

実験データを用いてパラメータを 求める「システム同定」

藤原 大悟

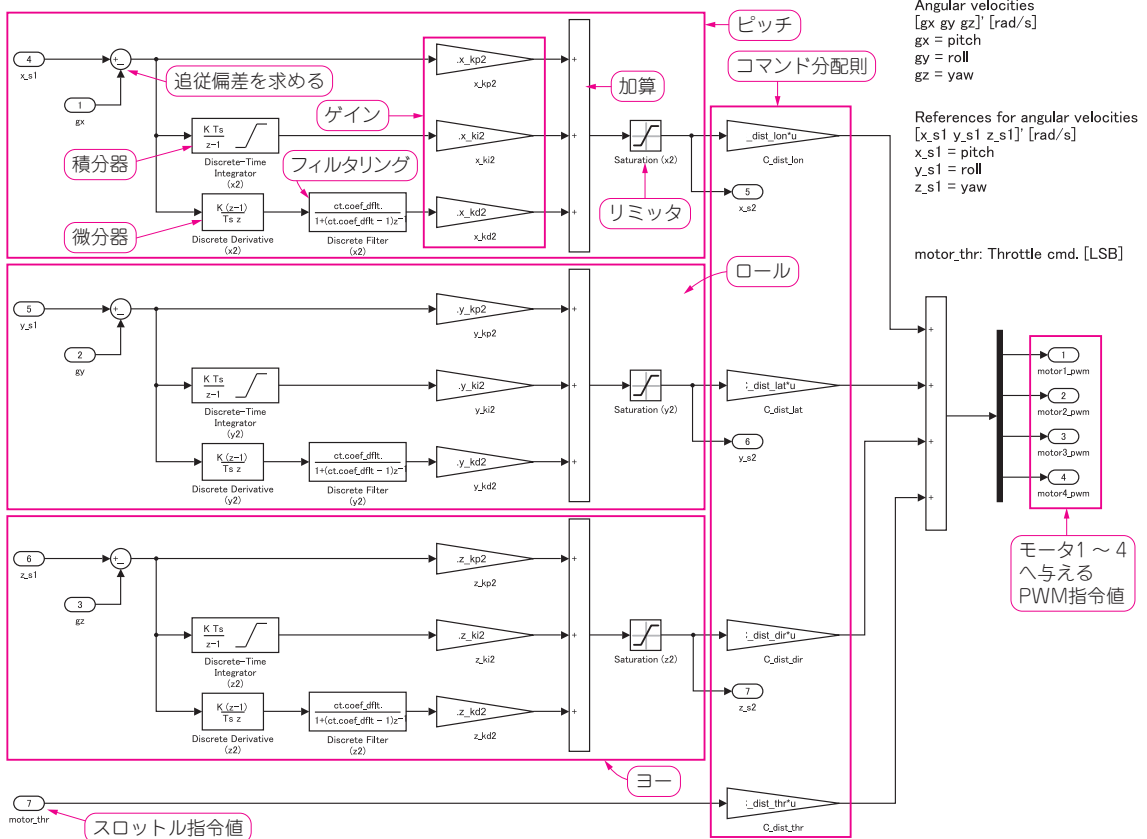


図1 SimulinkによるSTEVAL-DRONE01の角速度制御器の実装 (PDFで確認できる)
sim_model/Inner-loop controller (default)

準備…STEVAL-DRONE01の 角速度制御器/角度制御器の実装

ドローン評価キットSTEVAL-DRONE01 (STマイクロエレクトロニクス)のFCU (Flight Control Unit : 飛行制御ユニット)ファームウェアには、角速度制御器と角度制御器が実装されています。システム同定実験の際はこれらも動作させるので、Simulinkモデルにも同等のものを実装しておきます。sim_modelの最上位階層(第4章の図2)のうち、Inner-loop controller

(default)が角速度制御器、Outer-loop controller (default)が角度制御器です。それぞれサブシステムであり、中身は図1、図2です。制御器のファームウェア・プログラムの詳細については文献(1)を参照してください。文献(1)は以下のウェブ・ページからも参照できます。

<https://interface.cqpub.co.jp/2209tb2/>

図1と図2は誌面の都合上、見づらいと思います。図1と図2のPDFもダウンロードできます。

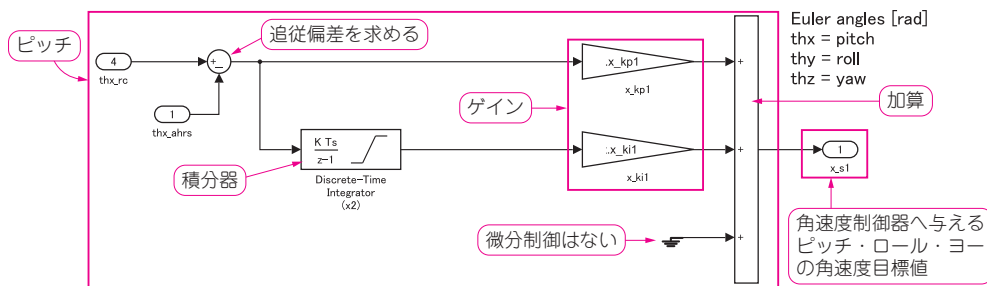


図2 SimulinkによるST-DRONEの姿勢角度制御器の実装…ロールとヨーについても同様(PDFで確認できる)
sim_model/Outer-loop controller (default)

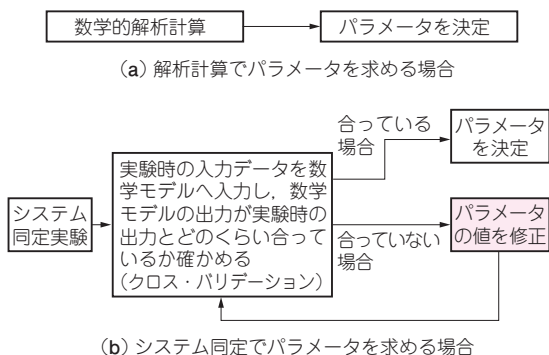


図3 モデル・パラメータの値の決定方法

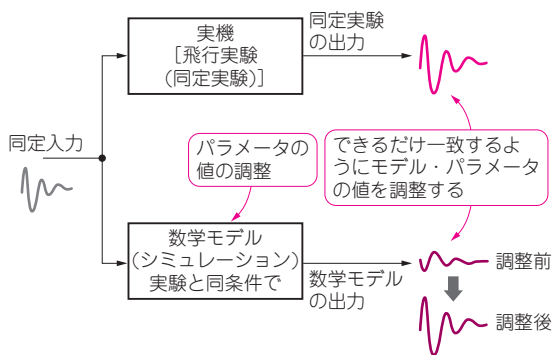


図4 クロスバリデーションのイメージ

● 角速度制御器

角速度制御器(図1)から見ていきます。大きく分けて左半分は、上からピッチ、ロール、ヨーの制御器とスロットル指令値に関する計算で、右半分はコマンド分配則の計算となっています。ピッチ、ロール、ヨーの制御器は構成要素がほとんど同じなので、ピッチについて説明します。

図1の左上に注目します。角度制御器から来る角速度目標値 x_s1 と、Drone model(実際の機体の場合はジャイロ・センサ)から来る角速度 g_x を入力信号とし、Sumブロックで引き算して追従偏差を求めます。この結果を3つに分け、一番上はそのまま比例ゲイン x_kp2 を掛けます。

2番目は、離散時間積分であるDiscrete-Time Integratorブロック(分類: Discrete)へ入力します。この積分は、積分結果に対するリミッタの機能もあります。その後、積分ゲイン x_ki2 を掛けます。

3番目は、離散時間微分であるDiscrete Derivativeブロック(分類: Discrete)へ入力し、その後、高周波成分をカットするノイズ・フィルタを実装するDiscrete Filterブロック(分類: Discrete)へ入力し、最後に微分ゲイン x_kd2 を掛けます。ゲインを掛けて求めた3つの値をSumブロックで加算し、Saturationブロック(分類: Discontinuities)でリミッ

タを掛けて、ピッチ角速度の制御入力がかかります。そして、コマンド分配則をかけて、各モータのPWM指令値を求めます。

● 角度制御器

角度制御器(図2)を見ていきます。上からピッチ、ロール、ヨーの3つに分かれています。誌面の都合でロールとヨーについては省略しました。PDFで確認できます。角速度制御器との大きな違いは、微分制御がないこと、コマンド分配則がないことです。それ以外は角速度制御器と同様の計算をしています。入力信号は角度目標値 thx_rc 、 thy_rc 、 thz_rc とオイラー角 thx_ahrs 、 thy_ahrs 、 thz_ahrs 、出力信号は角速度目標値 x_s1 、 y_s1 、 z_s1 です。

システム同定

数学モデルのパラメータが解析計算だけで求められない場合、実験(システム同定実験)を行い、得られた実験データを用いてパラメータを求めるシステム同定を行います。また、システム同定の 일환として、実験時の入力データを数学モデルへ入力し、数学モデルの出力が実験時の出力とどのくらい合っているか確かめることを、クロスバリデーション(Cross-Validation:)

◆参考文献◆

(1) 藤原 大悟; まずはここから…PID制御飛行プログラム, Interface, 2020年3月号, pp.54-69, CQ出版社.

交差検証)と呼びます(図3, 図4)。

第3章に出てくるパラメータのうち、 K_T (推力/アクチュエータ・モデルのゲイン [N/LSB])、 τ_T (推力/アクチュエータ・モデルの時定数 [s])、 K_Q (トルク・アクチュエータ・モデルのゲイン [Nm/LSB])、 τ_Q はシステム同定によって求めました。 τ_Q は第3章に出てきていませんが、トルク・アクチュエータ・モデルの時定数 [s] です。本章のシステム同定の結果、値が0になったので、第3章の式(2)では省略しました。

● 精度を高めるには実験が重要になる

システム同定の精度を高めるには、システム同定実験が重要です。 K_T 、 τ_T はプロペラ推力に関するパラメータですので、プロペラ推力がさまざまな値をとるような実験を行う必要があります。具体的には、プロポのスロットル・スティックを可能な限り上下に大きく、かつ、さまざまな周波数で振る、といった飛行のさせ方をします。 K_Q 、 τ_Q は空力トルクに関するパラメータですので、これに関係が深い、機体の方向の変化が生じるような実験、つまり、飛行中にプロポのラダー・スティックを振る操作をします。もちろん、墜落や衝突などの事故が起きない程度に、十分注意して操縦します。まずは、筆者が上記のようなシステム同定実験を行い、得られたデータをMATLABで作図してみます。

同定実験の結果をMATLABでグラフ化する

● 3つの実験

実験データのファイルは以下の3つがあります。

- expdat_sysid_01_thr.txt... K_T 、 τ_T を同定するため、スロットル・スティックを振った実験
- expdat_sysid_02_dir.txt... K_Q を同定するため、ラダー・スティックを振った実験
- expdat_sysid_03_lat.txt... K_T 、 τ_T の同定精度を確認するため、エルロン・スティックを振った実験

作図のための.mファイルは、dispexpdat_sysid.mです。dispexpdat_sysid.mはイントロダクション3に示した方法で閲覧できます。本誌サポートページからはPDFを閲覧できます。

<https://interface.cqpub.co.jp/2209tb2/>

これを実行すると、上記の実験データのうち1つを読み込んで、グラフを3枚出力します。この.mファイルを簡単に説明します。

● 読み込み元のファイル名

12～14行目で、読み込み元のファイル名を設定し

ます。あらかじめ3つのファイル名を書いてあり、読み込みたいファイルの行頭のコメント・マーク%を削除し、それ以外のファイルには行頭に%を付けます。このリストでは、expdat_sysid_03_lat.txtを読み込む場合の設定になっています。

17～18行目は、読み込み元の実験データ・ファイルの様式を変数type_fileに指定します。実験データは無線でダウンリンクして地上のPCに記録したので、データ・レートの都合上、取得したデータが実験ごとに異なり、ファイルの中身が異なるためです。12～14行目の設定内容に合わせる形で、expdat_sysid_01_thr.txtの場合は1、それ以外の2つの場合は2を指定します。

● サンプリング時間の設定

21～22行目はサンプリング時間の設定です。21行目のts_usr_1_expdatはFCUファームウェアの変数cnt_usr_1がカウントアップされる周期、22行目のts_d_expdatは実験データ・ファイル1行ごとの時間間隔です。

● 実験データをファイルから読み込み

30行目でワークスペースにある古い変数expdatrawとexpdatを消去し、33行目で実験データをファイルから読み込み、構造体変数expdatrawの変数dへ行列として格納します。関数dlmread()は、区切り文字で区切られた数値データ列を一度に読み込む関数です。区切り文字は第2引数で指定し、この場合は半角スペースです。また、冒頭のヘッダ行を読み飛ばすこともでき、第3引数で読み飛ばす行数を指定します。

● 読み込んだデータを種類別に分ける

36～58行目は、読み込んだデータを種類別に分け、意味のある値へ変換します。例えば、36行目では、expdatraw.dの1列目を10倍して、構造体expdatの変数cnt_usr_1へ格納します。expdatraw.dの右にある丸括弧()は、行列の指定の成分を取り出す命令で、カンマで区切られ、1つ目は行番号、2つ目は列番号です。(:,1)のようにコロンを指定すると、全ての行(または列)を取り出す、という命令になります。

38～40行目のpiは円周率 π を表すMATLABの組み込み定数です。

41～58行目は、変数type_fileの値に応じた場合分けをしています。switch文はC言語と同じ意味です。type_fileが1の場合は、4つのモータのPWM指令値が記録されていますが、角速度目標値が記録されていないので、値が0のダミー・データを作

(2) 藤原 大悟：高度な制御のための運動のモデル化&定式化, Interface,

2020年3月号, pp.74-80, CQ出版社。

ります。逆に、`type_file`が2の場合は、4つのモータのPWM指令値が記録されていないので、値が0のダミー・データを作ります。

● 作図に必要な追加のデータを作る

60～82行目では、作図に必要な追加のデータを作っています。62行目と64行目はサンプリング時間、66行目はグラフの横軸に使う時刻のデータです。71～75行目はコマンド分配則、76行目はその逆行列を作り、77～81行目は4つのモータのPWM指令値からピッチ、ロール、ヨーの制御入力`expdat.x_s2`、`y_s2`、`z_s2`へ変換しています。`x_s2`、`y_s2`、`z_s2`はFCU内にあるデータですが、無線ダウンリンクのデータ・レートの制約によりダウンロードできなかったため、モータのPWM指令値からの逆算で作っています。82行目は不要になった中間変数`dfunc`をワークスペースから削除します。

● 作図

85行目以降は作図になります。1枚目のグラフについて説明します。

93行目で古い変数`haxes`をワークスペースから削除した後、94行目の関数`figure()`でグラフを表示するためのFigureウィンドウを作ります。引数は、Figureウィンドウに付ける番号です。戻り値の`hfig`は、そのFigureウィンドウを指すハンドルという変数で、後で使います。95行目でFigureウィンドウの中身をクリアします。この命令はなくても構いません。

96行目の関数`subplot()`は、Figureウィンドウ内をいくつかのグラフに分ける命令です。第1引数は行数(縦の数)、第2引数は列数(横の数)であり、ここでは3行1列に分ける指示です。第3引数は、この後のどのグラフを操作するかを示します。これは、1行1列、1行2列、…、2行1列、…の順の連番で指定します。ここでは1、つまり、1行1列目のグラフを操作することを指定しています。戻り値は、操作対象のグラフの軸へのハンドル変数です。ここでは、ベクトル変数`haxes`の1番目に代入しています。MATLABでは変数の宣言が必要ないことは既に説明しましたが、ベクトルや行列についても、事前に型やサイズの宣言は不要で、サイズは必要に応じて自動的に拡張されます(ただし拡張のため動作が遅くなる)。

97行目の関数`plot()`は、グラフにデータをプロットする命令です。第1引数が横軸、第2引数が縦軸のデータを示します。第3引数は、プロットの種類や色を指定します。`'b-'`は青色の実線です。' 'で囲まれたものは文字列を表します。

98～99行目の関数`xlabel()`、`ylabel()`は横軸と縦軸のラベルを出力する命令です。これも ' 'で

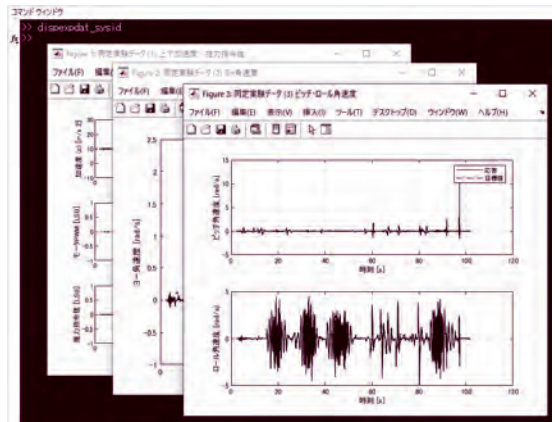


図5 筆者が行った同定実験の結果をMATLABで表示した様子(`dispexpdat_sysid.m`)

囲んで文字列で指定します。

100行目はグリッドを表示する命令です。`grid`は関数形式`grid()`による書き方もあります。これでFigureウィンドウ内にグラフが1つ作られました。101～107行目では2つ目のグラフを作っています。1つ目のグラフとほぼ同じですが、102行目の関数`plot()`の引数が増えています。これは、複数の線をグラフに重ねて描く指示です。先ほど説明した3つの引数の組を、4つ目以降に順次指定していけばよく、ここでは4本の線を表示しています。102行目の末尾に...とあるのは、次の行に続く、という意味です。MATLABスクリプトは改行で命令の末尾となるので、C言語のように途中で改行できません。1つの命令を次の行に続けて書く場合は、必ず行末に...を書きます。106行目の関数`legend()`は、凡例を表示する命令です。ここではモータの番号を表示しています。

108～112行目で3つ目のグラフを作った後、113行目で関数`linkaxes()`を呼び出しています。これは、変数`haxes`に格納している3つのグラフの軸について、横軸(x軸:第2引数で指定)を互いにリンクし、グラフを拡大、縮小、移動した場合に、3つのグラフの横軸が連動して動くようにします。3つのグラフとも横軸は時刻なので、このように設定した方が見やすくなります。114行目で、Figureウィンドウのタイトル・バーに表示する名称を設定します。Figureウィンドウが多くなると、ウィンドウを探すのが大変なので、名前を付けておきます。関数`set()`はハンドルに関連付けられたプロパティに値を設定する命令で、ここではハンドル`hfig`のプロパティ`Name`の値として、文字列'同定実験データ...'を設定しています。

以上で、1つのFigureウィンドウを作り終わりました。その後、同じようにして2つ目と3つ目のFigureウィンドウを作成します。

イントロ
第1部

A / 画像処理基礎

第2部

テスト駆動開発

第3部

ドローン飛行制御

第4部

フィードバック制御

第5部

姿勢推定

● プログラムの実行

dispexpdat_sysid.mを実行すると、図5のように3枚のFigureウィンドウが表示されます。実験データexpdat_sysid_01_thr.txtはFigure 1ウィンドウ、expdat_sysid_02_dir.txtはFigure 2ウィンドウ、expdat_sysid_03_lat.txtはFigure 3ウィンドウに、主要なデータを表示しています。各グラフの上にマウス・カーソルを合わせると、グラフの右上にズームイン(“+”の虫眼鏡)、ズームアウト(“-”の虫眼鏡)、移動(手のマーク)などのツールが表示されるので、見やすいように表示を変えて、データを観察してみてください。

実験データを流し込んでこれまでに作った数学モデルを検証する

ここからは実験データを用い、クロスバリデーションによってパラメータの同定をしていきます。クロスバリデーションでは、実験データを流し込んで数学モデルのシミュレーションを行います。その準備と実行、結果表示はrunsim_sysid.mで行います。runsim_sysid.mはイントロダクション3で紹介した方法で確認できます。

● 読み込むデータを選択

runsim_sysid.mの中身を簡単に解説します。最初に11行目でsim_param.mを実行し、モデル・パラメータをワークスペースへ読み込みます。21行目の変数sel_expdatに代入する値により、読み込む実験データを選択します。番号と実験データとの対応は27～89行目のswitch文を見てください。

23～24行目はサンプリング時間の設定です。

● 使用する実験データに対応する各種設定

27～89行目は、使用する実験データに対応する各種設定を行います。case 1はクロスバリデーションをしない設定で、case 2～4がクロスバリデーションとなります。

44～58行目のcase 2を見ていきます。45行目で実験データのファイル名、46行目でファイルの様式を設定します。

47～50行目では、数学モデル内の積分計算の動作を選択します。case 2は上下加速度の値(推力の代わり、推力そのものが計れないため)を使ってクロスバリデーションするので、角速度、姿勢、速度、位置はいずれも使いません。値に2を指定し、モデルの積分器に0を入力することで、これらの値を初期値に固定します。

51行目は、モデルのPWM指令値にどの信号を入力するかを指定しており、ここでは同定実験時のPWM指令値を入力するので、2とします。

53～56行目は、角速度制御器へ入力する目標値とスロットル指令値としてどの信号を入力するかを指定しますが、case 2の場合は、角速度制御器とドローンの数学モデルの間の信号がつかっていないので、設定値に意味はありません。

57～58行目は、実験データのどの部分を使うか、開始および終了時刻を指定します。switch文を抜けて、91～137行目で実験データを読み込み、展開します。

● 実験データの切り出し

139～156行目で、実験データの必要な部分を切り出します。140行目では、切り出す開始時刻と終了時刻を用い、先頭から何番目のデータを切り出すかを探し、該当する全てのインデックスを変数i_fldに格納し、それを使って141～156行目でデータを切り出します。なお、141行目において、切り出す先頭の時刻を差し引くことで、先頭の時刻を0sとします。切り出したデータは構造体expdat_cに入れます。

159行目ではシミュレーション時間を切り出した実験データの時間に合わせます。160～164行目では、切り出した時系列データをSimulinkモデルに流し込むための変数を準備します。構造体u_inは角速度制御器への入力信号、構造体dltx_inはドローンの数学モデルへ入力する各モータのPWM指令値に対応する構造体メンバのうち、timeは時刻の列ベクトル、signals.valuesは信号そのもので、行(縦)方向が各時刻、列(横)方向に各信号の値が並ぶ行列です。従ってtimeとsignals.valuesの行数は一致します。

● シミュレーションの実行

以上でシミュレーションの準備ができたので、172行目でsim_model.slxのシミュレーションを実行します。実行が終了すると、173行目以降が引き続き実行されます。シミュレーションを行うと、その結果として時系列データが構造体state_outとctrl_outに格納され、ワークスペースに入ります。175～218行目でそれらを改めて展開し、構造体simdatへ格納します。221行目以降では、シミュレーション・データを4つのFigureウィンドウに作図します。

パラメータ調整の手順

クロス・バリデーションによるモデル・パラメータの調整の手順は次の通りです。

● 推力/アクチュエータ・モデルのゲインと時定数(K_T , τ_T)

K_T , τ_T , K_Q , τ_Q の値は、sim_param.m(第4章参照)内では12～15行目に対応します。最初は正

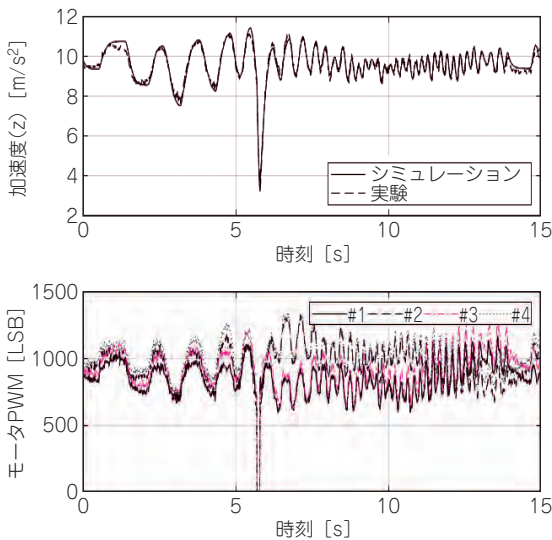


図6 上下加速度(推力)モデルのシステム同定実験と同定結果
#1～#4はプロペラの番号

の実数で仮の値を設定しておきます。最初は K_T 、 τ_T の同定をするため、`runsim_sysid.m`の21行目の`sel_expdat`の値として2を指定します。`runsim_sysid.m`を実行し、Figureウィンドウの結果を見て、機体z軸の加速度の値について、同定実験とシミュレーションの両者を比較します。合っていない場合は、 K_T 、 τ_T を変えて、`runsim_sysid.m`を再度実行し、繰り返し調整します。両者がよく合えば終了です。

● トルク・アクチュエータ・モデルのゲインと時定数 (K_Q , τ_Q)

次に K_Q 、 τ_Q の同定へ移ります。`sel_expdat`の値として3を指定します。`runsim_sysid.m`を実行し、Figureウィンドウの結果を見て、ヨー(機体z軸周り)の角速度の値について、同定実験とシミュレーションの両者を比較します。合っていない場合は、 K_Q 、 τ_Q を変えて、`runsim_sysid.m`を再度実行し、繰り返し調整します。両者がよく合えば終了です。

● 姿勢変化の応答

最後に姿勢変化の応答についてクロスバリデーションします。`sel_expdat`の値として4を指定します。`runsim_sysid.m`を実行し、Figureウィンドウの結果を見て、ロール(機体y軸まわり)の角速度の値について、同定実験とシミュレーションの両者を比較します。数学モデルの数式やそのSimulinkへの実装が問題なく、 K_T 、 τ_T がきちんと同定できていれば、両者は完全に一致しないまでも、かなり近い応答になるはずですが、さもないと、モデルの数式の作り方が根本的に誤っている可能性があります。

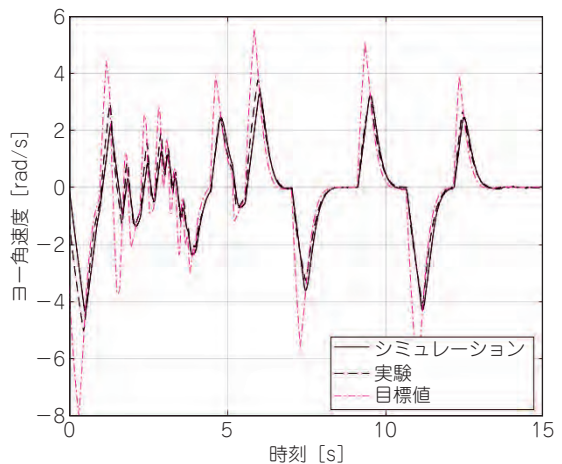


図7 ヨー角速度(トルク)モデルのシステム同定実験と同定結果

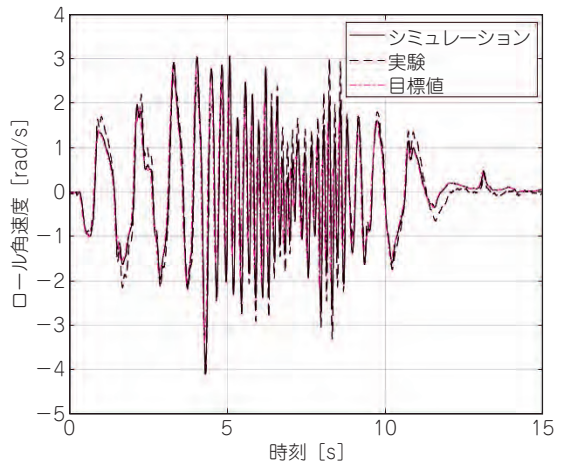


図8 ロール角速度モデルのシステム同定実験と同定結果

● 調整結果

4つのパラメータを同定し終わった後の、クロスバリデーションの結果を図6～図8に示します。図6は`sel_expdat`を2とした場合で、上のグラフに示されたz軸加速度について、同定実験と同定結果(シミュレーション)の両者がよく合っていることが分かります。図7は`sel_expdat`を3とした場合で、ヨー(z軸周り)角速度について、同定実験と同定結果(シミュレーション)の両者がよく合っていることが分かります。図8は`sel_expdat`を4とした場合で、ロール(y軸周り)角速度について、同定実験と同定結果(シミュレーション)の両者を比較すると、高い周波数で機体を揺さぶった5～8sは振幅が若干異なりますが、それ以外は振幅、位相ともよく合っています。以上で、ドローンの数学モデルの完成です。

ふじわら・だいご

イントロ
第1部

AI / 画像処理基礎

第2部

テスト駆動開発

第3部

ドローン飛行制御

第4部

フィードバック制御

第5部

姿勢推定