメータの正規化といった工夫により、チューニングの容易さを実現している。

本章の構成は以下の通りである。3.1節では3自由度構成について、3.2節では制御変数の整定時間を調整するパラメータについて述べる。3.3節では線形MPCによる制御解演算について説明する。3.4~3.6節では、目標値追従動作、フィードバック動作、フィードフォワード動作について、各動作の実現機構について説明する。3.7節では制御演算設定部で自動的にされる設定について述べる。

### 3.1 制御器の構成

SORTiA-MPCは、目標値追従動作、フィードバック動作、フィードフォワード動作が独立している3自由度構成である。図2に概念図を示す。

目標値追従動作は目標値(SP)に適用される前置フィルタ( $F(z)$ )で決まる。このフィルタは目標値変更時の制御系の望ましい挙動を定める参照モデルとなる。フィードバック制御は、未知外乱により発生する予測誤差(制御変数値と予測値の差)を基に、外乱を抑制するように働く。フィードフォワード制御は、観測外乱(DV)から制御変数への影響を打ち消すように働く。各動作が独立していることが、チューニングの見通しを良くすることに寄与している。

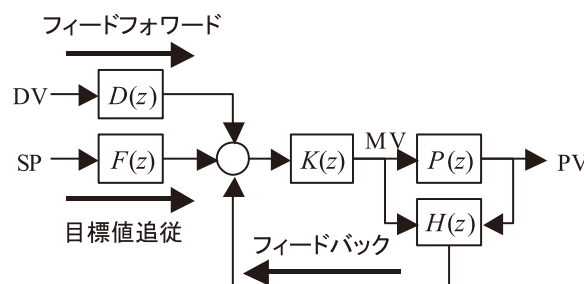


図2 3自由度構成の概念図

### 3.2 制御変数の整定時間を調整するパラメータ

制御変数の整定時間を調整するパラメータは以下の3つである。

- (1) 最適化速さ
- (2) フィードバック速さ→強さ
- (3) フィードフォワード強さ

最適化速さは制御変数が上下限制限内に制御され、最適化を行っている場合に適用される。具体的には、最適目標値が上下限内で変わった時に新しい目標値へと整定させるまでの時間や、上下限内で未知外乱の影響を受けた際に最適値へ戻す整定時間を決めるために使われる。この整定時間は開ループ応答の平均整定時間を基準として次のように決められる。

$$\text{整定時間} = \frac{\text{開ループ応答平均整定時間}}{\text{最適化速さ}} + \text{むだ時間} \quad (1)$$

このように調整パラメータが正規化されていることも、チューニングの容易さに役立っている。実際の高度制御プロジェクトでも、デフォルト値から開始して、そこからの微調整で済むことが多い<sup>(4)</sup>。

フィードバック強さは上下限制限から外れた制御変数を制約内へと制御する際に使われる。例えば、現在値よりも内側に上下限を変えた時の整定時間、未知外乱によって制御変数が上下限から逸脱した場合に制約内へと戻す時間が該当する。整定時間の計算方法は最適化速さの場合と同様である。

フィードフォワード強さは、観測外乱が制御変数に及ぼす影響をフィードフォワード制御で抑制する際に適用される。詳しくは3.6節で説明するが、やはり正規化されたパラメータになっている。

### 3.3 制御解演算の定式化

制御解演算部のコアは、従来からある線形モデル予測制御である。SORTiA-MPCでは、

- (1) 操作変数が上下限を守る
- (2) 操作変数の1制御周期あたり変化量が指定された上下限を守る
- (3) 制御変数の予測ホライズンまでの将来応答が過渡応答制約を守る

といった制約を満たしつつ、

- (A) 制御変数の予測ホライズンまでの将来応答、および、操作変数の制御ホライズンまでの時系列を参照軌道にできるだけ近づけ、
- (B) 操作変数の1制御周期あたり変化量の二乗和をできるだけ小さくする

ような操作変数時系列を求め、得られた時系列の最初の値を操作変数値として出力する。

この問題は制約付きの二次計画法になる。以下、制御変数、操作変数が1つずつで、制御変数のみを参照軌道に近づける場合の定式化を例として示す。

操作変数の時系列を $\hat{u}(k+i|k)$ 、その1制御周期変化量を $\Delta\hat{u}(k+i|k)$ で表す。すなわち、

$$\hat{u}(k+i|k) = u(k-1) + \sum_{l=0}^{i-1} \Delta\hat{u}(k+l|k) \quad (2)$$

である。時系列 $\hat{u}$ を制御対象に入力した場合の制御変数 $y(k)$ の応答を $\hat{y}(k+i|k)$ で表し、その参照軌道を $r(k+i|k)$ で表す。操作変数の下限と上限を $\underline{u}$ ,  $\bar{u}$ で、その1制御周期あたり変化量の下限と上限を $\underline{\Delta u}$ ,  $\overline{\Delta u}$ で表す。また、過渡応答制約を $\underline{y}(k+i|k)$ と $\bar{y}(k+i|k)$ で表す。これらは定常時は上下限と一致し、目標値変更時は新しい目標値に漸近する時系列となる。予測ホライズンを $H_p$ で、制御ホライズンを $H_w$ で表す。

以上のように定義すると、制御解演算は、

$$\underline{\Delta u} \leq \Delta\hat{u}(k+i|k) \leq \overline{\Delta u} \text{ for } k=1, \dots, H_w-1 \quad (3)$$

$$\underline{u} \leq \hat{u}(k+i|k) \leq \bar{u} \text{ for } k=1, \dots, H_w-1 \quad (4)$$

$$\underline{y}(k+i|k) \leq \hat{y}(k+i|k) \leq \bar{y}(k+i|k) \quad (5)$$

for  $k=H_w, H_w+1, \dots, H_p$

を制約とした、以下の評価関数の最小化問題になる。

$$V(k) = \sum_{i=H_w}^{H_p} Q(i) (\hat{y}(k+i|k) - r(k+i|k))^2 + \sum_{i=0}^{H_w-1} R(i) \Delta\hat{u}(k+i|k)^2 \quad (6)$$

ただし、 $H_w$ は1以上 $H_p$ 未満の整数、 $Q(i)$ は制御変数の応答と参照軌道との差に掛かる重み、 $R(i)$ は操作変数の変化量に掛かる重みである。前者は制御された応答を参照軌道に近づけるため、後者は操作変数の変化を抑制するためにある。

なお、実際には、予測ホライズン、制御ホライズン共にブロッキング(間引き)されているため、それを考慮した式が使われている。

### 3.4 目標値追従動作

本節では目標値追従動作を実現する仕組みについて述べる。また、数値シミュレーションにより機能が実現されていることを例示する。

#### 3.4.1 動作原理

目標値変更時に新しい目標値へと追従する動作は、目標値に適用する前置フィルタで実現している。

まず、制御周期毎にフィルタ演算を行い、フィルタ適用後の値を求める。それから、現在の目標値が継続すると仮定して、予測ホライズンまでのフィルタ出力値時系列を反復演算で求める。この系列を参照軌道 $r(k+i|k)$ として制御解演算部に与える。

図3に目標値をステップ状に変更した場合の例を示す。現在時刻(0)で目標値(破線)が0から1に変わったとすると、太線のように、指定された整定時間をかけて0から1へと漸近するような参照軌道が算出され、制御解演算部に渡される。

この演算は最適目標値のみ用い、制御変数の現在値は使わない。そのため、最適目標値へと追従させる途中で外乱が入った場合に整定時間が延びるという問題が構造上排除されている。これは3自由度構成から来る利点である。

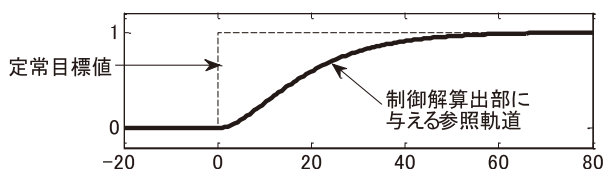


図3 目標値変更時の参照軌道

### 3.4.2 数値例

例で用いる制御対象は、いずれも制御変数と操作変数を2つずつ持つ。以下では1番目と2番目の制御変数をCV1, CV2と表記する。操作変数についても同様にMV1, MV2と表記する。

次式の伝達関数で表される制御対象を考える。

$$G_{CV/MV}(s) = \begin{pmatrix} \frac{-1.2}{25s^2+8s+1} & \frac{0.6}{16s^2+8s+1} \\ \frac{-0.4}{4s+1} & \frac{1.4}{6s+1} \end{pmatrix} \quad (7)$$

CV1とCV2の下限と上限は-1と1, MV1とMV2の下限と上限は-2と2とする。初期値は全ての変数で0とする。定常目標値最適化の目的はMV1の最小化(LP)とする。最適化速度は、CV1の整定時間が150, CV2が50になるように設定した。

図4にシミュレーション結果を図示する。実線が各変数の時系列、破線は最適目標値を表している。CV1とCV2がそれぞれ、最適化速度によって指定された時間で目標値に到達していることがわかる。

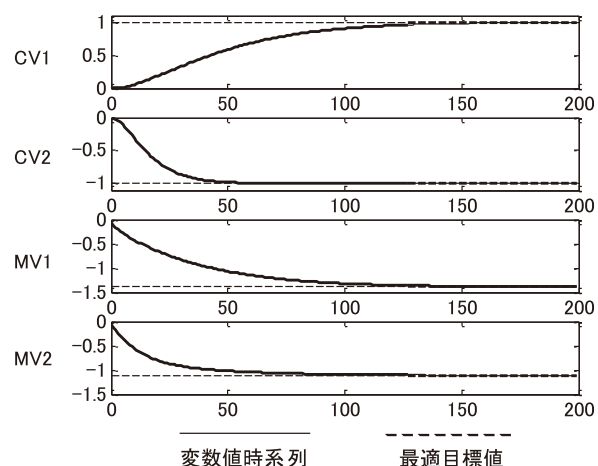


図4 目標値追従応答

## 3.5 未知外乱に対するフィードバック制御

本節では未知外乱に対するフィードバック制御とその強さを調整する方法について説明する。また、数値シミュレーションにより、機能が目標値追従動作と独立して実現していることを例示する。

### 3.5.1 動作原理

制御変数に入った未知外乱を抑制する動作は、予測誤差(制御変数の測定値と予測値の差)をフィルタに通してフィードバックすることで実現している。

オフセット外乱が入った場合を例に説明する。図5上段のように、時刻0でオフセット外乱が入り、予測誤差が1になったとする。ここで、予測誤差を自由応答に反映することでフィードバック制御の効果が得られるが、その際に予測誤差をそのまま加えるのではなく(細線)、指定された時間で漸近する時系列(太線)を加えている。この演算は、目標値から外れた制御変数に対して現在値から目標値まで参照軌道をひくことと等価である(図5下段)。よって、漸近する時系列を生成するフィルタの整定時間によって、フィードバック制御動作を調整できる。

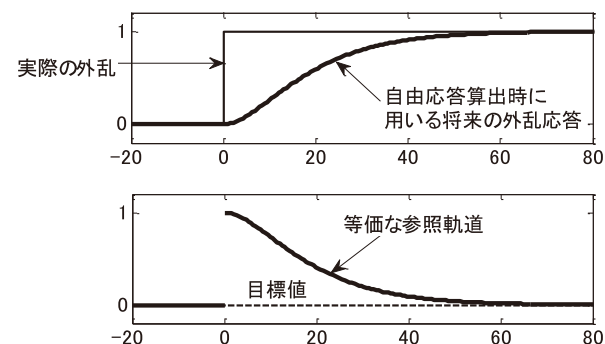


図5 オフセット外乱に対するフィードバック制御動作

### 3.5.2 数値例

制御対象, 上下限設定, 最適化設定は前節と同じである。初期値はCV1が0.5, CV2が-0.5, MV1とMV2が0である。最適化速度はCV1, CV2の整定時間が60となるように、フィードバック強さは、CV1, CV2の整定時間が20となるように設定した。

本数値例の目的は、制御変数に加えられた未知外乱に対する操作とその結果を見ることである。この例では、時刻が30でCV1に+0.4のオフセット外乱を印加する。

図6にシミュレーションの結果を示す。細かい実線は外乱が無い場合の応答、太い実線は外乱があった場合の応答である。CV1の破線は目標値かつ上限, CV2の破線は目標値かつ下限を示す。なお、本例のCV1, CV2は、外乱が入っても最適解は変わらない。

CV1が外乱により上限から外れた後、フィードバック強さで決まる整定時間で上下限内に戻されている。また、最適目標値に到達するまでの時間は外乱の有無に関わらず最適化速度で定まる整定時間になっており、フィードバック制御の影響を受けていないことがわかる。

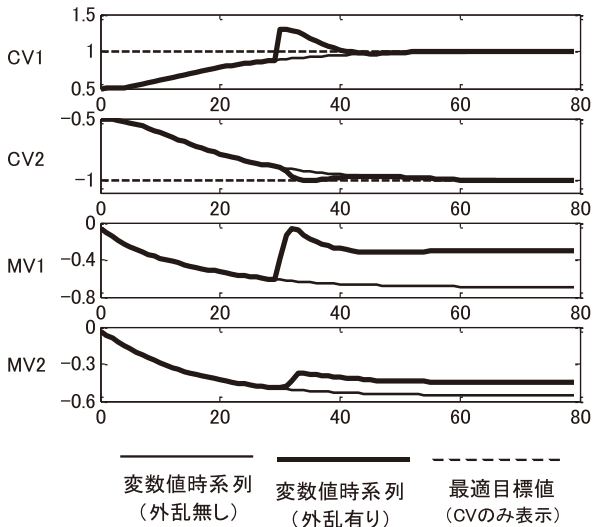


図6 未知外乱に対するフィードバック制御

### 3.6 観測外乱に対するフィードフォワード制御

本節では観測外乱に対するフィードフォワード制御とその強さを調整する方法について説明する。また、数値シミュレーションにより、操作の強さが調整できていることを例示する。

#### 3.6.1 動作原理

フィードフォワード制御は、自由応答算出時に外乱変数からの寄与を加算することで行う(文献(2) 5章参照)。こうすると、外乱変数からの寄与を抑制するような操作変数系列が得られ、フィードフォワード制御が実現できる。この方法はMPCでフィードフォワード制御を行う最も簡単な方法ではあるが、強さの調整はできない。外乱変数の影響を完全に抑えようとすると操作変数の変化が大きくなるが、そのような操作は現場では敬遠されるため、フィードフォワードの強さを調整できることが要求される。

そこで、制御対象のモデルで計算した応答をそのまま使わずに、意図的に遅く修正した応答を使う(図7)。例えば、観測外乱による制御変数への影響が、図7の細線のようにになると予測されたとする。ここで自由応答を算出する際に、太線のように意図的に遅く修正した予測を使う。修正が無ければ、図7下段の実線で示した自由応答を破線の目標値に近づけるよう制御される。しかし、修正により参照軌道を図7下段の太線のようにしたと等価になる。この結果、フィードフォワード操作が弱められる。

修正した応答は、元のモデルの代わりに次式の伝達関数で応答計算することで得ている。

$$G_{FF}(s) = G_N(s)/G_D(s/\gamma) \quad (8)$$

ただし、 $G_N(s)$ は観測外乱から制御変数への伝達関数の分子、 $G_D(s)$ は分母である。式(8)の伝達関数は、フィードフォワード強さ $\gamma$ で調整でき、 $\gamma$ が1の場合は応答が元のモデルと一致し、0に近づけるにしたがって応答が徐々に遅くなるようになっている。

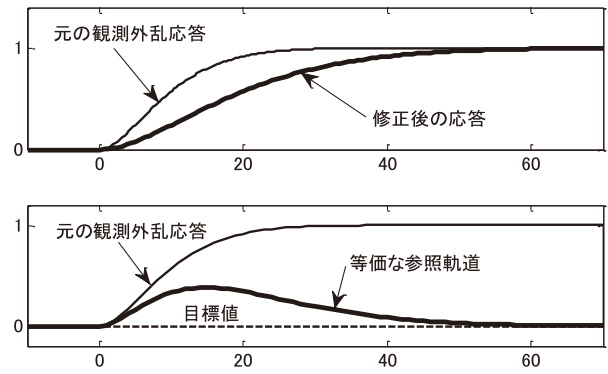


図7 フィードフォワード制御の動作

#### 3.6.2 数値例

制御対象は前節までのものに外乱変数DV1を追加したものである。DV1からCV1, CV2への伝達関数は次式で表される。

$$G_{CV,DV}(s) = \begin{pmatrix} \frac{0.7}{30s^2+10s+1} \\ -0.3 \\ \frac{0.7}{25s^2+10s+1} \end{pmatrix} \quad (9)$$

上下限は前節までと同じに設定した。最適化の目的も前節までと同じでMV1の最小化である。初期値はCV1が1, CV2が-1, MV1とMV2が0であり、既に最適解に到達した状態となっている。

本数値例の目的はDV1に観測外乱が入った場合の応答とフィードフォワード強さの関係を確かめることである。この例では、時刻10でDV1に+0.1のオフセット外乱が印加される。これは、CV1, CV2を上下限外に出すような外乱である。

図8にシミュレーション結果を図示する。細い実線はフィードフォワード強さを1.0に、太い実線は0.6にそれぞれ設定した時の応答である。CV1とCV2の破線は図6と同じ意味である。

フィードフォワード強さが1.0の場合はCV1, CV2が速やかに上下限内に戻されており、DV1の影響を強く抑えている。一方で、操作変数の立ち上がりが急である。それに対し0.6の場合は、操作変数の立ち上がりが抑えられており、徐々に目標値へ近づく操作になっている。このため、CV1とCV2が上下限内に戻るまでの時間も長くなっている。

### 3.7 制御解演算の自動設定

3.3節で示したMPC演算を実行するためには、予測ホライズンや制御ホライズン、重み $Q, R$ を調整する必要がある。SORTiA-MPCではこれらの設定が自動で行われるようになっており、チューニングの負荷軽減に寄与している。

予測ホライズンは制御対象のモデルから開ループ応答時間の重み付き平均値を求め、その値を基に決めている。制御ホライズンは、予測ホライズンまで制御が可能となるように決められる。

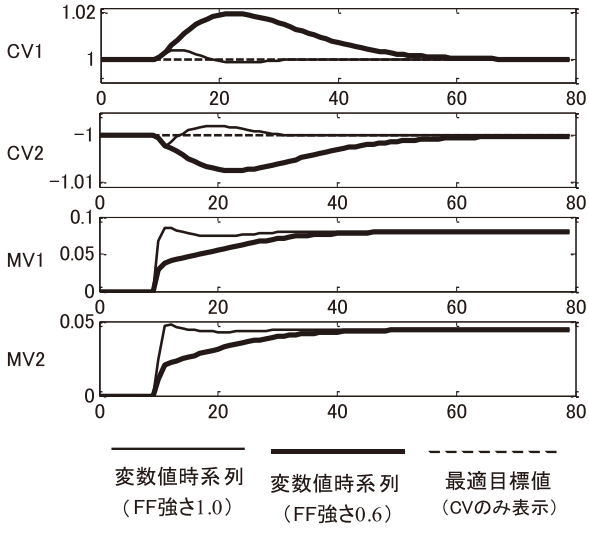


図8 観測外乱に対するフィードフォワード制御

重み $Q, R$ の自動調整はコントローラの構造が故に比較的容易となっている。本コントローラは制御の挙動を参照軌道と自由応答の修正によって調整しており、 $Q, R$ にはあまり依存しない。このため、変数のスケーリングをした後であれば、制御対象に依存したチューニングはあまり必要ではない。文献(2)の7章では、重み $Q, R$ をチューニングするよりもフィルタを適切に設計する方が容易であると述べられているが、本コントローラはその事実に沿った設計になっている。ただし、重みの調整が必要なケースもあるので、 $R$ を相対的にチューニングするパラメータを用意している。

#### 4. 積分要素を持つプロセスの定常最適化

本章では、積分要素を持つプロセスの最適目標値を算出する方法について述べる。

積分要素が無い通常のプロセスでは、定常状態における制御変数と操作変数の関係は直流ゲイン行列で記述できる。そして、直流ゲイン行列に加えて、制御変数と操作変数の上下制限約、評価関数を与えれば、最適化問題を定式化できる。

しかし、プロセスが積分要素を持つ場合は、以下の理由で同じ方法が通用しない。

- (1) ある操作変数からある制御変数への伝達特性に積分要素があると、対応する直流ゲインが無窮大となるため、制御変数と操作変数の関係を直流ゲイン行列で記述できない。
- (2) 積分要素に掛かる制御変数が定常状態で傾きを持つと、制御変数の定常状態での予測値などが発散して計算不能となる。

よって、積分要素を持つプロセスで最適目標値を決めるには何らかの工夫が必要となる。本章では、SORTIA-MPCで使われている手法について説明し、数値シミュレーションでその有効性を示す。

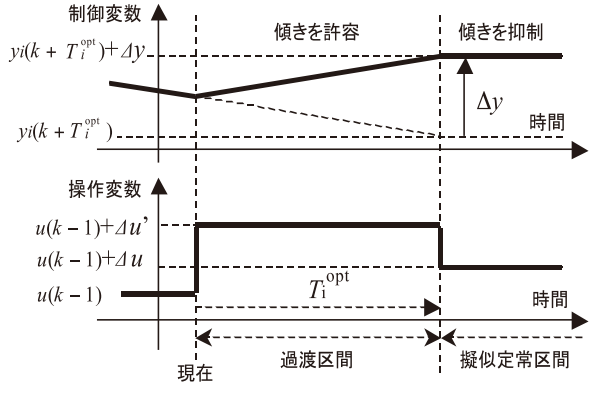


図9 積分要素を持つプロセスの最適化の定式化

#### 4.1 最適化問題の定式化

積分要素を持つプロセスの最適化を可能にするため、以下の方針で最適化問題を定式化した(図9)。

- (1) 積分要素に掛かる制御変数の値については、本来の意味での定常状態の値ではなく、所定の最適化時間を決め、その時間が経過した後の値を最適化対象とする。評価関数も積分要素に掛かる制御変数については最適化時間後の値を、それ以外の制御変数と操作変数については定常値から算出する。
- (2) 将来の時間区間を、最適化時間を境に過渡区間と擬似定常区間の2つに分割し、各区間の変数値を定式化に組み込む。すなわち、変数の値が両区間で異なることを許容する。
- (3) 値に対する上下制限約は各区間についてそれぞれ設定する。
- (4) 積分要素に掛かる制御変数の傾きについては、擬似定常区間のみ傾きを0にする制限約を設定し、過渡区間は傾きを許容する。

以下、具体的な式で説明する。操作変数を $u_1, \dots, u_m$ , 制御変数を $y_1, \dots, y_n$ で表す。最適化は非制御定常値を原点とし、そこからの差分に対して行う。すなわち、1つ前の制御周期の操作変数と、その状態を維持したと仮定した場合の制御変数の定常予測値(積分要素を持たない場合)、または所定の最適化時間後の予測値を原点とする。

$$\mathbf{u}^0 = [u_1(k-1), u_2(k-1), \dots, u_m(k-1)]^T \quad (10)$$

$$\mathbf{y}^0 = [y_1^0, y_2^0, \dots, y_n^0]^T$$

$$y_i^0 = \begin{cases} \hat{y}_i(\infty), & \text{積分系を持たない場合} \\ \hat{y}_i(T_i^{\text{opt}}), & \text{積分系を持つ場合} \end{cases} \quad (11)$$

ここで、 $\hat{y}_i$ は $i$ 番目の制御変数の予測値であることを表し、 $T_i^{\text{opt}}$ は $i$ 番目の制御変数の最適化時間を表す。 $k$ は現在の制御周期を表す。

原点からの差分は次式のようなベクトルで表す。ダッシュがついている方は過渡区間の値、ついてない方は擬似定常区間の値である。

$$\Delta \mathbf{u}' = [\Delta u'_1, \dots, \Delta u'_m]^T, \Delta \mathbf{u} = [\Delta u_1, \dots, \Delta u_m]^T \quad (12)$$

$$\Delta \mathbf{y}' = [\Delta y'_1, \dots, \Delta y'_n]^T, \Delta \mathbf{y} = [\Delta y_1, \dots, \Delta y_n]^T \quad (13)$$

これらをまとめたベクトルを  $\mathbf{x}$  で表す。

$$\Delta \mathbf{x} = \begin{pmatrix} \Delta \mathbf{u} \\ \Delta \mathbf{y} \end{pmatrix}, \mathbf{x}^0 = \begin{pmatrix} \mathbf{u}^0 \\ \mathbf{y}^0 \end{pmatrix}, \mathbf{x} = \mathbf{x}^0 + \Delta \mathbf{x} \quad (14)$$

$\Delta \mathbf{u}$  と  $\Delta \mathbf{y}$  の関係は以下のように表される。

$$\begin{pmatrix} \Delta \mathbf{y}' \\ \Delta \mathbf{y} \end{pmatrix} = \begin{pmatrix} G^S + G^I & 0 \\ G^I & G^S \end{pmatrix} \begin{pmatrix} \Delta \mathbf{u}' \\ \Delta \mathbf{u} \end{pmatrix} \quad (15)$$

ここで、 $G^S$ は積分要素を持たない部分の定常ゲイン行列、 $G^I$ は積分要素を持つ部分の疑似定常ゲイン行列であり、 $j$ 番目の操作変数から $i$ 番目の制御変数への伝達関数を  $G_{i,j}(s)$ 、むだ時間を  $L_{i,j}$  とすると、それぞれの $i$ 行 $j$ 列の要素は以下で表される。

$$G_{i,j}^S = \begin{cases} G_{i,j}(0), & \text{積分要素無し} \\ 0, & \text{積分要素有り} \end{cases} \quad (16)$$

$$G_{i,j}^I = \begin{cases} 0, & \text{積分要素無し} \\ S_{i,j}(T_i^{\text{opt}} - L_{i,j}), & \text{積分要素有り} \end{cases} \quad (17)$$

$$S_{i,j} = \lim_{s \rightarrow 0} s G_{i,j}(s) \quad (18)$$

式 (10) ~ (18) のように定義すると、最適化問題は、

$$\underline{u}_i - u_i(k-1) \leq \Delta u_i, \Delta u_i' \leq \bar{u}_i - u_i(k-1) \quad (19)$$

$$\underline{y}_i - y_i^0 \leq \Delta y_i, \Delta y_i' \leq \bar{y}_i - y_i^0 \quad (20)$$

$$S \Delta \mathbf{u} = 0 \quad (21)$$

を制約とした上で、評価関数

$$J(\mathbf{x}) = \mathbf{x}^T H \mathbf{x} + \mathbf{c}^T \mathbf{x} \quad (22)$$

を最小化する問題として定義できる。ただし、 $S$  は  $S_{i,j}$  を  $i$  行  $j$  列の要素とする行列である。

#### 4.2 数値シミュレーション例

本節では、積分要素を含む制御対象を対象とした数値シミュレーションを例示する。まず、制御対象の伝達関数を示す。

$$G_{\text{CV/MV}}(s) = \begin{pmatrix} \frac{1}{4s+1} & \frac{0.5}{10s+1} \\ -\frac{0.01}{s(4s+1)} & \frac{1}{25s^2+10s+1} \end{pmatrix} \quad (23)$$

CV2を見ると、MV1からの伝達関数には積分要素を含むが、MV2からの方は含まない。すなわち、積分系とそう

でない系が混在した系になっている。

最適化の評価関数は次式のように設定した。MV2とCV2の最大化が最適化目的である。

$$V = -\text{MV2} - \text{CV2} \quad (24)$$

CV1の上限と下限は-0.5と0.5に、CV2は-0.1と0.1に、MV1とMV2は-1と1に設定した。初期値はCV1が0、CV2が0.1、MV1が0、MV2が0.8である。CV2の傾きの初期値は0とする。

MV2には最大化の余地があるが、CV2の初期値が既に上限であり、そのままでは値を増やせない。ここでは、MV1とCV2の間にある積分系を利用してMV2の最大化を可能にする。

図10にシミュレーション結果を示す。開始後、MV1を増やすことでCV2を押し下げ、MV2を増やす余地を作っている。この余地を利用してMV2を操作し、上限への到達に合わせてMV1を元に戻す。MV1が元に戻ればCV2の傾きも0になるため、CV2が発散することは無い。以上のように、積分要素に掛かる制御変数の傾きを一時的に許容することで、他の変数をさらに最適化することを可能にしている。

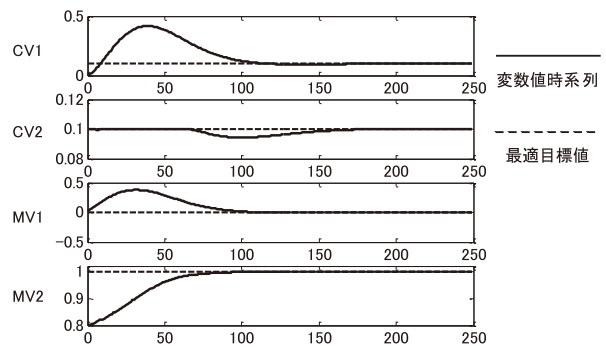


図 10 積分要素を持つプロセスの最適化

### 5. 多装置間最適化との連携

多装置間最適化は、複数の装置が接続されているプラントにおいて、プラント全体の省エネルギーや省CO<sub>2</sub>、生産量最大化や原単位改善を行うためのものである。装置単位の最適化だけではプラント全体として最適化できないような場合に利用される。本章では、多装置間最適化のロジックと連携した場合のMPCコントローラの動作について簡単に説明する。また、数値例によってその特長を示す。

図11は多装置間最適化を行う場合のシステム構成である。MPCは各装置のプロセス値を収集し、非制御定常値を予測する。予測結果は多装置間最適化ロジックに送信される。多装置間最適化ロジックは非制御定常値、ゲイン情報、装置間の接続情報を考慮し、与えられた評価関数を最適にする目標値(以下、グローバル最適目標値と呼ぶ)を求める。グローバル最適目標値は各MPCに送信され、MPCは担当する装置がその目標値へと向かうように操作量を決める。このように、定常最適目標値を決める多装置間

最適化ロジックと、その目標値へと動的制御を行うMPCとが役割を分担する分散型システムとなっている。

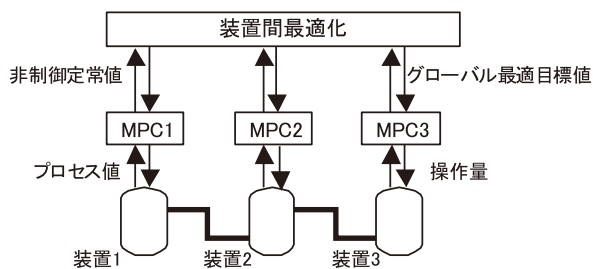


図 11 多装置間最適化ロジックとの接続

多装置間最適化実行時のMPCの制御演算だが、基本的には図1の構成で行われる。ただし、定常目標値は上位で既に決められているので、目標値最適化部は不要となる。この部分は、定常的なグローバル最適目標値を修正して、過渡的なローカル目標値を決める目標値修正部に置き換わる。また、各MPCは上流装置の変数値を外乱変数として扱う。

目標値修正部が必要となる主な理由は2つあるが、いずれも装置間の遅延に起因する。

多装置間最適化ロジックが算出したグローバル最適目標値は、装置全体が定常状態となり、上流から下流まで変化が行き渡った状態の目標値である。ところが過渡状態では、上流装置の変数値がグローバル最適目標値に収束しているとは限らない。多装置間最適化の解は上流装置もグローバル最適目標値に収束していることを前提としているため、各装置でその目標値に近づけようとしても、過渡状態では可能解が得られない場合がある。このような場合、目標値の修正が必要となる。

また、可能解ではあっても、一時的に多装置間最適化の目的と相反する目標値になる場合もある(機会損失)。この状態は最適化が進むにつれて解消されるが、一時的とはいえ生産量低下やエネルギー増加をもたらすので、発生しないに越したことはない。SORTiAでは多装置間最適化とMPCコントローラが連携して機会損失を抑制するようになっており、過渡状態でも最適状態に近づけるよう制御する。

#### 数値例

機会損失の抑制について、2つのMPCからなる簡単な例で示す。制御対象の構成図を図12に示す。上流側のCTL1は制御変数と操作変数が1つずつ、下流側のCTL2は制御変数、操作変数、外乱変数が1つずつのコントローラである。CTL1のCV1がCTL2のDV1に接続されており、CTL1のMV1を操作すればCTL2にも影響する。

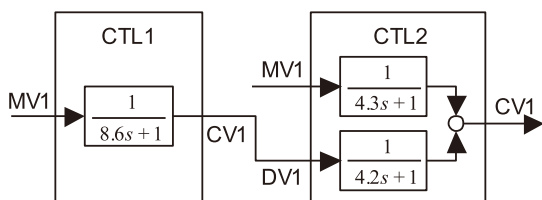


図 12 多装置間最適化シミュレーションの設定

図13にシミュレーションの時系列データを示す。実線が変数の値、破線が変数のグローバル最適目標値である。制御変数の下限は全て0、上限は1である。操作変数の下限は全て0、上限は2である。シミュレーション開始時の最適化目的はCTL2のCV1の最大化であり、どの変数も既に最適目標値に到達して定常状態となっている。

ここで、時刻20で最適化目的を追加し、CTL1のMV1の最大化も行うようにする。CTL1のMV1を大きくするためには、CTL2のMV1を小さくしてCTL2のCV1が上限を違反しないようにする必要がある。ここで、CTL2のMV1を小さくすることは可能なので、可能解は存在する。しかし、だからといってMV1をすぐに減らすと、DV1が増加するより先にCTL2のCV1へ影響が出るため、一時的に値が減少する。最適化目的にはこの変数の最大化も含まれるので、過渡的とはいえ目的に反する動きとなり、最適化の機会を損失していることになる。

SORTiAの場合は図示したように、CTL1のCV1の増加に合わせてCTL2のMV1が操作される。このような操作により、CTL2のCV1は最適状態を維持することができ、過渡的な機会損失を回避できる。

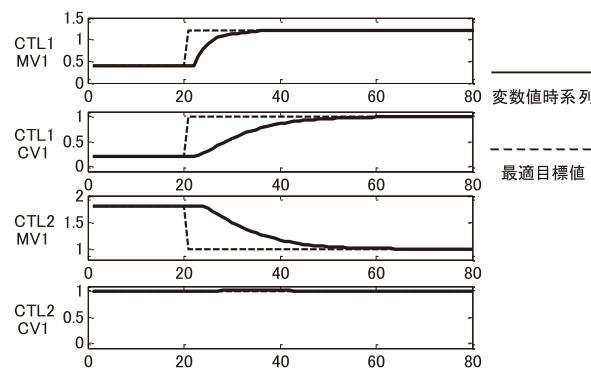


図 13 多装置間最適化における機会損失抑制

## 6. おわりに

本論文では、高度制御ソリューションSORTiAで使われている制御、最適化技術について説明した。

SORTiAの中核であるSORTiA-MPCは、チューニングが容易という特長を持つモデル予測制御コントローラである。その特長は、制御器の3自由度構成、参照軌道と自由応答を利用した振る舞いの制御、制御演算の自動設定、正規化された調整パラメータによるものである。

また、積分要素を持つプロセスの最適化を可能にする定常目標値最適化アルゴリズム、過渡状態の制御性能劣化を抑制する多装置間最適化手法についても説明し、シミュレーションでその有効性を示した。

文献(4)では、SORTiA-MPCを石油精製プラントに適用した事例や、SORTiA-LQPをエチレン装置のモデルに適用したシミュレーション結果を紹介しているので、こちらも参照されたい。



<参考文献>

- (1) S. Joe Qin, Thomas A. Badgwell: A survey of industrial model predictive control technology, Control Engineering Practice, 2003, Vol. 11, pp. 733-764, Elsevier
- (2) Jan M. Maciejowski (足立修一, 菅野政明訳): モデル予測制御 — 制約のもとでの最適制御 —, 2005, 東京電機大学出版局
- (3) 清水 洋: 多変数モデル予測制御活用のポイント, 計測と制御, Vol. 43, No. 9, pp. 665-668, 計測自動制御学会
- (4) 岩本聡一: 現場で価値を創る高度化ソリューション, 計装, 2013, Vol. 56, No. 7, pp. 61-66, 工業技術社

<商標>

SORTiAは, アズビル株式会社の商標です。

<著者所属>

田原 鉄也 技術開発本部商品開発部  
藤江 真也 技術開発本部商品開発部