

モデルベース開発におけるプロダクトライン可変性モデルの提案

M2009MM005 平野 将貴

指導教員 青山 幹雄

1. はじめに

組込みシステムは、大規模化、複雑化による開発期間延長やコスト増大などの問題がある。この対応策として、モデルベース開発やプロダクトライン開発が注目されている。本稿では、プロダクトライン開発に基づき、制御システムの可変性を表現するモデルを提案、評価する。

2. 研究の背景

(1) モデルベース開発

モデルベース開発は、主に連続系の制御システムを対象とする開発方法である。システムを抽象度の高い制御システムモデルで記述して開発する。連続系制御システムモデルはブロック線図で表現する。ブロック線図は、計算を行う基本ブロックと基本ブロック間の信号の流れを表現する矢印で構成される。制御ロジックの設計を、ブロック線図を用いて設計し、ツールを用いてソースコードに自動変換する。本稿では、Math Works 社が開発している MATLAB/Simulink を使用する。

(2) プロダクトライン開発

プロダクトラインは、ソフトウェアの組織的な再利用を行うための開発方法である。コア資産と呼ばれるソフトウェア群を作成し、それらを組み合わせることによって各ソフトウェアを開発する。コア資産は、開発するすべてのプロダクトで共有される共通部と、各プロダクトに特化した可変部がある。共通部と可変部を表現したものを可変性モデルと呼ぶ。可変性モデルは可変点と変異体から構成される。可変性モデルを基にコア資産を開発する。

3. 研究課題

本稿では、連続系制御システムの共通部と可変部を定義し、表現方法として可変性モデルを提案する。連続系制御システムの可変性定義の課題は以下の3点である。

(1) 制御システムの共通部と可変部の分離

複数のプロダクトでコア資産として共有する共通部と、各プロダクトを特徴づける可変部を定義する必要がある。

(2) 開発プロセスの定義

モデルベース開発にプロダクトライン開発を適用した開発プロセスを定義する。

(3) 可変性モデルの定義

制御システムの可変性を明示するために、制御システムと可変性モデルを同一の図で表現する方法を定義する。

4. 関連研究

本稿に関連する研究として以下の二つがある。

(1) 可変性を持つ制御システムモデルの提案

制御システムモデルの部品を組み合わせ、可変性モデルにおける変異体として構成する方法を提案している。制御ロジックを表現した制御システムモデルと可変性モデルを一つの図で表現できるので、二つのモデル間の関連が表現できる(図1)。

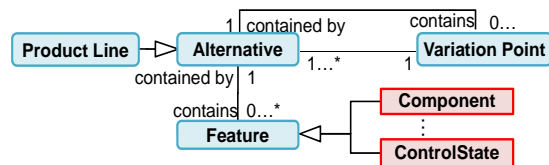


図1 制御モデルの要素を付加した可変性メタモデル

(2) Simulink モデル内での変異体選択実装法

制御システムのコンポーネント内に複数の変異体を構築する。そして、その変異体を選択するための制御ブロックを設ける[1]。制御ブロックのパラメータをシミュレーション前に設定することで、コンポーネントの振る舞いを決定できる。

5. アプローチ

(1) 共通部と可変部の定義

ブロック線図は、ブロック構造とブロック毎のパラメータから構成される。ブロック構造とは、基本ブロックを組み合わせたものである。基本ブロックの組み合わせによってどのような制御を行うかが決まる。パラメータとは、各基本ブロックに定義される変数である。パラメータを設定することで、基本ブロックでどのような計算が行われるかが決まる。

本稿では、ブロック構造とパラメータをまとめたものを System Model と定義し、プロダクトラインの共通部とする。

一方、可変部は各パラメータの値と定義する。プロダクト毎にパラメータを設定することで、個別の振る舞いをするプロダクトが開発できる。

図2は、System Model の例を示している。プラントモデルのパラメータ M を設定するとプラントモデルの設定を変更できる。コントローラモデルの各パラメータを設定することでプラントモデルの出力を適切な値に制御できる。各パラメータ値を変更すると、同一の System Model を用いて異なるプロダクトを開発できる。

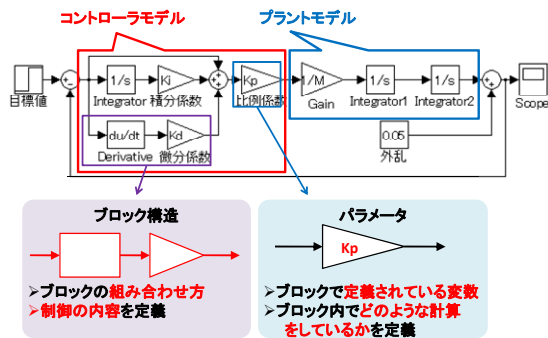


図2 連続系制御システムのブロック構造とパラメータ

(2) 可変性の表現方法

可変性を表現するために可変性モデルを提案する。可変性モデルは、図1のメタモデルを拡張し、連続系制御システムプロダクトラインの共通部と可変部を表現できるようにしたものである。Product Line要素にSystem Model要素を関連づけて共通部と定義し、変異体として定義されているFeature要素にパラメータ値を継承させて可変部として定義する。

6. モデルベースプロダクトライン開発

6.1. 開発プロセス

6.1.1. 開発プロセス概要

本稿で提案する開発プロセスは、文献[3]で提案されている開発プロセスをモデルベース開発に適用するために拡張したものである(図3)。開発プロセスは、コア資産となるSystem Modelと可変性モデルを開発するドメイン開発プロセスと、各プロダクトを開発するアプリケーション開発プロセスで構成される。

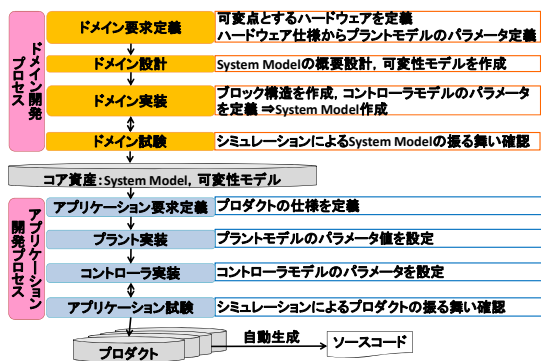


図3 モデルベースプロダクトライン開発プロセス

6.1.2. ドメイン開発プロセス

(1) ドメイン要求定義サブプロセス

ドメイン要求定義では、製品ロードマップを作成し、制御対象の制御目標を定義する。交換可能なハードウェアのパラメータを定義し、値の範囲を定義する。このハードウェア

が可変点となる。一方、各プロダクトで共通利用するハードウェアのパラメータは定数で定義する。可変点となったハードウェアと、そのパラメータを可変性モデルとして表現する。これらのプロダクトラインに共通な要求仕様をドメイン要求仕様書としてまとめる。

(2) ドメイン設計サブプロセス

System Modelにおけるブロック構造の概要設計書と可変性モデルを作成する。ブロック構造の概要設計は、制御システムモデルの各コンポーネントを定義して配置する。またブロック構造に定義されているコントローラモデルのパラメータを可変性モデルに追加する。

(3) ドメイン実装サブプロセス

ブロック構造の詳細設計を行い、Simulinkを用いてコンポーネントを実装する。ブロックにプラントモデルパラメータとコントローラモデルパラメータを定義し、System Modelを作成する。

(4) ドメイン試験サブプロセス

System Modelの各パラメータにドメイン要求定義サブプロセスで定めた範囲内のプラントモデルパラメータ値を設定する。シミュレーションを行い、振る舞いが要求仕様を満たすか確認する。

6.1.3. アプリケーション開発プロセス

(1) アプリケーション要求定義サブプロセス

製品ロードマップと可変性モデルから、各プロダクトのハードウェア仕様やシステムの性能を抽出する。これを基に、ドメイン要求仕様書において開発するプロダクトに特化した部分を拡張し、各プロダクトの要求仕様を定義する。これをアプリケーション要求仕様書としてまとめる。

(2) プラント実装サブプロセス

各プロダクトの要求仕様を基に制御対象のハードウェアを選択し、プラントモデルのパラメータを設定する。

(3) コントローラ実装サブプロセス

プラントモデルの振る舞いがアプリケーション要求仕様を満たすようにコントローラモデルパラメータ値を設定する。

(4) アプリケーション試験サブプロセス

シミュレーションを行い、プロダクトの振る舞いがアプリケーション要求仕様を満たすことを確認する。振る舞いが要求仕様と異なる場合は、コントローラ実装サブプロセスに戻り、コントローラパラメータを設定し直す。

6.2. 提案する可変性モデル

6.2.1. 可変性モデルのメタモデル

本稿で提案する可変性モデルは、文献[4]で提案されている可変性モデルを拡張して定義した(図4)。可変性モデルには、可変点と可変点における変異体を定義する。さらにSystem Modelにおいて可変性に関するプラントモデルパラメータとコントローラモデルパラメータを定義する。

制御システムモデルと可変性モデル間の関係を表現し、両モデルを対応付けるモデルとして定義した。

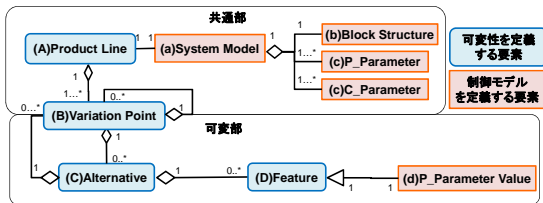


図4 提案する可変性モデルのメタモデル

6.2.2. 可変性メタモデルの構成要素

(1) 可変性を定義する要素

(A) Product Line 要素

開発するプロダクトラインを示す。コア資産である System Model と System Model における複数の Variation Point とそれに対する Alternative から構成される。Product Line 要素は、各プロダクトの共通部である。

(B) Variation Point 要素

制御システムの変異点を示し、変異体となる複数の Alternative または Variation Point を持つ。Variation Point は、各プロダクトの共通部である。しかし構成要素として Variation Point を持つ場合、最上位の Variation Point のみ共通部となり、それ以下の Variation Point は可変部となる。

(C) Alternative 要素

Variation Point で選択する変異体を示す。Alternative は、ハードウェアのパラメータを示すので複数の選択肢から一つだけ選択される。よって Alternative は、相互排他的な変異体となる。

(D) Feature 要素

変異体となる制御システムモデルの要素を示し、Alternative の構成要素として定義する。Feature 要素はハードウェアのパラメータ値を示す。

(2) 制御システムを定義する要素

(a) System Model 要素

5章で定義した System Model を示す。文献[4]では、System Model は Component として定義されているが、本稿では、各プロダクトで同一の System Model を利用するので共通部と定義する。

(b) Block Structure 要素

5章で定義したブロック構造を示す。ブロック構造は一つの System Model に対して一つ存在する。System Model と同様に共通部である。

(c) P_Parameter 要素, C_Parameter 要素

P_Parameter は、プラントモデルのパラメータを示す。C_Parameter は、コントローラモデルのパラメータを示す。P_Parameter と C_Parameter 要素は、System Model の構成要素なので、共通部である。

(d) P_Parameter Value 要素

プラントモデルパラメータに設定される値を示す。この値によって制御対象の仕様がプラントモデルに反映される。

P_Parameter Value は Alternative の構成要素となるので可変部である。

6.2.3. コントローラモデル/パラメータ値が定義できない理由

本稿で提案する可変性モデルでは、コントローラモデルパラメータ値は定義できない。コントローラモデルパラメータ値は、プラントモデルのパラメータを設定後、シミュレーションを行い設定する。可変性モデルは、プラントモデルパラメータ値のバリエーションを定義するモデルである。プラントモデルパラメータ値を設定するプロセスではコントローラモデルのパラメータ値は設定できない。よってコントローラモデルパラメータ値は、可変性モデルで定義できない。

7. 可変性モデルの適用と評価

7.1. ライントレーサ制御システムへの適用

7.1.1. ライントレーサの仕様

制御対象となるラインレーサは、二つの車輪が車体に取り付けられていて、各車輪の回転数を調整して左右に曲がる[2]。ラインレーサの変異点となるハードウェアのパラメータは、車輪半径 R と車幅 W とする。ラインレーサにはライン検知センサがあり、センサからの情報によりラインに沿って走行する。

7.1.2. 可変性モデル作成プロセス

本ケーススタディでは、ドメイン要求定義サブプロセスとドメイン設計サブプロセスを行い、ラインレーサ制御システムプロダクトラインにおける可変性モデルを作成した。

(1) ドメイン要求定義サブプロセス

ラインレーサの制御目標は、センサからの情報に基づいてラインに沿って走行することである。

本稿で提案する可変性モデルは、交換可能なハードウェアが可変点となる。ラインレーサにおける交換可能なハードウェアは車輪と車体とする。その仕様である車輪半径 R と車幅 W をプラントモデルパラメータと定義する。ラインレーサのその他の仕様は、プロダクトラインで共通に利用されるので定数と定義する。可変点とプラントモデルパラメータ、パラメータ値の範囲を可変性モデルで表現する(図5)。

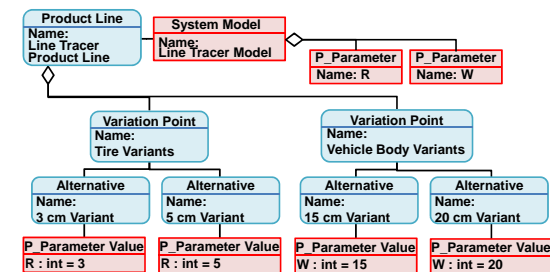


図5 Variation Point と P_Parameter を定義した可変性モデル

(2) ドメイン設計サブプロセス

ラインレーサ制御システムは、プラントモデルの出力を目標値と比較し、目標値との差を 0 にするフィードバック制

御を行う。プラントモデルの出力を Scope ブロックで表示できる。ライントレーサ制御システムのブロック構造の概要設計書を図 6 で示す。

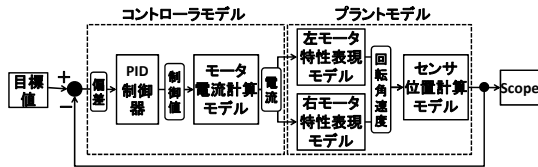


図 6 ライントレーサ制御システムモデルの概要設計図

コントローラモデルのパラメータとして PID 制御器の K_p , K_i , K_d を定義した。コントローラモデルパラメータとブロック構造を可変性モデルに追加する(図7)。

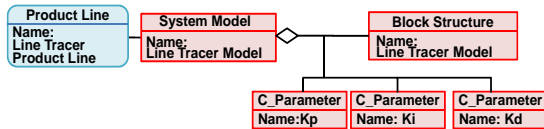


図 7 Block Structure と C_Parameter を定義したモデル

7.1.3. ライントレーサ制御システムの可変性モデル

ライントレーサ制御システムの可変性モデルを図 8 に示す。Line Tracer Product Line の共通部は、可変点である車輪半径と車幅と、System Model を構成する Line Tracer Model と R , W , K_p , K_i , K_d の各パラメータである。可変部は、Alternative と P_Parameter Value である。

Product Line には、二つの Variation Point が含まれる。各 Variation Point では、複数の Alternative を選択できる。Tire Variant では、車輪半径を 3cm から 5cm の間で変更可能である。Vehicle Body Variant では、車幅を 15cm から 20cm の間で変更可能である。

System Model の P_Parameter をハードウェア仕様に基づき設定する。シミュレーションを行い、System Model が要求定義通り動作するよう C_Parameter を設定する。

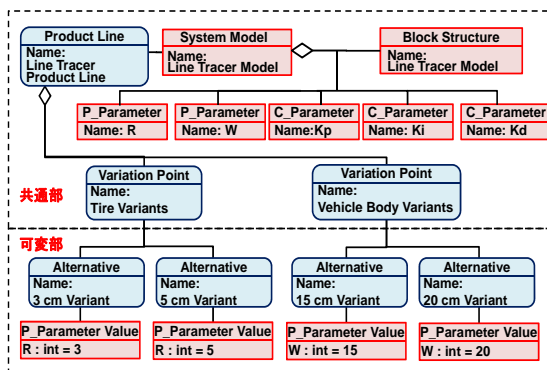


図 8 ライントレーサの可変性モデル

7.2. 評価

(1) 連続系制御システムの可変性を定義

ブロック構造とパラメータを共通部とし、プロダクト毎のパラメータ値を可変部とすることで、連続系制御システムの可変性を定義できた。可変部であるプラントモデルのパラメータ値を設定し直すことで、共通部である System Model の再利用が可能になった。

(2) 可変性モデルの表現能力

提案した可変性モデルは、制御システムの共通部と可変部を分離して表現できた。制御システムの可変性に関するパラメータを可変性モデルのパラメータと定義したことで、制御システムと可変性モデルの関連付けが可能になった。

7.3. 考察

提案した可変性モデルは、連続系制御システムにおける共通部と可変部を定義し表現した。このモデルをアプリケーション開発プロセスで利用することによって、何が共通に利用可能で、何に変更が必要なのかを理解しやすい。よって変異体の定義が容易になる。また可変性定義と制御システムモデルの定義を一つのモデルで表現できるので、制御システムモデルと可変性との関連が理解しやすい。

8. 今後の課題

(1) 可変性モデルの複雑化への対処

ハードウェアのバリエーションが増えた場合、可変点と変異体が複数階層に及ぶ。するとハードウェア同士の関係が複雑になり、プラントモデルパラメータ間に相互作用が発生することが考えられる。

(2) 開発プロセスの各サブプロセスの成果物の詳細化

本稿では、コア資産を定義する System Model と可変性モデルを定義した。しかしドメイン要求仕様書など、その他の成果物に関しての詳細は未定義である。

9. まとめ

本稿では、プロダクトライン開発に基づき制御システムの可変性をモデル化した。制御システムのブロック構造とパラメータに着目することで、共通部と可変部を定義し可変性モデルを作成した。提案した開発プロセスと可変性モデルをライントレーサ制御システム開発に適用し、評価した。

参考文献

- [1] C. Dziobek, et al., Functional Variants Handling in Simulink Models, Math Works Automotive Conference, 2008.
- [2] 平野 将貴, 杉浦 由季, オブジェクト指向を取り入れたモデルベース開発の提案, 南山大学 2008 年度卒業論文, 2009.
- [3] K. Pohl, et al., Software Product Line Engineering, Springer, 2005.
- [4] B. Schatz, et al., Product Lines and Model-Based Development, Electronic Notes in Theoretical Computer Science, Vol.182, 2007, pp. 171-186.