

〔事例報告〕

# モデルベース開発 (MBD) と制御工学 — 制御理論にもとづくモデルを使えるエンジニアの育成 —

清水 良\*

\*日本文理大学工学部機械電気学科

## Control Engineering for Model-Based Development — Education of Engineers Who Can Utilize Models Based on Control Theory —

Ryo SHIMIZU\*

\*Department of Mechanical and Electrical Engineering, School of Engineering,  
Nippon Bunri University

### 1. はじめに

モデルベース開発 (以降 MBD と記す) は主に自動車会社を中心に、開発期間の効率化 (時間/費用) と品質の向上を果たすべく、導入されてきた開発のやり方である。

広義には機能開発をモデル予測を駆使して行うことを指すが、最近では電子制御を備えた製品 (機械、例えば自動車) をいかにコンピュータ上でシミュレーションして仕様を摺りあわせ、試作品を作らずに完成度を高めるか、という意味で用いられる<sup>(1)</sup>。いずれにせよ機能 (性能) を予測できることが必要で、そのため計算できる「モデル」が必要となる。

このモデルは、物理・数学を基礎に工学のさまざまな理論式や経験式、実験で得られたデータで構成され、コンピュータシミュレーションに供する高度なものから、それこそ手計算で算出するような簡易的なものまで含まれる。MBD を実践するには、モデルのレベルはどうあれ、とにかく各機能がすべて予測できるようにすることが重要である。そうすることで、各機能の配分を調整し、製品の狙いに合致した最適化を行うことが可能になる。

一方、大学での科目である制御工学だが、歴史的背景からしても、こういう製品 (機械) が作動する際にいかに安定して動かすことができるか、精度よく動かせるかについての理論であり、当然設計に反映されるべきものである。しかし、特に最近の電子制御を有した機械において、著者の経験からしても、MBD に積極的に取り込まれているとは言い難いのが現状である。

そこで、本稿ではまず MBD について概説した後、いかに制御工学を取り込んでいくのか、そのためにはどんなことを学び理解する必要があるのか、ということについて示す。具体的には自動車の電子制御における制御工学の適用事例を用い、その「有難み」に触れつつ、MBD に適用できる技術者の育成における課題やその対応について述べていく。

### 2. モデルベース開発

#### 2-1 開発プロセス

MBD のプロセスのスタートは対象物 (製品) を性能であれば、それを構成する機能に分解し、その上位から各機能毎に目標を設定することである (最上位は性能目標)。この機能目標の設定時にも各機能単位での予測 (シミュレーション) を行い、各機能目標の配分を行う。分



図1. 自動車用エンジンの機能展開

解していった各機能(図1)毎に影響する因子を抽出(部品のスペック)し、それを元にそれぞれモデル(予測式)を作り、上位機能とつないでいく。そして上位から下位へ目標達成の予測(モデル計算、もしくはシミュレーション)を行いながら目標を再配分しつつスペックを調整して、狙いにマッチするよう最適化を図っていく。

2-2 電子制御システムへの適用

電子制御システム(特にソフトウェア)におけるMBDプロセスでは、少し様相が異なる。なぜなら、制御システムの場合、例えばエンジンの最高出力のようなパフォーマンスとしての目標値設定がほぼできないためである。通常、制御システム(機械的な機構で制御するものも含めて)は何らかの目標にいかにか合致しているかがその目的であり、フィードバックシステムは言うまでもなく、オープンループシステムにおいても、狙った値にどれだけ近いかがそのよし悪しを決める評価になる。このため、制御の目標は、いわゆる精度(ばらつき、誤差)と追従性(応答性)に集約できてしまい、すべてこれらからの展開となってしまうため、個々の細かい機能及び目標を明確にするのが難しい。

例えば、空調システムであれば、その冷却能力の目標はコンプレッサーなどの性能で決まるが、電子制御は、いかに目標温度に素早く到達できるか、その精度はいくらか、ということになる。だが、この目標は?、どのく

らいであればOKか、というのは状況に左右され、設定は難しい。さらに、これらは電子制御だけでは決まらず、図2のように部屋の広さなどにも依存するため、単純にこの目標値の予測モデルを作るのは、製品単体では難しい。いずれにせよ、何らかの仮定が必要になる。

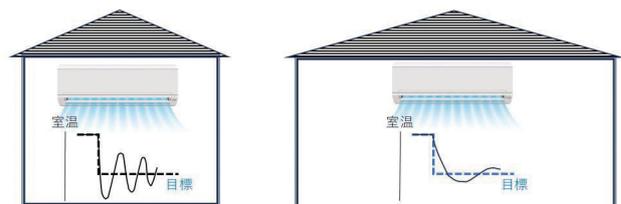


図2. 空調システムの制御

また、通常制御システムに要求されることは、「何事も起きない」「不具合がない」ことであり、これも目標とするのは困難である。尚、これらは問題となる事象を突き詰めて考えれば、一般に精度の1つの側面である制御目標値からの変動や急変しない(安定性)ことを目標として設定することができるケースが多い。

まとめると、制御システムの機能展開の最上位は精度、追従性とした上で、状況を仮定、つまりいろいろなその製品が使われるシーンを想定し、それぞれの場面で許容される範囲を何らかの考えで定め、機能展開していくことが必要である。FMEAや品質問題をベースに行うFTAの要領に近いと言える。

### 2-3 電子制御システムの設計と検証

ある程度の必要機能が明らかになれば、その因子のうち INPUT 情報として精度を得るために必要なものをセンサーとして採用し、制御するために必要なものを OUTPUT デバイスとして採用し、制御システムのハード構成が定まっていく。このとき、機械的な機構のみで実現できるか、電子制御にするかの判断も併せて必要である（電気コタツのように温度を ON/OFF だけで制御するぐらいの変動が許容されるなら、サーミスタの SW で十分機能を実現できる）。

ソフトウェア設計については、各必要な制御機能を考えるが、基本的には制御が何らかの操作を行ったときに制御対象（つまり製品）が目標通りに動けば良いので、制御対象のモデル（プラントモデル）をソフトウェアの中に持ち、その逆モデルで目標通りに動かす際の操作量を定めれば良い。完全な逆モデルを持てれば図3のように100%目標通りの制御ができるが、実際は何らかのモデルを作ることすら難しいため、目標からの誤差を検出して追従させるフィードバック制御などで補償することになる。これはセンシングにおいても同様で、空調システムの室温のように実際の温度を計測するといっても、周辺からの伝熱や輻射、センサー自体の熱容量などの影響を受け、真の室温は計測できていない。この場合、センサーに影響するすべての要因をモデル化し真の室温を

算出する式が作れば良いが、それは不可能なため、何らかの近似もしくはセンサー検出値そのもので制御を行うことになる。

ソフトウェアの設計は機能展開によって、個々の必要な機能に分解し、それぞれの INPUT/OUTPUT を明確にした上で個々のモデルを作り、それをコーディングしてモジュール(サブルーチンとか関数と呼ばれる単位)ができ、それらを集合して制御ソフトウェアができあがる。例えば、空調システムであれば、「室温を検出する」機能のために「センサーの電圧を読む」と「電圧を温度に変換する」という2つの機能が必要で、それぞれをプログラミングする、という手順である。

このようにして構築した制御システムの検証（予測）は制御が元々時間的な変化に対応するのが基本なので、何らかの動作シミュレーションで行われる。この場合ソフトウェアは元々コンピューター上で作動するものなので、まずは個々のモジュールが正しく動くか（いわゆるバグという品質確認も含めて）を検証する。個々の確認が取れば、制御システムとして製品機能（精度や追従性）が実現できているかを検証するのであるが、これをコンピューター上で行うためには、製品ハード側の動作を表現するプラントモデルが必要である。そして、制御とこのモデルを組み合わせ、最初に想定したいろいろなシーンで検証を行う。

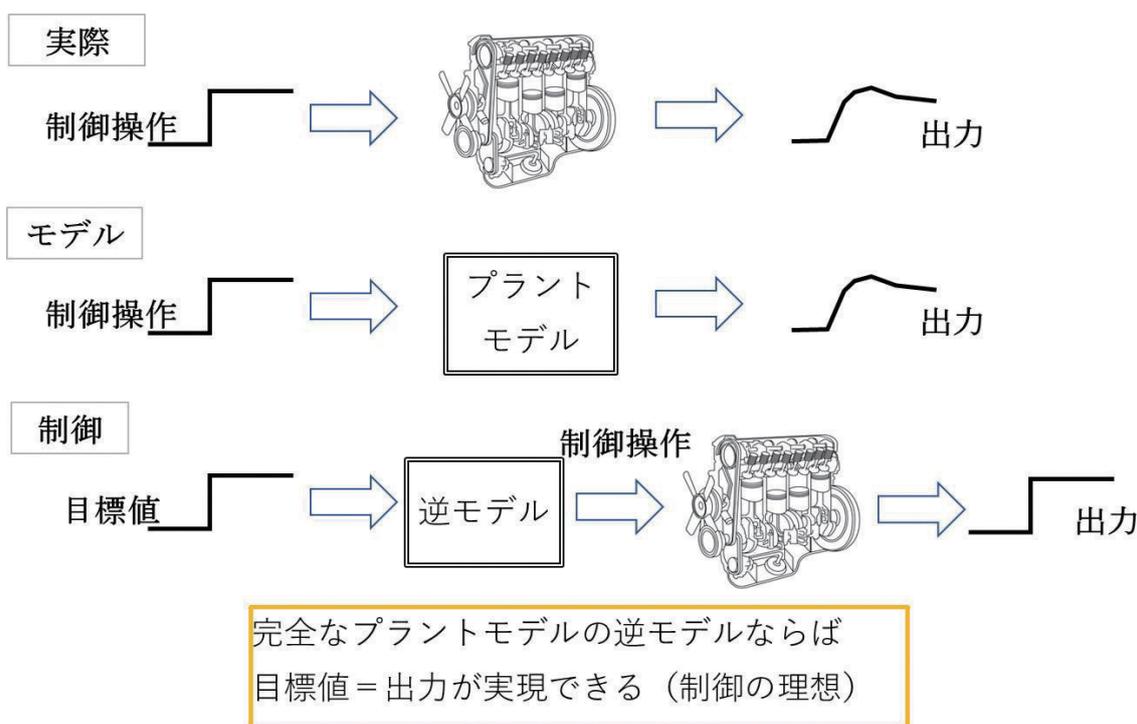


図3. 制御とプラントモデル

電子制御システムのソフトウェアにおける各機能を表現するモデルの多くは理論のみで構築されるものは少なく、実験的に求められたもの(実機テストやリグテスト)が多い。この実験式を少しでも理論に置き換えていくことが、実験を減らし、実験式の係数(データ)を決める作業を減らすので、効率化の観点からも重要である。

## 2-4 MBDの効果とV字プロセス

システム構築の際にその全体像としてV字プロセスで表現することが多いが、いわゆる製品開発も製品というシステムを作るので、適用が可能である。図4に従来の製品開発のプロセスを示すが、左側が設計プロセス、右側が実験プロセスになる。過去の経験や基準に従い、一部有識者のデザインレビューを受けつつ設計し、モノを作って評価し、NGがあればまた設計から。コンセプトレベル(構想設計)に立ち戻ることもあり、時間と費用のロスは大い。MBDの場合、この1つずつのプロセスで小さく予測やシミュレーションを行い、設計していき、その個々の機能目標に照らして、部品⇒ユニット⇒製品と実験評価の範囲を広げていくため、もし問題が見つかったとしても、後戻りところも小さく、時間/費用のロスを最小限に留められる。何より設計段階での検証をやり切れば、試作して実験で確認(評価、つまり課

題抽出ではなく)して完了とすることができる。これがMBDの最大の効果である<sup>(2)</sup>。

## 3. モデル事例

### 3-1 事例1 (簡単なモデル)

MBDにおけるモデルの一例として、実際のエンジンで採用された電子制御のプラントモデルを紹介する。このモデルは制御構想や基本仕様(ハード)の検討に用いただけでなく、ソフトウェアの目標値に対する操作量の決定のためにも用いられた。

この事例のシステムの全体図を図5に示す。エンジンに備えられた機械式の過給機(コンプレッサー)から吐出される空気を上流にリリーフするバルブを電子制御し、エンジンの過給圧を調整するシステムとなっており、このリリーフバルブ(エアバイパスバルブ、以下ABVと記す)を開くと過給圧が低下する。過給圧が上がればエンジンが吸入する空気量が増し、トルクが増加するため、基本的にはスロットル(アクセル)開度に応じて、過給圧を制御するシステムである<sup>(3)</sup>。単純にスロットル開度に追従するだけであれば昔からある機械式のABVを用いることも可能であったが、このときはアクセルを開いている状態でエンジントルクを低下させる

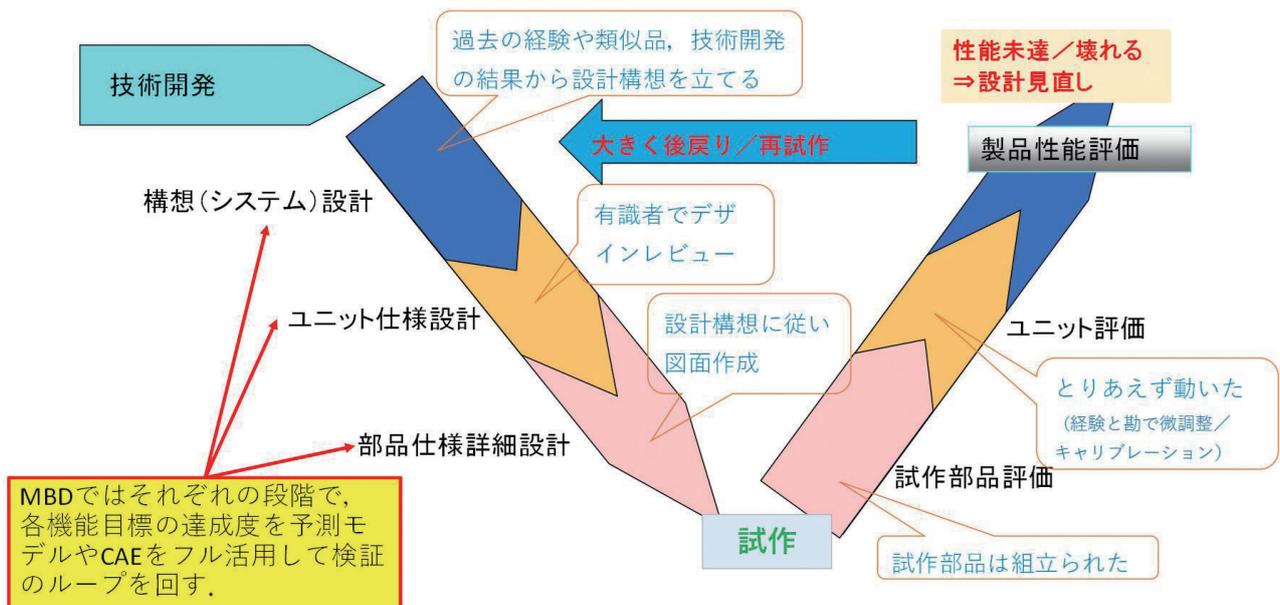


図4. V字プロセス

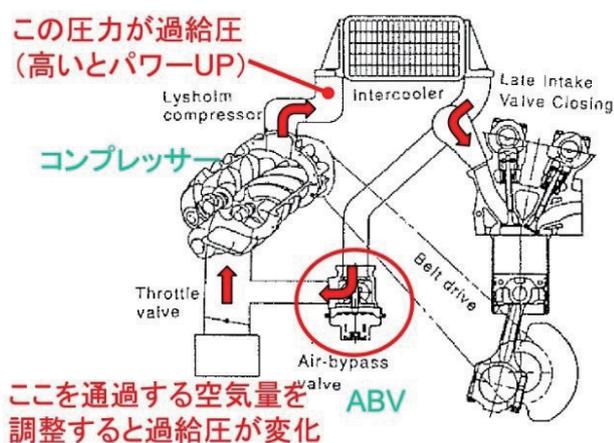


図5. 過給システム

要求機能を満足する必要があったため、電子制御化した（雪道でタイヤがスリップした際、エンジントルクを下げスリップを回復するトラクションコントロール等の要求があった）。

図6がこのプラントモデルになるが、基本的に定常での空気流量のつりあい（質量保存則）とオリフィス流れ（ベルヌーイの定理）の式で構成されており、定常であるため微分すらでてこない。後はABVやコンプレッサーの部品特性のリグテストデータでモデルは構築している。各部の空気流量と圧力ぐらいがパラメータで、若干のチョーキングに関する知識は必要だが、ほぼ四則演算でまかなえる程度の極めて簡単なものである。しかし、最終的にエンジンテストでのチューニングは必要だったが、自動車として市販するのに十分な精度を有す制御システムとすることができた。また、初期の検討段階で過給圧を制御するためABVにバルブ開度ストロークセンサーを取り付けて精度を狙う案があったが、モデルで検討したところ、そのストロークが0.5mm程度の有

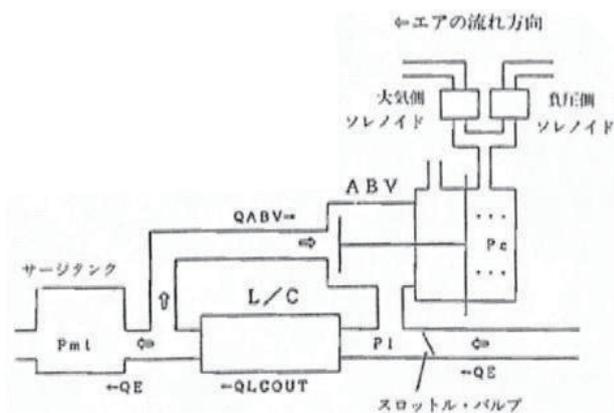


図6. プラントモデル

効範囲しかないことがわかり、採用せずに、背圧制御方式に決定するなど、システム方式の選定にも大きく寄与できた。

このように極めて簡単なモデルでも、事前検証が充実し、ほぼ確認のみで完成させることができる。にもかかわらず、なかなかこのモデル化が定着しない、大学で学んでも実践できないのが現状である。これは物理や工学を学ぶ際、具体的事例とはいえ、理想的なものが多く（例えば力のつりあいならば、摩擦がない、とか流体であれば非圧縮で非粘性とか）その前提条件が現実から乖離しているものが多い。実は身の回りの現象もかなりこれら理想的な物理や工学の法則で近似できるものは多数あるのだが、そういう事例はあまり示されていないので、具体的にどう適用するかがわからない。どこは理想モデルが使えるのか、実際の講義の中では（水力学のような実在流体を対象にしたものを除き）あまり触れられないということに課題があると考えられる。

### 3-2 事例2 (制御工学適用モデル)

近年のガソリンエンジンでは排気ガスの有害成分を除去（浄化）するため触媒が排気経路に取り付けられている。これは実に99%以上の浄化率を実現しているが、そのためにはエンジンが吸入する空気と燃料の比を理論混合比に制御する必要がある。精度的には数%程度の誤差しか許容されないため、電子制御システムとしてはいかに正確に吸入空気量を計測するかが重要となる。このため吸入空気の流れを計測するセンサー（エアフローセンサー：以下 AFS と記す）が設けられている。しかし、図7に示すシステム構成のように通常吸気通路の最上流に設置されるため、その下流の吸気経路容積の圧力変化分の空気も含めて計測することになる。ガソリンエンジンの場合、スロットルを閉じている場合は大気圧の半分程度までその下流の圧力は低下（エンジン負圧）しているが、スロットルを全開にすると、この圧力は大気圧付近まで上昇（自然吸気エンジンの場合）し、その差

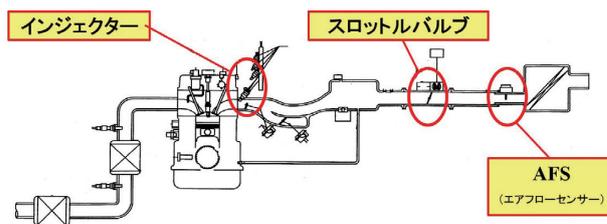


図7. エンジンシステム図

分の空気量も AFS は計測してしまう。従って図8に示すように AFS の出力信号に応じて燃料を供給してしまうと、加速時は大幅に混合気が過濃(燃料過多)となる。

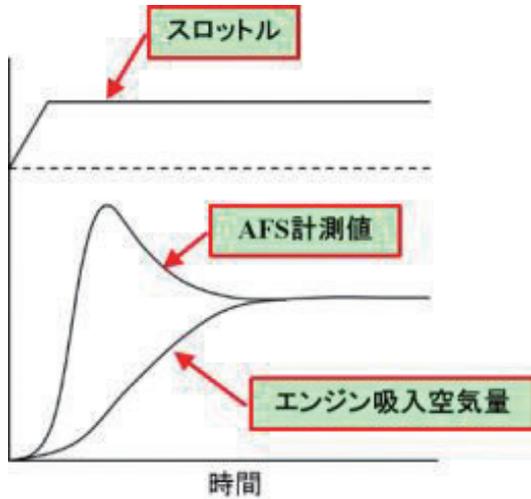


図8. AFS 計測誤差

このため、いろいろな実験式を作り、エンジンのテストを繰り返しながら式の修正やそのデータのキャリブレーションを行って、何とか理論空燃比付近で制御できるような作業を必要としていた(実験検証型の開発)。

これに対し図9に示すようなモデルを考える。

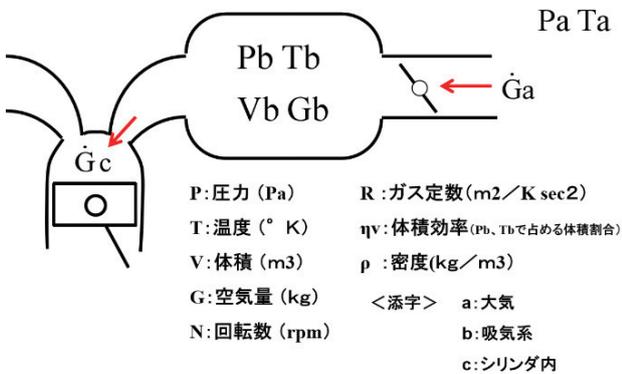


図9. プラントモデル

このモデルの運動方程式は、連続の式から

$$\dot{G}a = \dot{G}b + \dot{G}c \quad \text{①}$$

シリンダー内流入量は

$$\dot{G}c = Pb \times \eta_v \times Vc \times \frac{N}{60} \times \frac{1}{Rb \times Tb} \times n \quad \text{②}$$

n: 1 回転あたりの点火数

吸気系の状態方程式より

$$Gb = \frac{Pb \times Vb}{Rb \times Tb} \quad \text{③}$$

ここでスロットル開口面積が変化し、

$$\dot{G}a = \dot{G}a0 + \Delta \dot{G}a \quad Pb = Pb0 + \Delta Pb \dots$$

となったとすると(微小変化と考え T, N,  $\eta_v$  は一定、変化中スロットル開口面積も一定とする)

$$\text{①式は} \quad \Delta \dot{G}a = \Delta \dot{G}b + \Delta \dot{G}c$$

②式は4気筒とすれば

$$\Delta \dot{G}c = \Delta Pb \times \eta_v \times Vc \times \frac{N}{60} \times \frac{2}{Rb \times Tb}$$

$$\text{③式は} \quad \Delta Gb = \frac{Vb}{Rb \times Tb} \times \Delta Pb$$

これらの式をラプラス変換して整理する。

$$\Delta \dot{G}a(s) = s \Delta Gb(s) + \Delta \dot{G}c(s)$$

$$\Delta Gb(s) = \frac{Vb}{Rb \times Tb} \times \Delta Pb(s)$$

$$\Delta \dot{G}c(s) = \Delta Pb(s) \times \eta_v \times Vc \times \frac{N}{60} \times \frac{2}{Rb \times Tb}$$

$$\Delta \dot{G}c(s) = \frac{1}{1 + \frac{Vb \times 30}{\eta_v \times Vc \times N} \times s} \times \Delta \dot{G}a(s)$$

$$T = \frac{Vb \times 30}{\eta_v \times Vc \times N} \quad \text{とおくと}$$

$$\Delta \dot{G}c(s) = \frac{1}{1 + T \times s} \times \Delta \dot{G}a(s)$$

以上のように連続の式と気体の状態方程式から、制御工学における一次遅れの伝達関数で表すことができる。

一方制御モデルはできたが、このままではソフトウェアにできないため z 変換を用いてサンプル値系にする。z 変換の概念は図10に示すもので、 $\Delta t$  秒毎にサンプルして 0 次ホールドを用い、 $G_H(s) \times G(s)$  を z 変換すると、



図10. z 変換の概念

\*  $G_H(s)$  : 0次ホールドの伝達関数

$$G_H(s) = \frac{1 - e^{-s\Delta t}}{s}$$

$$Z[G_H(s) \times G(s)] = Z\left[\frac{1 - e^{-s\Delta t}}{s} \times \frac{1}{T \times s + 1}\right]$$

$$= \frac{1 - e^{-\frac{\Delta t}{T}}}{z - e^{-\frac{\Delta t}{T}}}$$

$$e^{-\frac{\Delta t}{T}} = K \quad \text{とおくと}$$

$$\Delta \dot{G}c(z) = Z[G_H(s) \times G(s)] \times \Delta \dot{G}a(z)$$

$$\Delta \dot{G}c(z) = \frac{1 - K}{z - K} \times \Delta \dot{G}a(z)$$

$$(1 - z^{-1} \times K) \times \Delta \dot{G}c(z) = (1 - K) \times z^{-1} \times \Delta \dot{G}a(z)$$

$Z^{-1}$ は1サンプル前の値を示すので

$$\Delta \dot{G}c[(n+1)\Delta t] = K \times \Delta \dot{G}c[n\Delta t] + (1 - K) \times \Delta \dot{G}a[n\Delta t]$$

このように差分形式にすることで、実際の制御ソフトウェアに実装することができるようになる。

尚、最近ではZ変換を用いず、微分を差分で近似して、直接運動方程式を実装する(現代制御ではこの手法が多い)、もしくは連続系の伝達関数のs項は微分を表すことから、これを差分形式に置き換えて、実装できる式を作る方法が主流である。例えば、微分を

$$\frac{d}{dt}G(t), s \times G(t) \rightarrow \frac{G((n+1)\Delta t) - G(n\Delta t)}{\Delta t} \quad \text{と近似し}$$

一次遅れの式の場合、これを用いて変形すると

$$(1 + T \times s) \Delta \dot{G}c(s) = \Delta \dot{G}a(s)$$

$$\Delta \dot{G}c(s) = \Delta \dot{G}a(s) - T \times s \times \Delta \dot{G}c(s)$$

$$\Delta \dot{G}c(n\Delta t) = \Delta \dot{G}a(n\Delta t) - T \times (\Delta \dot{G}c((n+1)\Delta t) - \Delta \dot{G}c(n\Delta t)) / \Delta t$$

$$\Delta \dot{G}c((n+1)\Delta t) = \left(1 - \frac{1}{T/\Delta t}\right) \times \Delta \dot{G}c(n\Delta t) + \frac{1}{T/\Delta t} \times \Delta \dot{G}a(n\Delta t)$$

と、z変換を適用した場合と同じ形の式を得ることができる。このとき、各項の係数が異なっているが、実際に  $\Delta t$  (サンプル周期) が小さければ

$$K = e^{-\frac{\Delta t}{T}} \cong 1 - \frac{\Delta t}{T}$$

と、この係数がz変換適用時の近似値になっており、最近のコンピュータでの実装において、差分を用いても精度的には問題がない。z変換は今回用いた0次ホールドの他に一次(三角)ホールドを用いる方法もあるが、この場合、サンプル周期が大きくても、精度を向上させられるため、処理能力の低いマイコン等に実装する場合は有効である。

運動方程式ができ、微分形式に持ち込めれば、ラプラス変換がわからなくとも差分形式にすることで、モデルの作成は容易である。しかし、このモデル作成が難しいと思われるのが現状で、これについて、モデル作成の各STEPのどこに壁があるか考えてみる。

一般にモデルの作成から制御構築のSTEPは

- ①力のつり合いなど、物理や工学の法則に従って運動方程式を立てる。
- ②微分方程式に置き換え、差分形式に持ち込む。
- ③シミュレーションにプラントモデルとして用いる。
- ④状態モデル等にして、制御工学を用いて、制御ソフトウェアを構築する。

このうち①での課題は事例1で述べたとおりであり、④はソフトウェアのレベルによっては直感的な制御(つまり制御工学を必要としない)でも十分な場合もあるため、ここでは②のSTEPに着目したい。

通常、高校物理では運動方程式などを学ぶが、その中で微分形式は登場しない。しかし大学では図11のように、前提として微分形式で記述され、その微分方程式を解くことで物体の運動などを算出する。つまり、大学入学直後から物理は微分形式でスタートするも、数学の講義では、その時点で微分方程式を教わっていないため、物理や工学の最初の内容である質点系の力学ですら理解できないということが生じる。これは、基本的な概念(図12)を身につけるタイミングがないということであり、このため数学と物理が全く連携していないまま、専門工学を学ぶということになる。従って、まずはこの基本的な概念をしっかりと腹落ちさせ、運動方程式と微分方程式が等価であると理解させることが必要だと考える。



図11. 運動方程式と微分

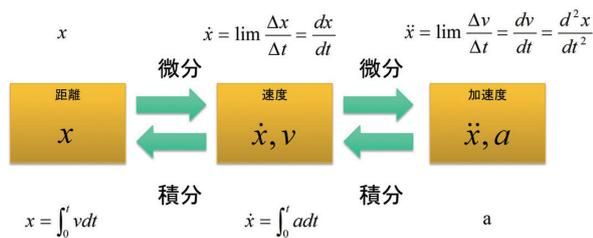


図12. 微分・積分の基本的な概念

3-3 事例3 (制御工学の概念)

MBDにおけるモデル化には直接寄与する事例ではないが、制御工学で学ぶべき重要な概念、「周波数軸で考える」、ということについての具体例を示す。

一般的に人は応答遅れのあるシステムを操作するとき、本能的にその応答遅れを解消するような操作を行う。例えば、部屋が暑くて冷房をかける場合、スイッチをONにしてもすぐには冷えず、希望の温度に下がるまで時間がかかる(応答遅れ)。このため自然に冷房能力をMaxにして室温が少しでも早く希望の温度になるように作動させる(応答遅れへの対応)。このような操作は「進み補償」と呼ばれるが、これを単純に制御で用いるとどうなるかを示したのが図13のAである。ここに示すのは3-1節で述べたABVで、エンジンの負圧を使って作動するダイヤフラム式のアクチュエーターだが、負圧の通路を開いてダイヤフラム室が負圧になるとバルブが引っ張られ開く仕組みである。このとき負圧の経路を開いてもダイヤフラム室の容積のためすぐに狙いの負圧値に到達せず(狙いの開度にならない)、応答遅れが生じる。この解消を果たすべく、ここに進み補償を入れたが、周波数特性という概念が全くないままに採用したため、一般的に高周波であるノイズ成分を拡大してしまい、大変動をもたらす結果となった。センサーからの信号には高周波のノイズが重畳されているのは常識であるが、進み補償が高周波を増幅するということを知らず、いろんな信号は多くの周波数を持つという概念がな

かったため、大きな失敗をした。結果的にローパスフィルタを前段に入れ(図13のB)、ノイズ除去を行ったが、制御工学を理解していない者が見れば、式的には「一次遅れ」×「一次進み」とキャンセルされるようなものになり(実際は周波数特性が異なる)意味が無いように見える。しかし、この方法で応答遅れを大きく解消し、狙いの制御を実現した。これも非常に簡単ではあるが、いかに制御工学の適用が優れているかを示す事例である。

4. エンジニア像

MBDを実践していくエンジニアの育成において、制御工学は必ずしも必須ではないが、プラントモデルを構築するスキルという観点からは非常に近いものがあり、そのため制御に携わる人材がMBDを推進していくことになる。これは筆者の企業での経験からも言えることである。ではこの制御工学の位置づけを考えると一般に図14のような関係があり、学問的なスキルとしては大学1年生レベルの数学・物理に加え、機械であればいわゆる4力の基本を理解していることが必要になる。



図14. 制御工学の位置づけ

つまり、制御対象(コントロールしたい対象物)をモデル化(微分方程式/状態方程式)し、制御理論を適用する。このモデル化のために対象物に見合った基本的な学問を理解することが必要になるということである。

一方、学問的なスキル以外で重要なこととしては、MBDのスタートラインである、機能展開を各開発ステージにおいて、適切な粒度(適切なモデル精度)で実施できるよう、機能要件の明確化/層別/機能目標の設定(ブレークダウン)を実施、もしくは推進する機能開発のスキルである。これは一般的には概念構成力とも呼ばれる論理的思考が求められ、大学の講義だけでは身に着けられない。何らかの課題に対し、自ら考え解決を導いていく場面で鍛えられていくもので、問題解決能力とも言われるが、大学でこれに近い場面というと、大学院での研究が相当するであろう(学部の研究は、教員や院生の指導で進められ、教えられる立場から脱却しないため不十分)。あとは課外活動など、企業が社員の採用試験で重視しているポイントと同じである。

また、この機能展開に限らずモデル作成の場面におい

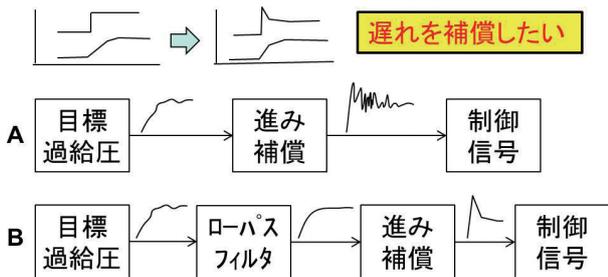


図13. 進み補償

でも必要になるのが、対象物をより深く理解することができる本質追求のスキルである。同じく論理的思考であるが、何故そうなるかを自分が納得いくまで追求し、事実ベースで、思いを排除した分析を行うことができる、という能力になるのだが、これはエンジニアの基本である「現場・現物」をいかに大事にするか、ということである。この本質を探求する目の育成の簡単な例として、自動車の実験の準備段階に行う図15のようなタイヤ脱着の作業で考えてみる。このとき、以下の3点に着目させ、その理由を考えさせる。

- ①ハブレンチが中途半端に短い。(標準的な人が端に体重をかけると、規定トルクになる長さに設定)
- ②ホイールナットの片方がテーパ形状。(ナットを締めるとセンタリングする)
- ③ホイールナットを対角線で締めていく。(1点のみ締めてしまうと中心がずれる、もしくは歪む)



図15. タイヤ脱着作業

このように、いろんなものがさまざまな理由で形状などが決められていることに気づかせ、同様のものがないかを探るように促していくことで、少しずつ成長していくと考える。

ここであげた、3つのスキルを備え持つエンジニアがMBDに限らず、ものづくりを行っていく上での理想像で、特に最初の学問以外は、学ぶ姿勢からは身につかないものであり、教えられるものではない。各個人が自発的に仕事や各種の活動の場において、これらの考え方を意識して訓練しないと体得できないので、この育成、育つ環境づくりは非常に難しい。

5. まとめ

MBDは、質と効率の両方を向上する開発手法であり、自動車に限らず、日本の得意とする「摺合せ」をモデルで行うことが、これからの「ものづくり」の切り札。そのプロセスは以下ようになる。

- ✓対象システム(自動車など、製品)で起こる全ての現象を解明(からくり解明)し、モデル化を行い、
  - ✓机上で種々の条件の元、対象システムを動かす。
  - ✓高い目標に対し、その実現に必要な機能を考え、目標機能を配分し、機能実現の手段をモデルを用いて、徹底的に机上検証する。
  - ✓全ての目標を達成するブレークスルーすべき機能について、集中してその技術を開発(一番ピン=選択と集中)する。
  - ✓最後に、実機でその確からしさを一発検証する。
- このプロセスを実践できる人材となるには、学問の習得

### ＜プラントモデル＞

**物理数学基礎～初級**  
(高校物理が怪しい人)

1. 数学の基礎	三角関数 指数と対数 単位/複素数 距離速度加速度/微積分
2. 物理基礎(運動)	2. 力と仕事とは 3. 運動方程式
3. 熱力学基礎	状態方程式 比熱・エネルギー 比熱・エネルギー 演習2 比熱・エネルギー 演習3 比熱・エネルギー 演習4
4. 流体力学基礎	非圧縮 圧縮性

### ＜制御＞

**古典制御～数学と物理と実際の式**  
(制御理論を電子制御に入れたい)

①物理と実際	バネマス系の物理と実験(MATLAB検証) OR回路の物理と実験(MATLAB検証)
②制御用数学	極限と微分、物理の速度/加速度 積分、物理の加速度/速度/距離 微分方程式と物理 ラプラス変換と微分方程式の解法 行列/行列式/固有値
③古典制御基礎	伝達関数とステップ応答 2次とインパルス応答 伝達関数とブロック線図 周波数応答と実験 安定性とふりかえり

高次系も積分要素(積分器)と比例要素(係数)で表現できる

図16. エンジン制御開発に必要な学問

と論理的思考の成長が必要である。学問としては広範な物理数学、工学の習得が必要（図16にエンジン制御開発の例を示す）だが、いずれも基礎的な範囲で十分である。この育成における課題は物理と数学の融合であり、これは大学での授業カリキュラム等で重点ポイントとすれば十分カバーできるものである。

学問分野以外の領域、論理的思考についてはその対象システム（開発する製品など）の機能、つまり具備すべき要件を正しく抽出、かつ階層的に展開できるという力と、常に物事の本質、メカニズムを探求することが求められる。基本的に教えることが主体の大学の授業で習得することは、まず難しいが、一部、実験／実習において「思い込み」を捨て「何故」そうなるかという姿勢を継続させることで鍛えられると考える。いわゆる「現場・現物」に徹するということである。企業の採用では「仕事力」の高い人材を求めているのであるが、理系では大学院修了生が採用試験の上位を占めている状況は、まさにこのことを物語っていると言える。

## 6. 最後に

MBD を実践できる人材の育成は、大学だけでは難し

く、実践的な場である企業での仕事において、大きく成長する。しかし、大学においてもいろんな課外活動も含め、そのきっかけを醸成することは可能であり、今後世の中に貢献する人材の育成として取り組んでいく。また、非常に効果的な環境である大学院への進学も推進していく所存である。

## 参考文献

- (1) JAMBE, “JAMBE | JAMBE 紹介動画”, <https://youtu.be/ALAMaCOOdyI> (2024年3月28日)
- (2) 電通総研, “電通総研 | ブログ | ソリューション | MBSE とシミュレーションをつなぐMBD |”, <https://mfg.dentsusoken.com/blog/detail/001574.php> (2024年3月28日)
- (3) Ryo Shimizu, Masaki Fujii, Takasi Suzuki, Masao Inoue, Susumu Niinai, “Miller cycle engine management system and its distinctive feature.”, *JSAE Review*. 15.4 P305-308. 1994

---

(2024年6月26日受理)