# 第12章 制御系のHILS実験

これまでに2台の宇宙船が協力して協調的な作業を行うための制御アルゴリズムを作り、シミュレー ションを行いました.

この章では、シミュレーションに使ったモデルをプラント部と制御部に分割して、通信サブシステ ムを付加してビルドします.

ビルドしたプログラムを2台のパソコンへダウンロードして、HILS形式の制御実験を行い、結果を Simulinkのシミュレーションと比較します.

# ■ 12.1 機材の直進と回転運動

HILS実験を行うために、宇宙船による機材運搬の制御モデルをプラントとコントローラに分離します.

ホストにおいてモデルをビルドして、プラント部はターゲット1ヘコントローラ部はターゲット2へ ダウンロードします.

ここで使用するパソコンの役割分担を図12.1に示します.

パソコン間の通信方式を表12.1に示します.

プラントからコントローラに対して4個の変数,

[Body]のワールド座標(x, y, z)

[Body]の方向(z軸に関する回転角)

を送信します.

コントローラは、この情報に基づいて制御量、

宇宙船1の推力

宇宙船2の推力

を決定して、これらをプラントへ転送します.

プラント-コントローラ間において授受されるデータを図12.2に示しました.

プラント-コントローラ間の通信は、シリアル通信です.

xPC Targetにおけるシリアル通信の使い方については、第5章と第6章において詳しく述べました.





表12.1 パソコン間の通信方法

パソコン		通信方法
ホスト	ターゲット1	シリアル通信
ホスト	ターゲット2	LAN
ターゲット1	ターゲット2	シリアル通信

図12.1 パソコンの役割分担



図12.2 転送する情報

本章では、これらの章において制作した通信サブシステムを使用します.

では、具体的なモデルを取り上げて分割の過程を示します.

まず,前章で制作したモデルm1101(259ページ)をベースにします.

このモデルは2台の宇宙船に対して同量の噴射・逆噴射の推力を与えて, [Body]を直進, 停止します. プラントのモデルをmtg1, コントローラのモデルをmtg2とし, ディレクトリm1201に格納します. mtg1を画面12.1に示します.

プラントの主部は、SimMechanicsを用いて構築した[CG Revolute]サブシステムです.[CG Revolute]サブシステムの内容はこれまでと同じです.変更はありません.

宇宙船1に対して,

	A
センサ [Sensor1] サブシステム	
宇宙船2に対して,	
アクチュエータ [Actuator2] サブシステム	x
センサ [Sensor2]サブシステム	





画面12.1 mtg1

🖾 e tat Voce P	lan 142	
7H308 ##8	) 義宗学 バルトの学 書詞学 5	48 499
	1 X N B S C I F × F	
10%	1008	isele .

画面12.2 comPlant42

ADDI DALON (HAROSH)	
ABDI Booth Decole aud shine to octor, weather	
Ph-1	
Pontel Pres.	
2000	
Number of vehicles.	
è	
Validate Agenci	
From Second	

# 

画面12.4 [ASCII Encode] ブロックの設定ダイアログ

画面12.3 [ASCII Decode] ブロックの設定ダイアログ

#### 表12.2 受信する変数

設定項目	設定値
Format string	%f,%f;¥r
Number of variables	2
Variable type	{ 'double' ' double' }

を接続します.

[Actuator] サブシステムは、宇宙船の重心に対してz軸回りのトルクと直進推力を与えます.

[Sensor] サブシステムは、宇宙船のワールド座標系における重心の位置とz軸に関する回転角を観測します.

これらのサブシステムの構造は, すでに述べたとおりで変更はありません. mtglの通信部のモデル comPlant42サブシステムの構造を**画面12.2**に示します. 制御は宇宙船1, 2への直進推力なので, [ASCII Decode]ブロックの出力を2に修正しました. [ASCII Decode]ブロックの設定ダイアログを**画面12.3**に示しました. コントローラから受信する変数は宇宙船1, 2の直進推力の目標値なので,**表12.2**に示すように設定 します. 'double' はデフォルトなので省略が可能です. 状態出力は[Body]の*x*, *y*, *z*座標および[Body]の*z*軸に関する回転角です.

[ASCII Encode] ブロックの設定ダイアログを画面12.4に示します.

設定の内容を表12.3に示します.



表12.3 ブロックの設定

設定項目	設定値
Format string	%f,%f,%f,%f;¥r
Number of variables	4
Mux output string length	128
Variable type	{ 'double' ' double' ' double' }





画面12.6 comCont24 サブシステム

二つの宇宙船の状態はコントローラへ送信しません.

コントローラmtg2を画面12.5に示します.

フィードバック制御ではないので直接フィードスルーは存在しません.したがって,ダミーの微分・ 積分ブロックを入れる必要ありません.

一次公司の

[Body]のx軸変位を[Scope(xPC)]によって観測します.

シリアル通信を行う comCont24 サブシステムを画面12.6 に示します.

[ASCII Decode] と [ASCII Encode]の設定は、表12.1、表12.2を参照してください.

B(00) 9-M0

準備が完了しました.

ターゲット1とターゲット2の電源をONにして、待機状態にします.

MATLABの[Command Window] において,

>> xpcexplr

と入力して[xPC Target Explorer]を立ち上げます.

[xPC Target Explorer]の[xPC Target Hierarchy]ペインにおいてTargetPC1が選択されて、

## TargetPC1

と太字表示になっていることを確認します.

mtg1のメニューから[表示]→[モデルエクスプローラ]とクリックして, [Model Explorer]を立ち 上げます.

[Model Explorer]の[モデルの階層]ペインにおいてノードmtglを開き,[Configuration (Active)] を選択して,[コンテンツ]のペインにおいて[Solver]を選択して**画面12.7**に示すように設定します. 設定の詳細を表12.4に示します.





[コンテンツ]のペインにおいて[Real-Time Workshop]を選択して, さらに[システムターゲット ファイル]において,

### xpctarget.tlc

を選択します.

mtglのウインドウのメニューから[ツール]→[Real-Time Workshop]→[モデルのビルド]とクリックします.

mtg1はビルドされ,実行ファイルmtg1.dlmが生成されてTargetPC1へダウンロードされます.

```
同じ手続きを適用してmtg2をビルドして、TargetPC2へダウンロードします.
```

MATLABの[Command Window] において、

```
>> tg1=xpctarget.xpc('TargetPC1')
```

```
>> tg2=xpctarget.xpc('TargetPC2')
```

とタイプして, xPC Targetオブジェクトtg1とtg2を生成します.

そこで, MATLABの [Command Window] において

```
>> tg1.start;tg2.start;
```

```
とタイプして、ほぼ同時に2台のターゲットをスタートします.
```

コントローラのディスプレイを画面12.8に示します.

[Body]がx軸に沿って1m移動したようすが[Scope(xPC)]によって観測されました.



力学の教科書に記載されているように,宇宙船の位置は噴射時に放物線状に増加し,逆噴射時に放物 線状に減速して停止します.

続いて、[Body]を回転する実験を行います.

モデルをmtg1, mtg2として、ディレクトリm1201Rotへ格納します.

mtg1は変更ありません.

mtg2を画面12.9に示します.

[Body]を回転するために、宇宙船1は直進推力を出力し、宇宙船2は逆噴射するように[Gain]ブ ロックを設定し、[Scope (xPC)]ブロックを[Body]のz軸角度に取り付けました.

mtg2をビルドしてターゲットへダウンロードし,実行します.

ターゲット2の画面を画面12.10に示します.

画面12.10の右下部を見てください.

[Body]の角度は徐々に変化しています. 停止しません.

この理由については、すでに述べたとおりです.





# ■12.2 宇宙船の直進運動のフィードバック制御

宇宙船の直進運動をフィードバック制御します.

モデルを格納するディレクトリはm1202です.

プラントのモデルmtglは変更ありません.

コントローラmtg2を画面12.11に示します.

通信サブシステム [comCont24] は**画面12.6**と同じです.

[Body]のx座標の目標値を[Constant]ブロックによって指定します.

[Gain2] ブロックは、比例制御のゲインです.

[Gain3] ブロックは、微分制御のゲインです.

[Gain], [Gain1] ブロックは, 2台の宇宙船へ与える推力の大きさを決めるゲインです. ここでは直 進運動をするために両者共に1にします.

[Gain], [Gain1] ブロックの出力は, 通信サブシステムを介してプラントへ送られます.

通信サブシステムの出力はプラントから受信した状態に関するデータ,すなわち[Body]の座標値で す.ここからx軸座標に関するデータを取り出して目標値と比較します.この差が偏差となります.

フィードバックの回路に微分,積分ブロックを挿入したのは,直接フィード・スルーを回避するため の方策です.

これらのブロックの必要性については、すでに説明しました.

それでは、準備ができたのでHILSの実験を行います.

ターゲットのパソコンを起動し、ローダの状態で待機させます.

[Model Explorer]を立ち上げて, [Solver]と[Real-Time Workshop]の設定をこれまで述べたとおり に行います.

[xPC Target Explorer]を立ち上げて, TargetPC1をデフォルトのターゲットとします. mtg1をビルドして, ダウンロードします.

TargetPC2をデフォルトのターゲットとします.



#### 画面12.11 コントローラmtg2

