

MATLAB EXPO 2023  
2023/05/31

# 有機膜形成装置の 温度制御における Model-Based Design適用事例

■発表者  
芝浦メカトロニクス株式会社  
技術本部 研究開発グループ  
杉内 佑輔

- 会社紹介
- 導入背景と課題
- 温度制御におけるモデルベース開発適用事例
- 開発実施結果とまとめ

- 会社紹介
- 導入背景と課題
- 温度制御におけるモデルベース開発適用事例
- 開発実施結果とまとめ

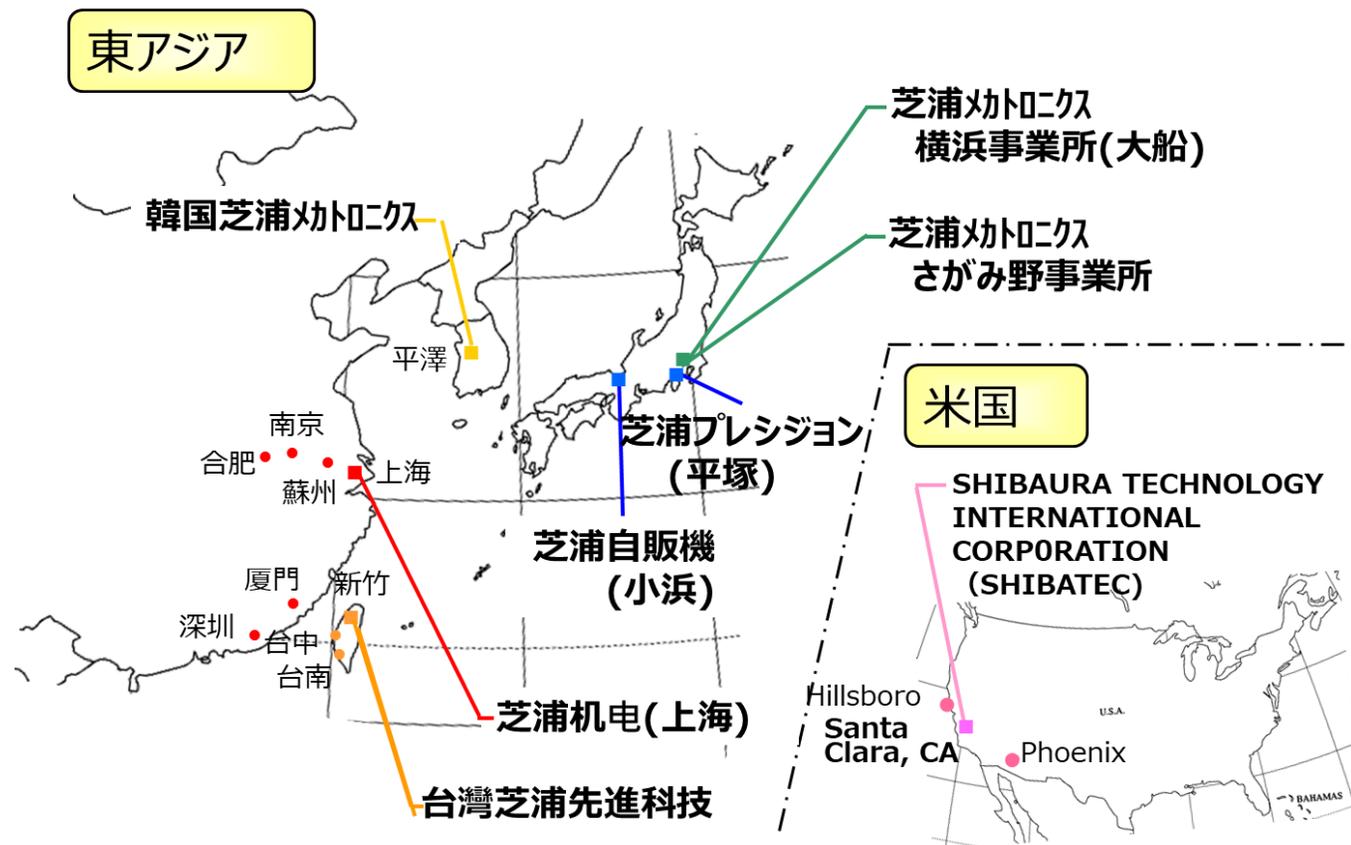
## 会社概要



- 商号 : 芝浦メカトロニクス株式会社
- 設立 : 1939年(昭和14年)10月12日
- 本社 : 神奈川県横浜市栄区
- 資本金 : 67億円(2023年3月末時点)
- 売上高 : 連結610億円(2022年度)
- 営業利益 : 連結109億円(2022年度)
- 従業員 : 連結1,221名(2023年3月末時点)

Smart Solutions & Services for Your Manufacturing

### 国内・海外拠点



## ファインメカトロニクス事業部

- 半導体製造装置（前工程）
  - ・ウェーハ洗浄／エッチング装置
  - ・マスク洗浄／エッチング装置
- F P D 製造装置（前工程）
  - ・ウェットプロセス装置
  - ・インクジェット装置

### 横浜事業所（本社）



## メカトロニクスシステム事業部

- 半導体製造装置（後工程）
  - ・ダイボンダ
  - ・フリップチップボンダ
- F P D 製造装置（後工程）
  - ・アウターリードボンダ
- 真空応用装置

### さがみ野事業所

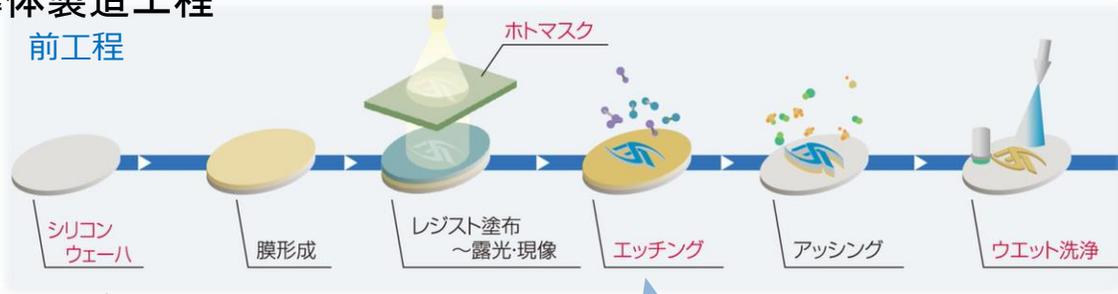




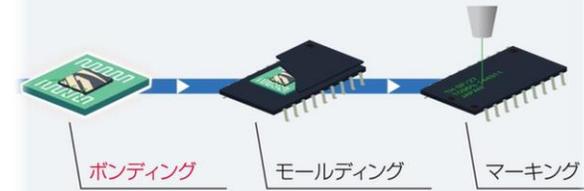
超高精度  
ハイブリッドボンダ



半導体製造工程  
前工程



後工程



# 主要製品群 (真空応用)



CCS-1300/2100/2800  
用途：カバーガラス (AR)  
ヘッドアップディスプレイ (AR、増反射) 等



stella (枚葉式)  
用途：CD、DVD、BD等



BM-600/650 (枚葉式)  
用途：車載/親水ミラー等



BM-700 (枚葉式)  
用途：エンブレム、内装品等



CFS-36PC  
用途：自動車外装 (バンパー、ランプリフレクタ等)  
自動車内装 (メータリング、装飾品等)



プラコートIV (枚葉式)

自動車産業

光学産業

電子産業

情報産業

真空技術

半導体産業

研究開発  
その他産業



CFS-12P-90H  
用途：LED、電子部品



!-Miller  
用途：電子部品、LED、バイオ、  
材料等の研究開発



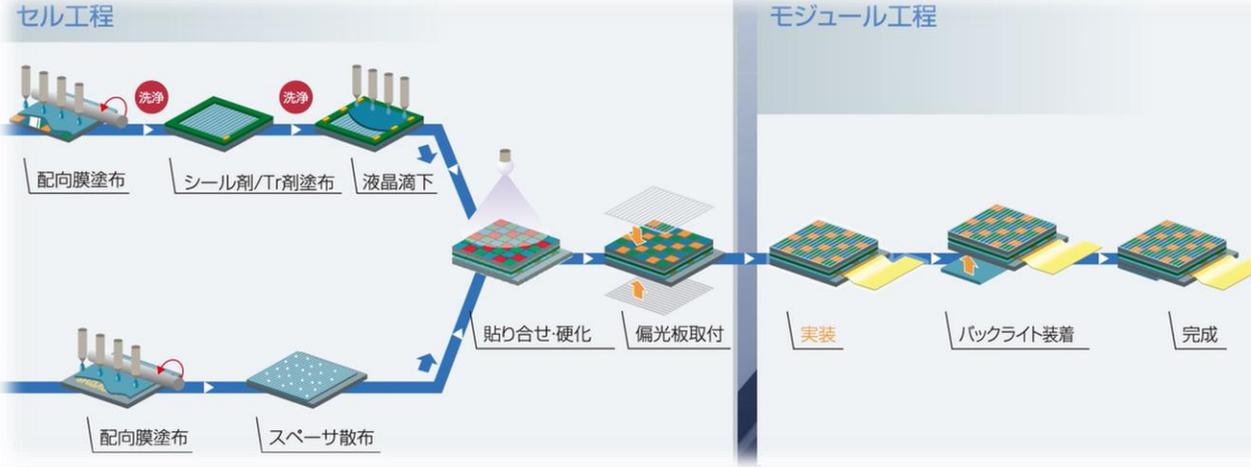
SWN-5000  
用途：パワーデバイス、LED



BM-1400E (枚葉式)  
用途：電子部品

# SHIBAURA 主要製品群 (フラットパネルディスプレイ関連) 他

## ディスプレイ製造工程



前工程向け製品群  
(ウェット工程)



後工程向け製品群  
(アウターリードボンダ)



フレキシブルOLED向け真空焼成炉

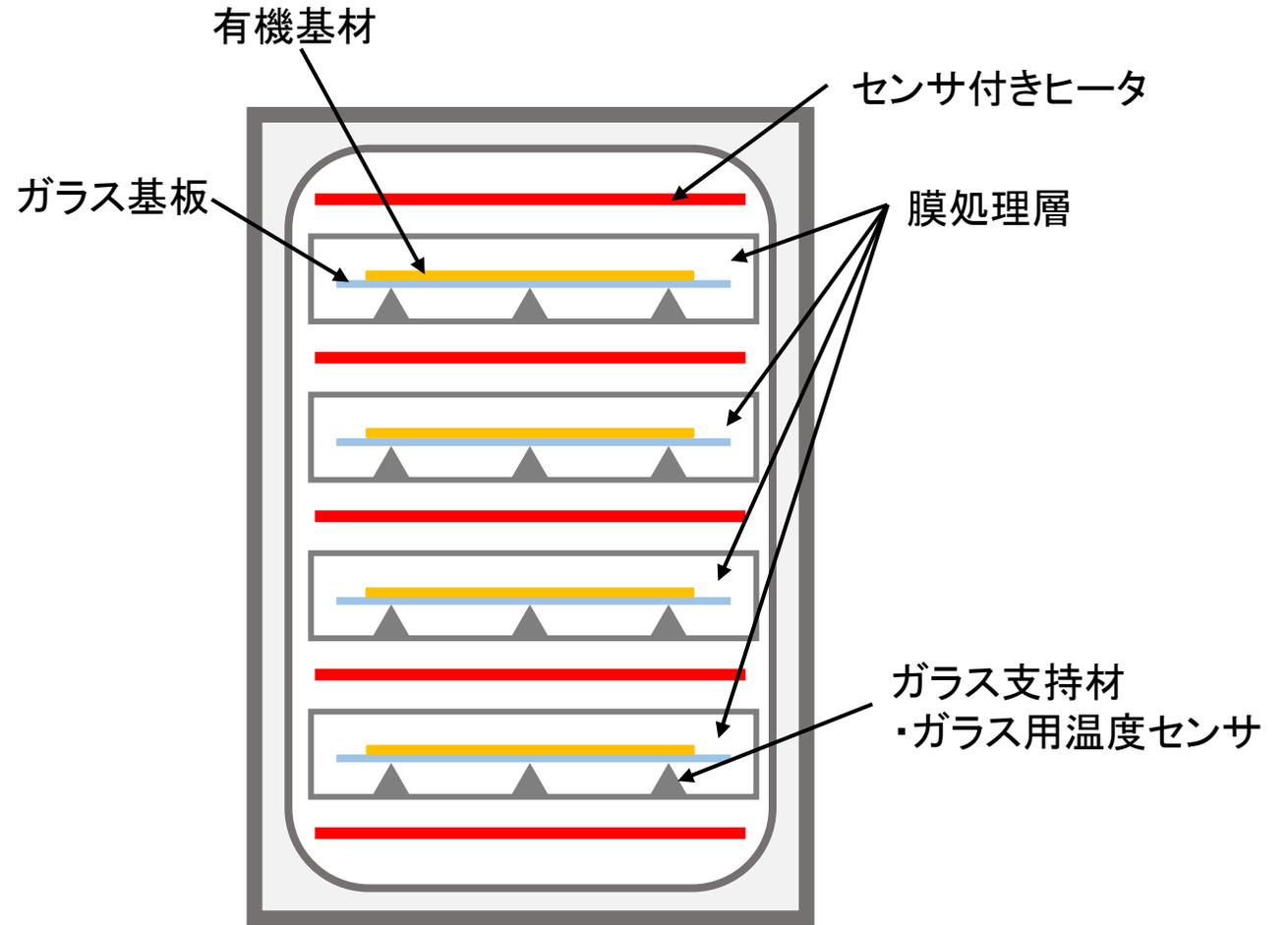
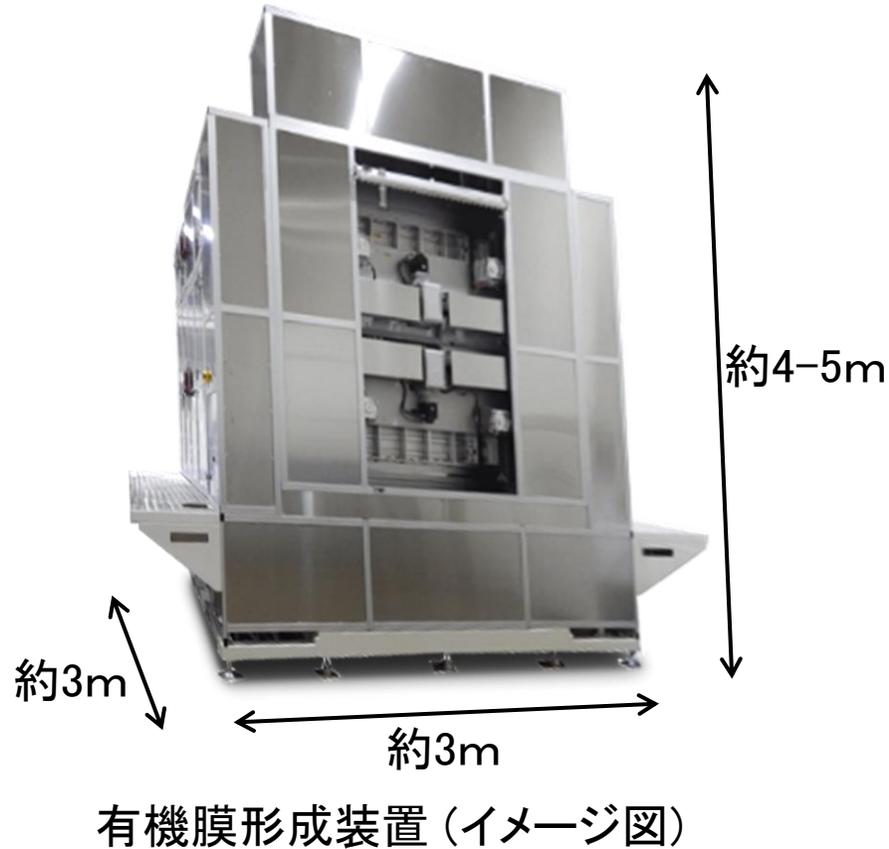


インクジェット銀剤印刷装置

- 会社紹介
- 導入背景と課題
- 温度制御におけるモデルベース開発適用事例
- 開発実施結果とまとめ

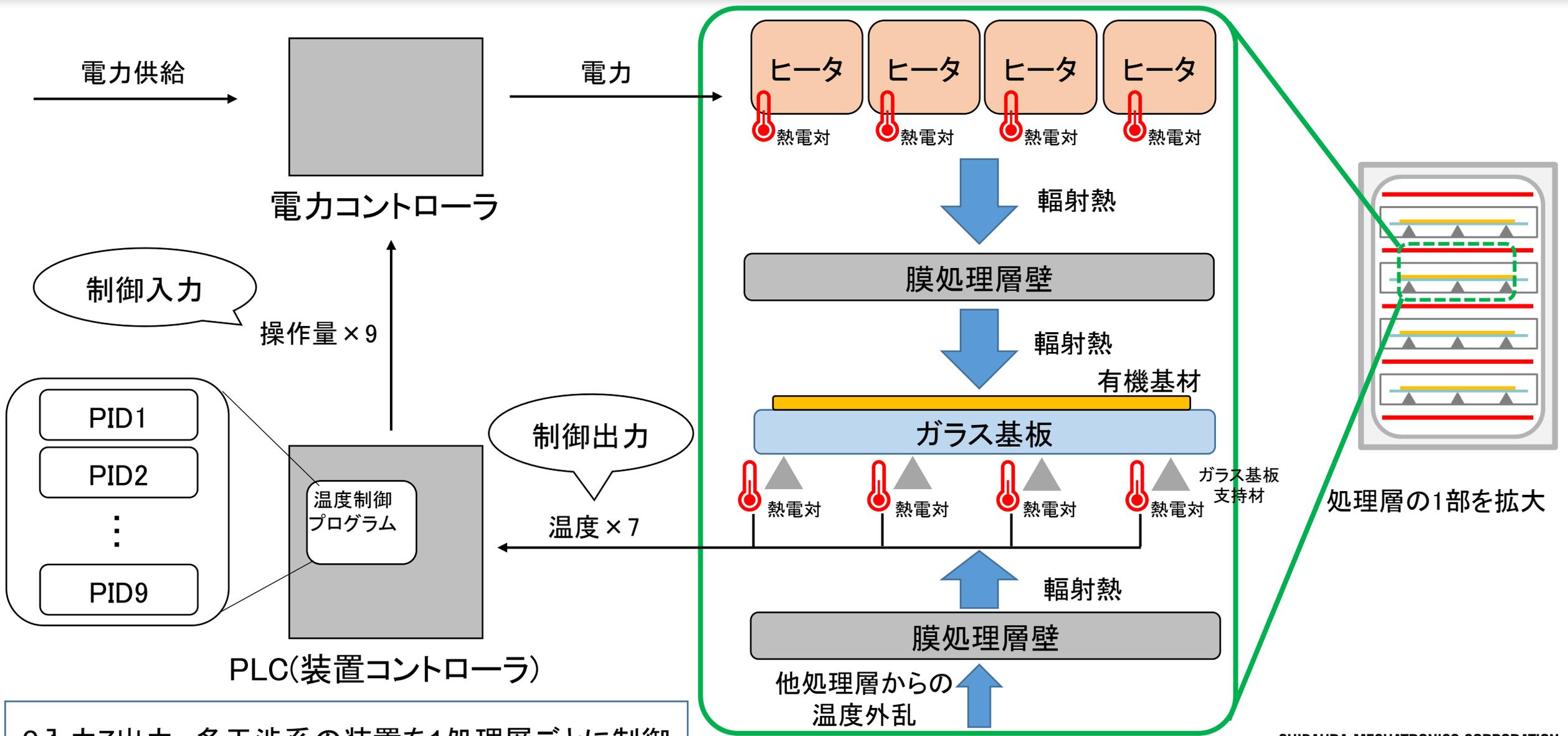
# 有機膜形成装置の概要

## 有機ELディスプレイ基材を作製する加熱炉



有機膜形成装置 内部模式図

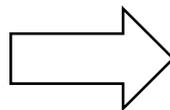
# 有機膜形成装置の温度制御方式



9入力7出力、多干渉系の装置を1処理層ごとに制御

## ○ 制御設計の課題 (What to Do)

- ▶ パラメータ調整の時間を短くする
- ▶ 調整ルールを単純にする
- ▶ 実機を使わずに調整する



- 手戻りを少なく
- 設計時間を短縮

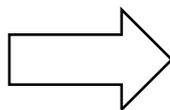
## ○ 制御設計の施策

シミュレーションを使った  
スマートな開発  
⇒ **モデルベース開発**

※ モデルベース開発 (Model-Based Design)

## ○ 装置の制御上の課題 (What to Do)

- ▶ オーバーシュートさせない制御
- ▶ 対象の伝熱特性を考慮した制御
- ▶ 多入力多出力が扱える制御



- より高度な制御方式の適用

## ○ 装置制御の施策

現代制御理論の導入  
⇒ **モデル予測制御**

## ○ モデルベース開発とは

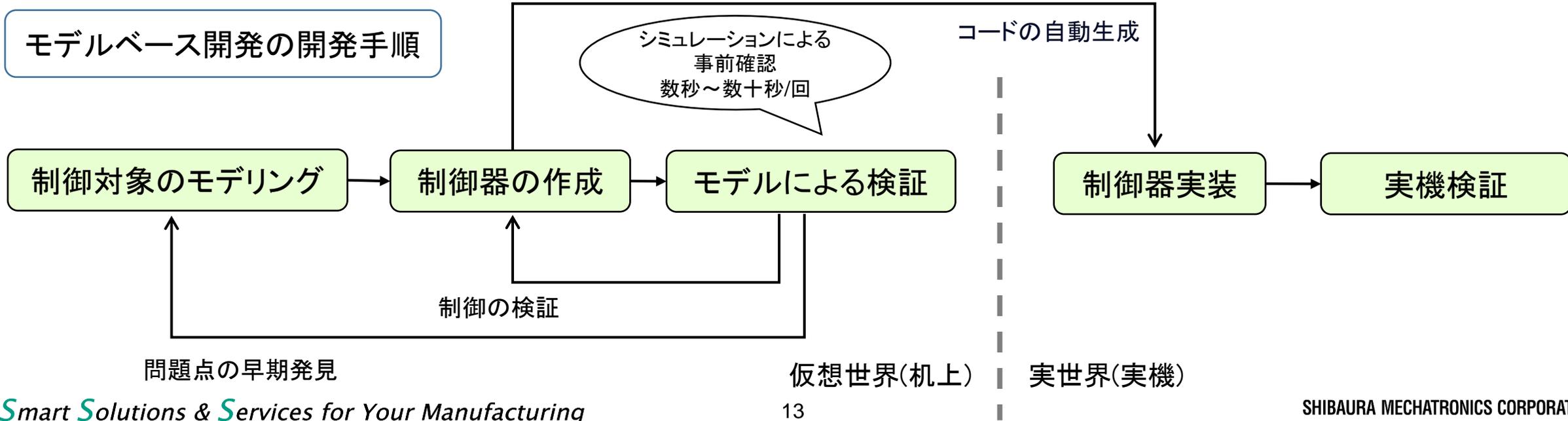
制御器や制御対象のシミュレーションモデルを作成し、机上で検証を行いながら開発を進めていく手法

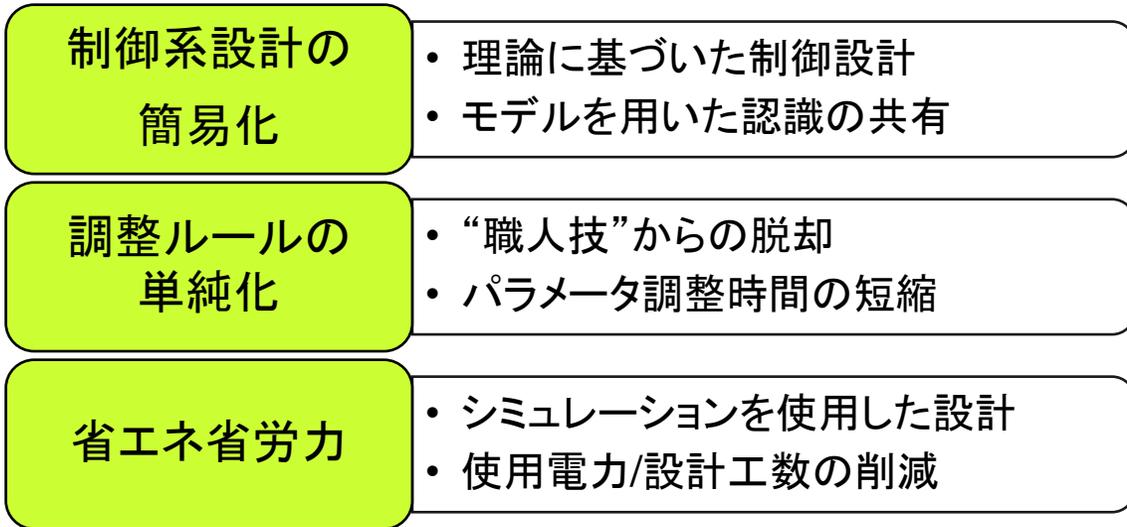
[モデルベース開発の利点]

- ・ 机上で制御の検証
- ・ 問題点の早期発見により、開発手戻りの削減
- ・ コードの自動生成が利用可能

自動車業界以外でも導入事例が多数

⇒ 効率のよい開発ができると注目

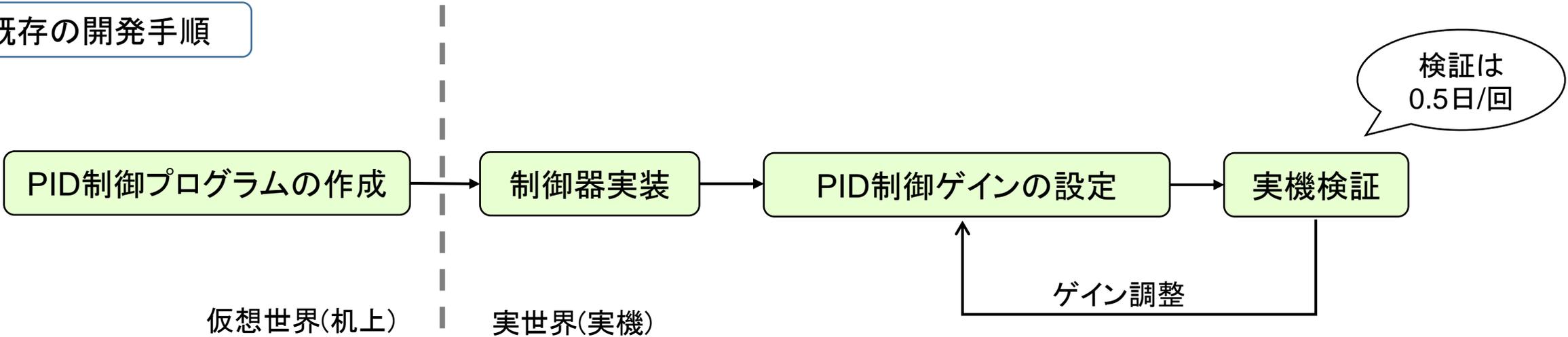




現代制御も導入してより高度に制御をしたい

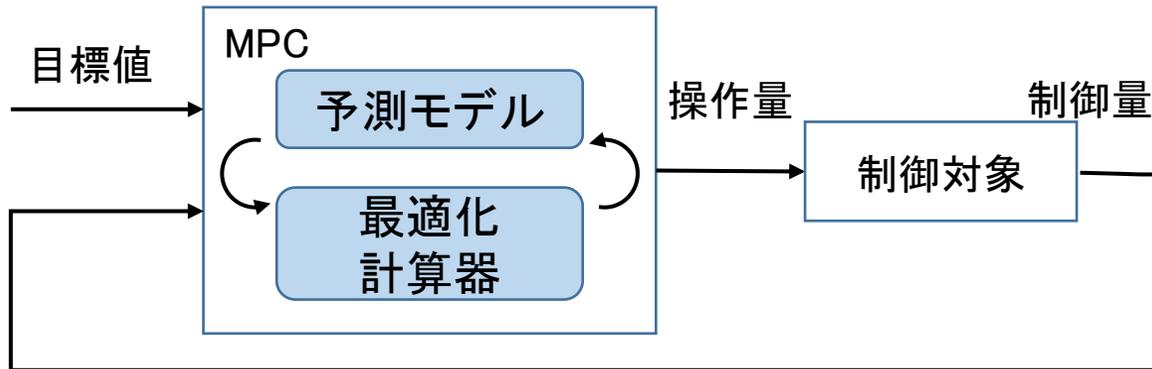
⇒ モデル予測制御を導入

既存の開発手順



## ○ MPC とは

制御対象の未来の動きを「予測」し、  
計算による「最適な」操作にて制御する手法



### [ MPCの利点 ]

- 目標値を先読みし、有限時間先までの動作を予測 (Look-Ahead機能、MPC内部の予測モデルを利用)
- 大規模かつ複雑な対象に対応 (制御対象モデルに基づく合理的な制御)
- 制約条件を守った制御 (リアルタイムの最適化計算)

⇒ 高性能かつ信頼性のある制御

### 最適化評価関数

$$\Sigma w_y \times (\text{追従誤差})^2$$

$$J = + \Sigma w_{\Delta u} \times (\text{操作量の変化率})^2$$

$$+ \Sigma w_u \times (\text{操作量の大きさ})^2$$

どれを重要視するかで  
重み(w)の値を調整可

制約条件を考慮して  
最適化が可能

ただし、

$$\text{下限値} \leq \text{制御量} \leq \text{上限値}$$

$$\text{下限値} \leq \text{操作量変化率} \leq \text{上限値}$$

$$\text{下限値} \leq \text{操作量} \leq \text{上限値}$$

# MPCに期待するメリット

多入力多出力が扱える

- 温度の多干渉性を考慮した制御
- 制御器の削減

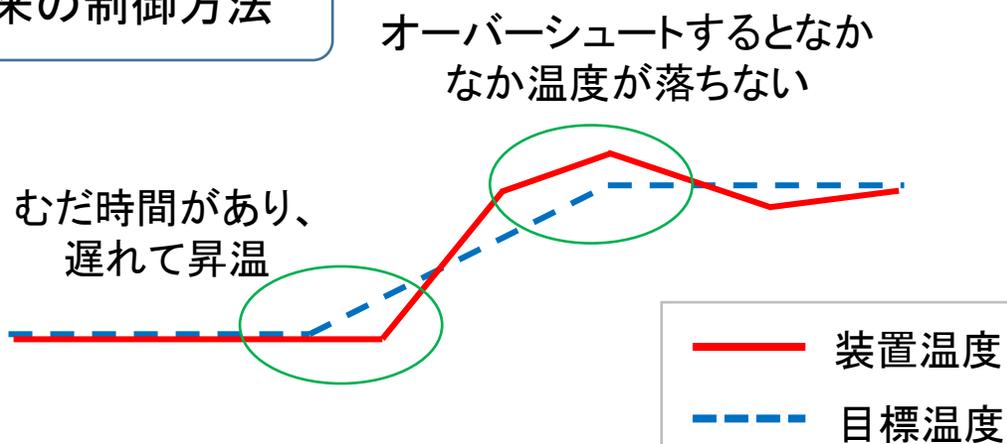
目標値の先読み

- 目標値先読み機能(Look-Ahead)の活用
- フィードフォワード制御による追従性の向上

制約の順守

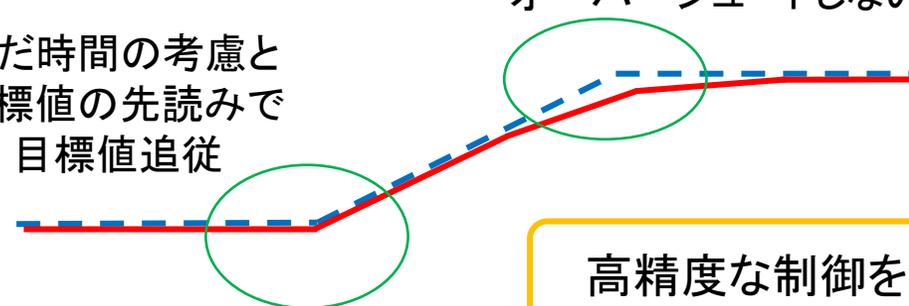
- ハードウェア制約の範囲内で制御可能
- PIDゲインのようなパラメータ調整が不要

従来の制御方法

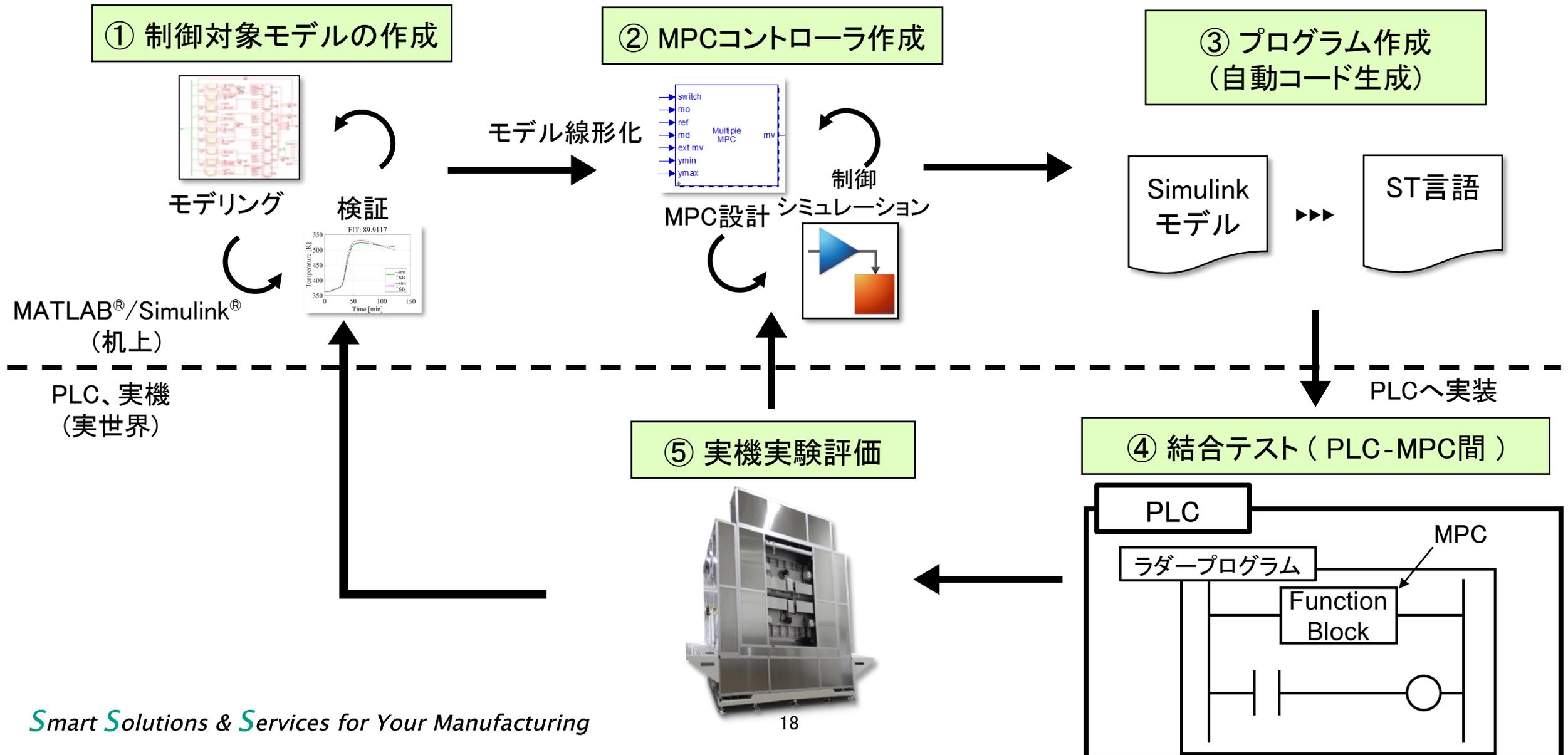


MPCを導入した制御方法

むだ時間の考慮と目標値の先読みで目標値追従

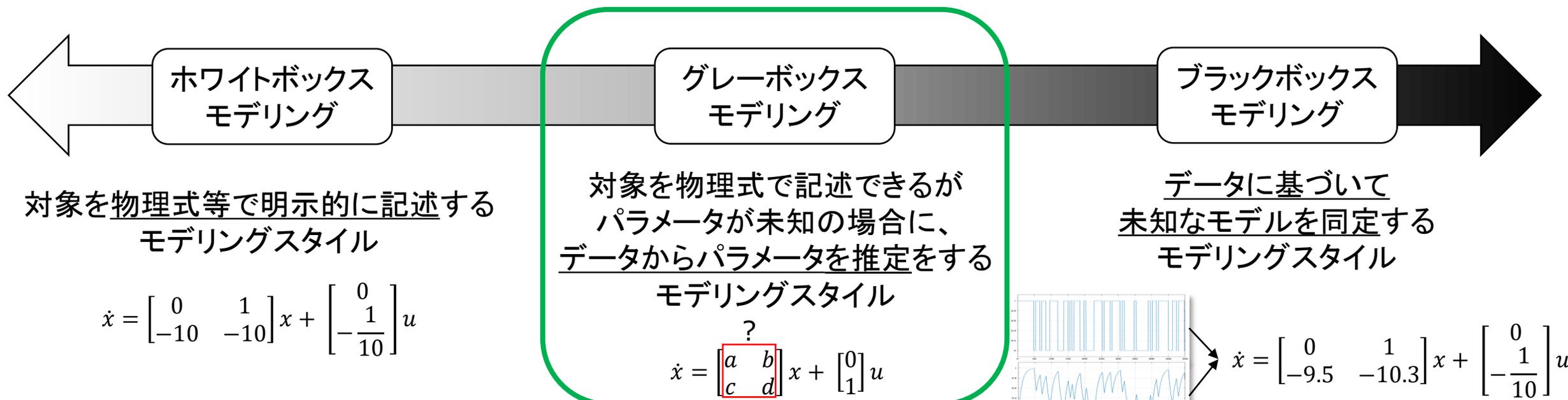


- 会社紹介
- 導入背景と課題
- 温度制御におけるモデルベース開発適用事例
- 開発実施結果とまとめ



モデル作成では**グレーボックスモデリング**を採用

- ▶ モデルが直感的に理解しやすい
- ▶ 装置の構造が既知だが、未知のパラメータが存在
- ▶ 実機と同じ動きをする非線形なモデルを作成したい



## グレーボックスモデリング : ① モデル構造作成

- ▶ 実機の構造からモデルを作成
- ▶ 制御対象は装置内の1層  
→ 1層ごとにモデリング

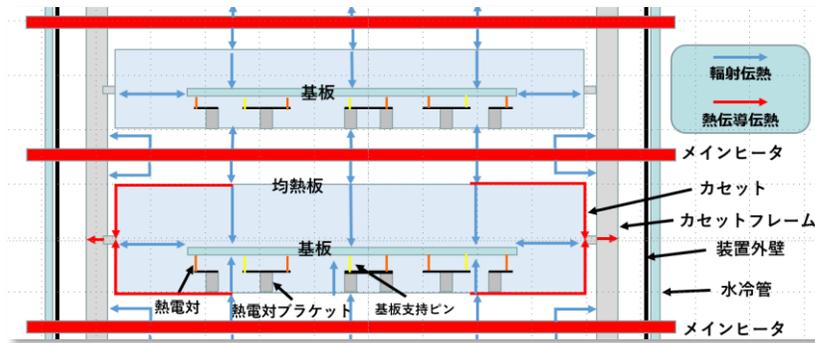
装置の要素ごとに伝熱式を記述  
↓  
Simulinkで結線、制御対象モデル作成

実機を観察 (設計図面/目視)

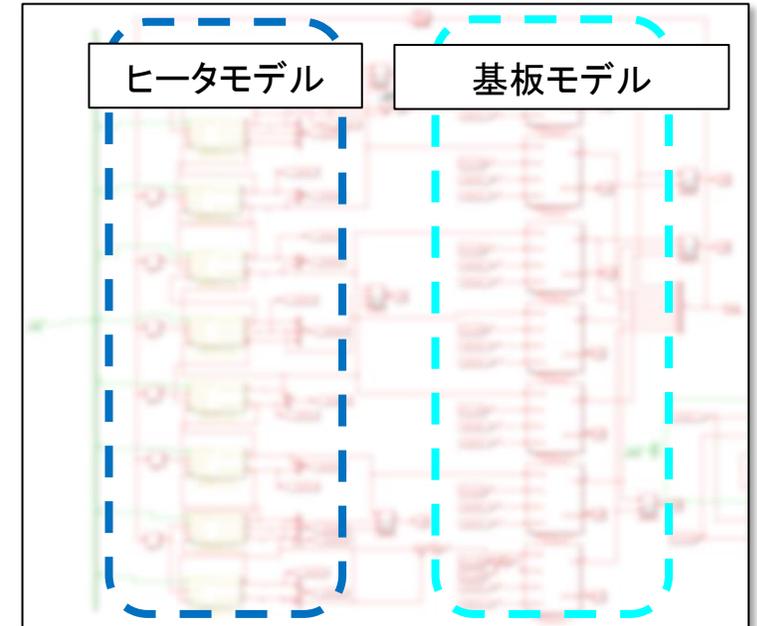


実機

- 伝熱経路を図示
- 伝熱式を立式



実機伝熱模式図



制御対象モデル

構造が同じパーツ(ヒータなど)は  
**ライブラリンク**を使用して工数削減

## グレーボックスモデリング：② パラメータ探索



伝熱式における未知のパラメータをリストアップ



輻射の  
伝熱方程式

$$\text{熱流束}_{BA} = \text{物理定数} * \frac{(\text{温度}_B^4 - \text{温度}_A^4)}{\left(\frac{1}{\text{放射率}_B} + \frac{1}{\text{放射率}_A} - 1.0\right)}$$

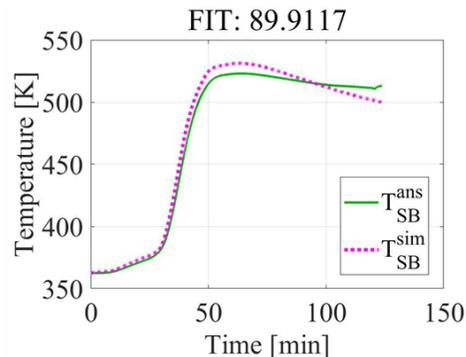
$$\text{温度変化量}_{BA} = \frac{\text{形態係数}_{BA} * \text{面積}_A * \text{熱流速}_{BA} * \text{制御周期}}{\text{熱容量}_A}$$

探索パラメータの一例  
伝熱方程式中の放射率と形態係数は  
事前調査では求められない

※ 高温・真空のため、放射率の測定が困難  
※ 内部構造が複雑なため、形態係数の決定が困難



モデルの応答と実機の波形が一致するようなパラメータを探索

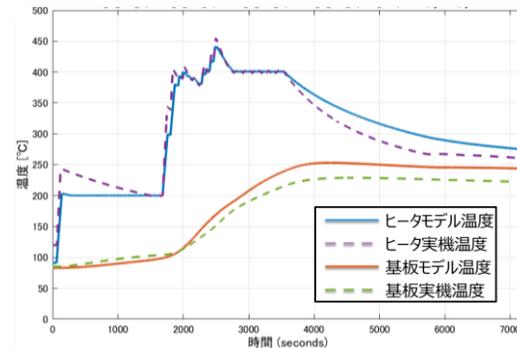


パラメータ同定結果

→ パラメータ探索には  

- Optimization Toolbox™
- Global Optimization Toolbox

 を使用し、自動探索

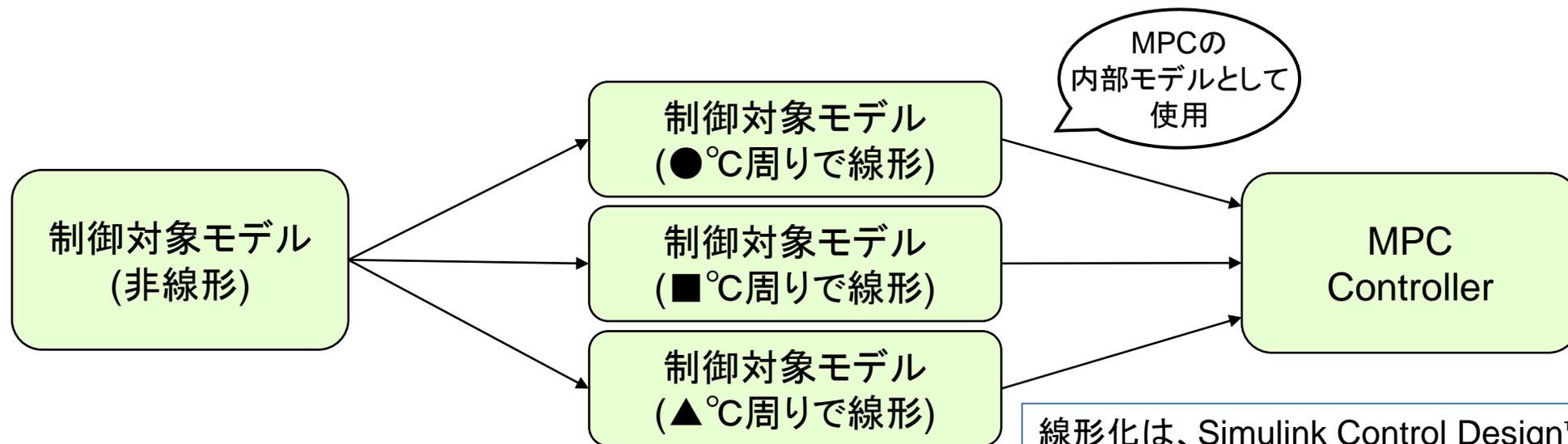


モデル検証

→ パラメータ探索の結果を用いたモデルと、  
実機の動作結果を比較して  
モデルの確からしさを検証した

MPCの内部モデルは**制御対象モデルを“線形化”**して作成

- ▶ 使用温度範囲が広く、温度帯によってダイナミクスが変化
- ▶ 非線形性が強いため、線形モデルを複数作成し、MPCへ適用
- ▶ シミュレーションを通じ、複数の代表的な温度点において線形化を実行



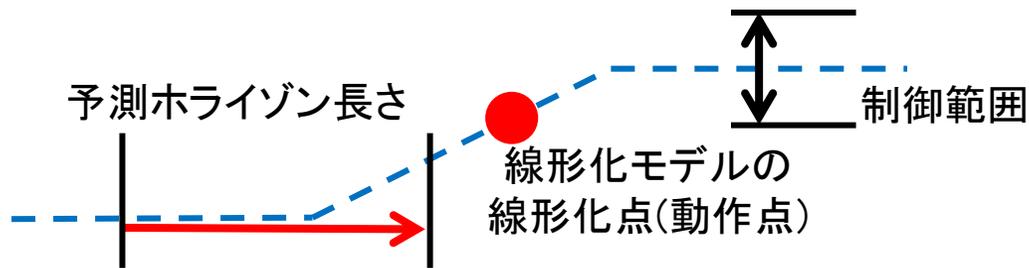
線形化は、Simulink Control Design™ の機能を使用

## Multiple MPC にて有機膜形成装置の制御器を設計

Model Predictive Control Toolbox™を使用

※ Multiple MPC : 別名ゲインスケジューリングMPC  
複数の線形MPCを切り替えて動作する

### ① 設計要件・制約条件を確認



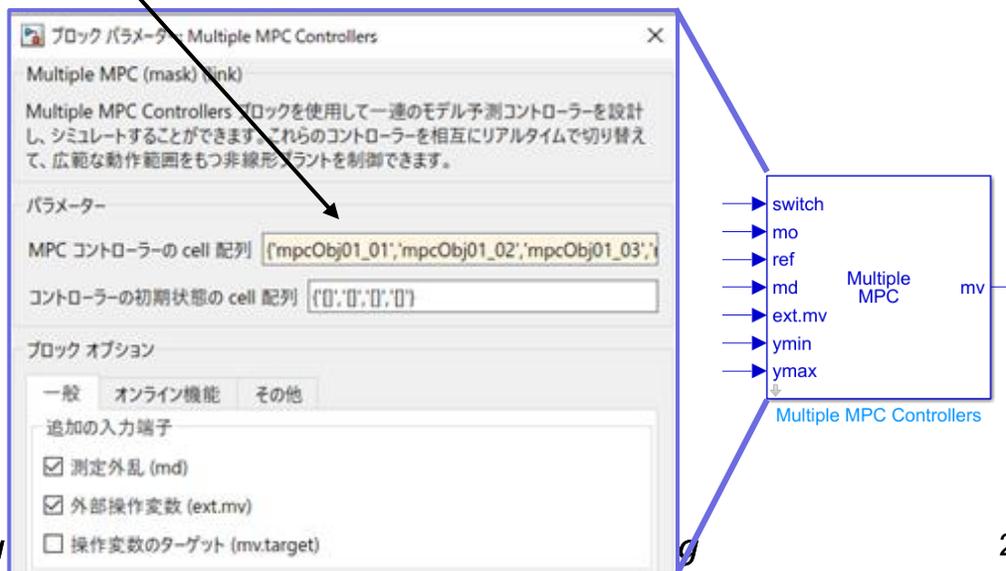
### ② 設計要件・制約条件に沿ったMPC設定をスクリプトベースで記述

```

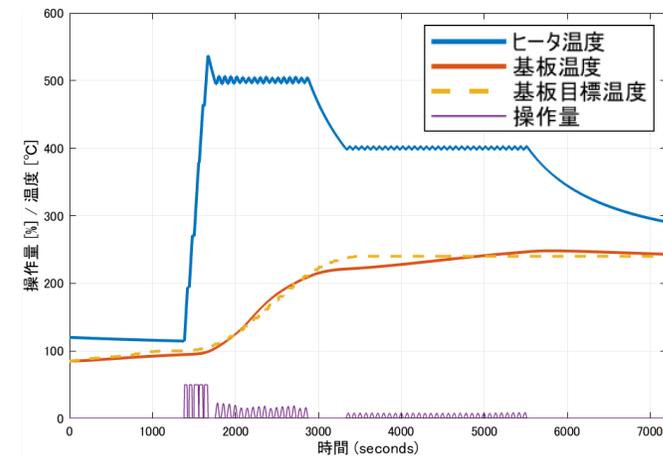
MPCオブジェクトへ設定を追記していく
% Make the MPC Object
mpcObj=mpc(Model,ts,predHriz,ctrlHriz);

%% specify nominal values for inputs and outputs
mpcObj.Model.Nominal.U = 100;
mpcObj.Model.Nominal.Y = 100;
    
```

### ③ 作成したMPCオブジェクトをSimulinkブロックの中へ記述



### ④ 動作をシミュレーションにて確認後、実装



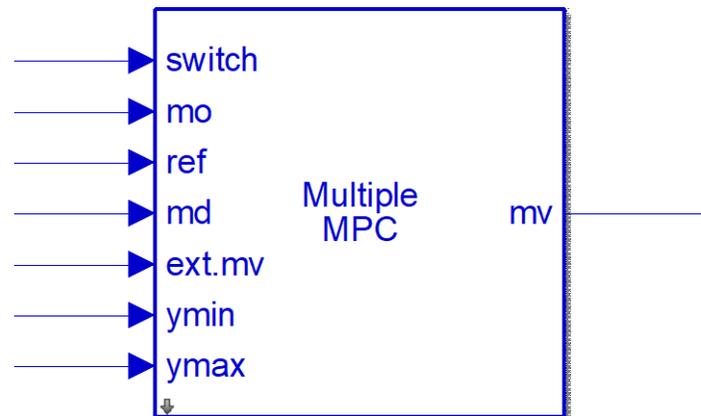
## MPCのPLCへの実装はSimulink PLC Coder™ を使用

- ▶ 設計したMPCをST言語へ変換(自動生成)
- ▶ 装置制御回路(ラダー)へMPC適用が可能に

当社における  
MATLAB/Simulink導入の  
決め手の一つ

### ○ STEP1 MPCブロックのSTコード変換

- ▶ MPCブロックをサブシステム化後、Atomicサブシステムへ変換
- ▶ 使用するIDEに基づいてSTコードを吐出



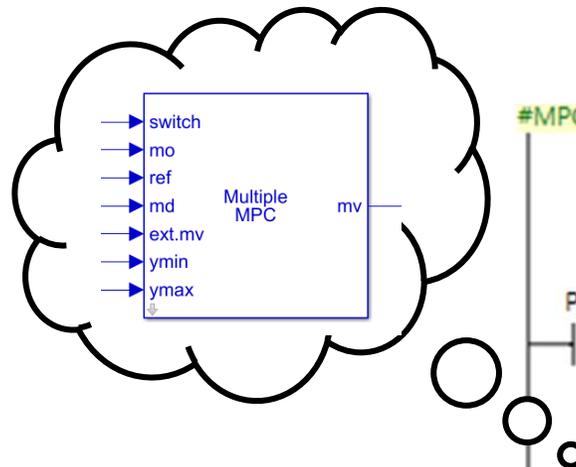
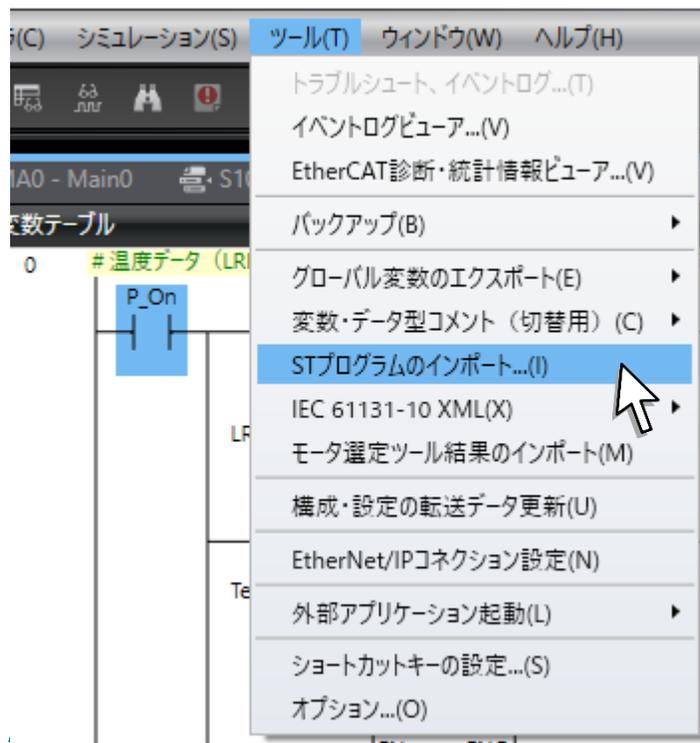
```

File: MPC_control.st
1  (*
2  *
3  * File: MPC_control.st
4  *
5  * IEC 61131-3 Structured text (ST) code generated for subsystem "MPC_control/MPC_Subsystem"
6  *
7  * Model name          : MPC_control
8  * Model version       : 1.1
9  * Model creator      :
10 * Model last modified by :
11 * Model last modified on : Fri Apr 14 16:44:37 2023
12 * Model sample time   : MPC_control/MPC_Subsystem
13 * Subsystem name     : MPC_control/MPC_Subsystem
14 * Subsystem sample time : 0.2s
15 * Simulink PLC Coder version : 3.5 (20221b) 14-May-2021
16 * Target plugin version : 2.20
17 * ST code generated on  : Fri Apr 14 16:45:10 2023
18 *
19 * Target IDE selection : OMRON Sysmac Studio
20 * Test Bench included  : No
21 *
22 *)
23
24 FUNCTION_BLOCK MPC_Subsystem
25 VAR_INPUT
26   mMethodType: SINT;
27   moorc: LREAL;
28   ref: LREAL;
29 END_VAR
30 VAR_OUTPUT
31   mv: LREAL;
32 END_VAR
33 VAR
34   Memory_PreviousInput: ARRAY [0..3] OF BOOL;
35   Last_PreviousInput: ARRAY [0..2] OF LREAL;
36   MathFunction2: LREAL;
37   MathFunction1: LREAL;
38   MathFunction: LREAL;
39   last_mv_UPDATE: LREAL;
40   e_a: ARRAY [0..3] OF SINT := SINT#-1,SINT#-1,SINT#-1,SINT#-1;
41   b_Min: ARRAY [0..3] OF SINT := SINT#100,SINT#100,SINT#0,SINT#0;
42   MPC_Subsystem_b_Kx: ARRAY [0..5] OF LREAL := LREAL#5.4697395818657615,LREAL#0.7092108285119211,
43   LREAL#4.18, LREAL#1.7470000008,LREAL#5.341801514770895,LREAL#0.69074045484870261,LREAL#407.77137297368924,
44   LREAL#0.1451072145622892,LREAL#-0.1780279296089797,LREAL#-0.21407418193041945,LREAL#-0.43912784438738422,
45   LREAL#-0.2853058464217895,LREAL#-0.3483730460769084,LREAL#-0.3882253156489306,LREAL#-0.43912784438738422,
46   LREAL#-0.4827046872808573,LREAL#-0.5400817076962554,LREAL#-0.60789638011467958,LREAL#-0.685868515838431,
47   LREAL#-0.7327620734805842,LREAL#-0.80045308728924438,LREAL#-0.86981953195262686,LREAL#-0.84121038508732355,
48   LREAL#-1.0151789780434622,LREAL#-1.0914771124704263,LREAL#-1.1700610448137079,LREAL#-1.250895470505866,
49   LREAL#-1.333895805402082,LREAL#-1.4108116883639121,LREAL#-1.506366244950879,LREAL#-1.597272243188795,
50   LREAL#-1.687107148090846,LREAL#-1.7804808145471673,LREAL#-1.8758042727697448,LREAL#-1.97303838911627,
51   LREAL#-2.072147178925583,LREAL#-2.1738191644663989,LREAL#-2.2752381171305263,LREAL#-2.3813446689574078,
52   LREAL#-2.4895222647896724,LREAL#-2.5945144411525914,LREAL#-2.7041107259241924,LREAL#-2.813327874145718,
53   LREAL#-2.928145562849147,LREAL#-3.045102937144938,LREAL#-3.1634912603557840,LREAL#-3.27684767795976
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

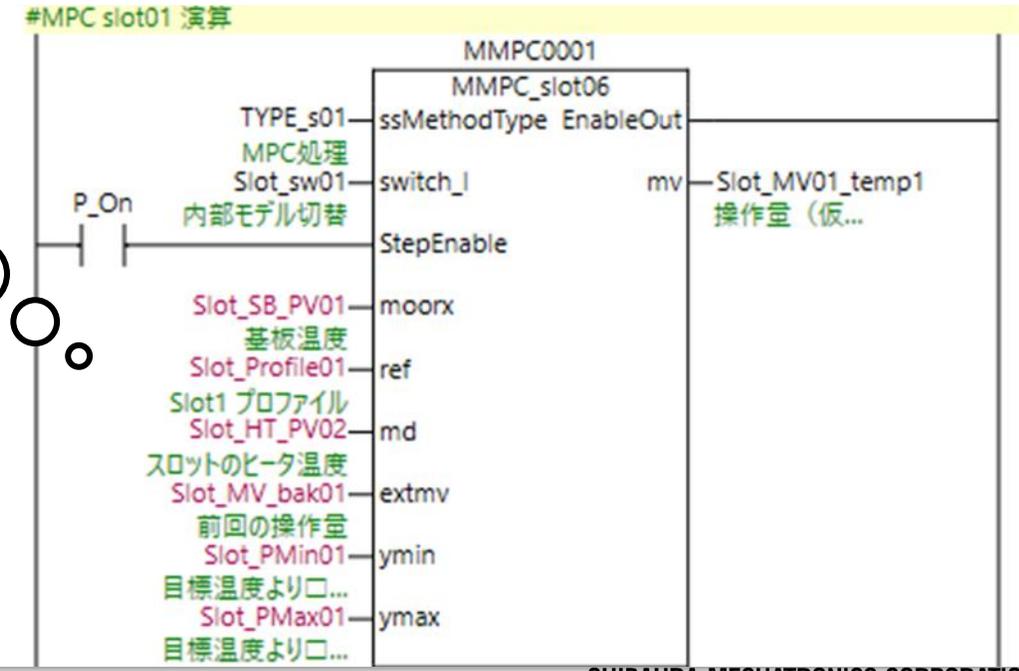
```

## ○ STEP2 ラダーへ組み込み

- ▶ IDEのメニュータブ「ツール」→「STプログラムのインポート」にて生成したSTコードを読み込む
- ▶ ラダー内でSimulinkと同じ入出力を定義し、実装



作成したSTコードを  
インポート

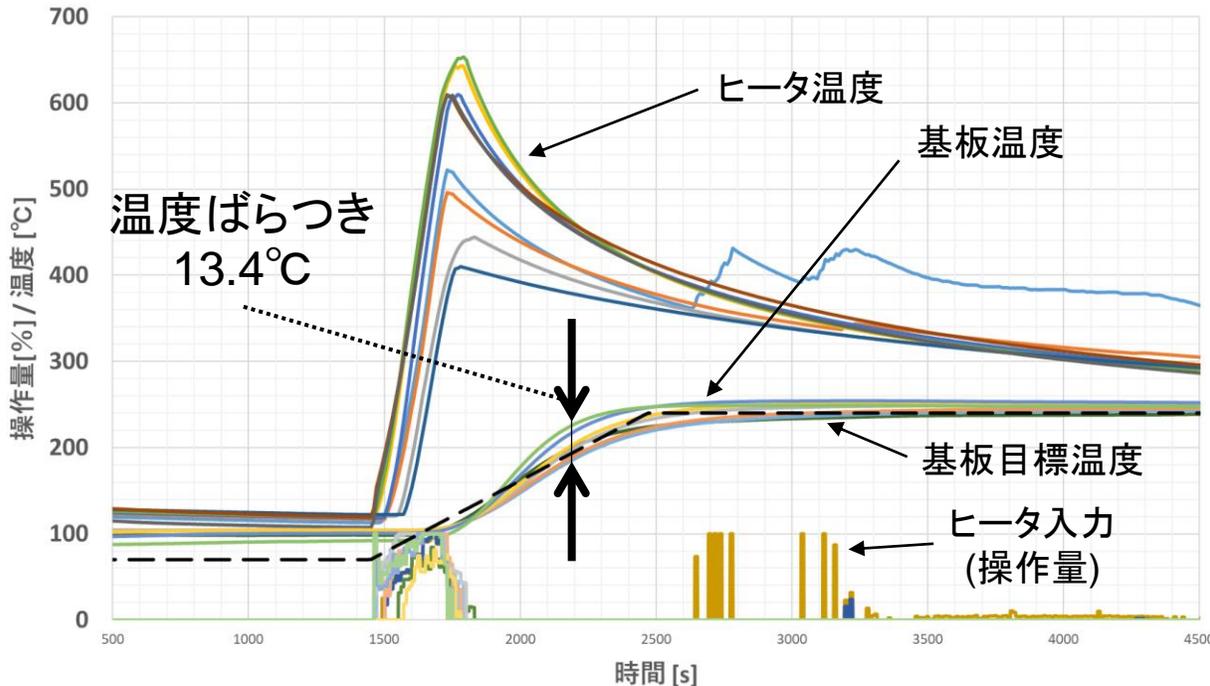


○ 以上の取り組みにより、温度制御開発を進行中

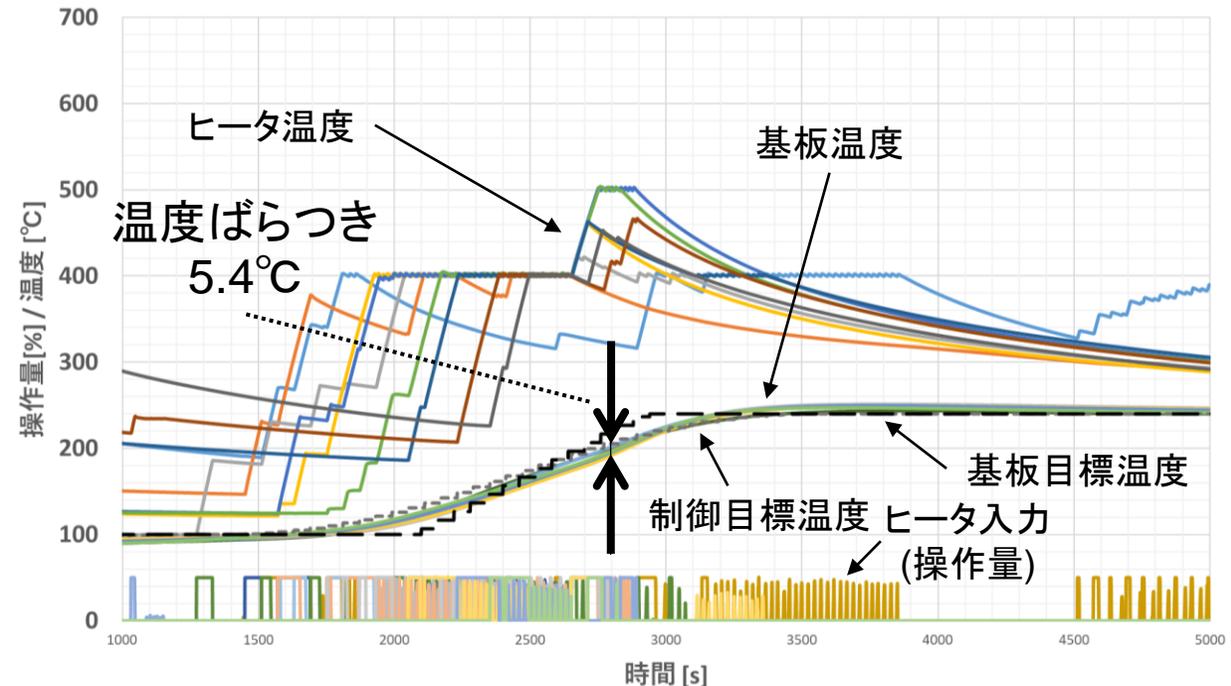
- ▶ 評価段階の結果(一部の処理層)のご紹介
- ▶ 目標の追従性向上・制御量ばらつき低下の効果が見られる

※ 開発中参考値

- 制御点面内ばらつき(標準偏差) 平均:  $13.4^{\circ}\text{C}$  →  $5.4^{\circ}\text{C}$
- 制御点追従誤差(MAPE)平均:  $14.1\%$  →  $3.4\%$



既存の制御手法 (調整途上値)



MPCによる制御手法

- 会社紹介
- 導入背景と課題
- 温度制御におけるモデルベース開発適用事例
- 開発実施結果とまとめ

## ○ モデルベース開発導入の良かった点/困難だった点

### 良かった点

- ▶ 実機レスで制御設計ができた
- ▶ パラメータ調整確認の時間が短縮できた
  - ・ 既存の方法: 0.5 日/回
  - モデルベース開発: 10 秒/回

### 困難だった点

- ▶ 制御対象のモデリングの難易度が高かった
  - ・ 設計部との連携が取ればよかった  
( CAD → Simulinkの変換機能の活用 )

## ○ MPC導入の良かった点/困難だった点

### 良かった点

- ▶ 既存制御より高精度な制御が一部でできた
- ▶ パラメータ数を削減できた
  - ・ コントローラ数 既存: PID制御器7つ
  - 本取り組み: MPC1つ
  - ・ 調整項 既存: 各ゲイン、操作タイミング
  - 本取り組み: 設計要件、制約条件

### 困難だった点

- ▶ 高精度な制御対象モデルがないと高度な制御ができない
  - ・ 現在も制御対象モデルを改善中
- ▶ MPC設定によってはPLCへ実装できない
  - ・ 実装できる最大コード長を超える場合がある

## ○ まとめ

- ▶ モデルに基づく制御系設計をし、実機レスでの制御開発を進めた
- ▶ 有機膜形成装置についてMPCを用いた制御ができた
- ▶ Simulink PLC Coderにより、MPCを即座にPLCへ実装できた
- ▶ 上記取り組みにより、高度な制御を実装することができた

## ○ 今後の取り組み

- ▶ モデリング技術の向上
- ▶ 他の装置についてもモデルベース開発の取り組みを横展開
  - 開発期間短縮や性能改善に寄与
  - 当社装置の価値と競争力の向上