Vol.2012-SE-175 No.19
2012/3/16

ハードウェア性能に対する組込みシステムモデルの影響分析手法の提案

鈴木辰昇[†1]　古川覚[†1]
上田賀一[†1]　中島震[†2]

組込みシステムはセンサやモータといった複数のハードウェアから構成されており，限られたコストで目標性能を実現するにはこれらの性能が重要になる．本研究ではハードウェア性能の変更やハードウェアが想定していた性能を発揮できない場合の影響分析手法を提案する．SysML モデルに対しハードウェア性能に関する記述を追加する．SysML モデルをもとに作成した Yices モデルを Simulink モデルと連携させた検証を行う．検証結果からハードウェアの変更が引き起こしたモデルの影響を分析する．適用実験として，二輪型倒立振子ロボットを使用したハードウェア性能変更による影響分析実験を行う．性能をどこまで下げることができるのか分析し，実験の分析結果と提案手法について考察する．

An Analysis Method for Hardware Performance Effects in Enbedded Systems Model

Suzuki Tatsunori ,[†1] Furukawa Satoru ,[†1]
Yoshikazu Ueda [†1] and Shin Nakajima [†2]

The embedded system is composed of multiple components such as sensors and actuators. To achieve desired hardware performance at a limited cost, it is important that we choose the sufficient hardware components. This paper proposes an analysis method of hardware performance limit and unexpected performance effect like as hardware failure. In first, this study shows ways to state the hardware performance into SysML diagram, and explains ways to convert it to Yices model. Next, the way to verify the constraint of hardware performance in the combination of Yices model and Simulink model is shown. This analysis method is demonstrated in a case study of an inverted pendulum robot. And we analyze the availability of hardware with lower performance. Finally, we discuss the proposed method.

1. はじめに

今日の組込みシステムは非常に複雑で高性能な機能が求められており，開発も大規模なものになってきている．それにもかかわらず，実際の開発現場においては開発期間の短縮やコストの削減によって厳しい状況下での開発を余儀なくされている．

さらに，組込みシステムの場合はセンサやモータといった複数のハードウェアから構成されており，限られた予算で目標性能を実現するにはこれらの性能が重要になる．しかし，どの程度の性能を必要とするのか，選定したハードウェアで目標性能を実際に発揮できるのか確認することは難しい．また，故障などの影響により本来の性能を発揮できない場合もあり，その影響が大きくならないよう設計段階で致命的欠陥を排除する必要がある．

制御ソフトウェア開発で利用されている MATLAB/Simulink[1)2)] はハードウェアを含むシステムの設計・シミュレーションをサポートし，システムの性能解析に非常に有効である．しかし，シミュレーションで得られる結果だけではパラメータの組み合わせに限りがあり，致命的な故障を引き起こす組み合わせを見落としてしまう可能性がある．

先行研究[3)] では，SysML のダイアグラムのひとつでありシステムに存在する制約を可視表現できるパラメトリック図を利用し，定義した制約の充足可能性を検証する手法を提案した．SysML は制約を含んだシステムモデルを表現するだけおり，離散系/連続系を考慮した検査を行うことができないため，Simulink で連続系モデルを作成し，SMT ソルバである Yices と連携させることで，フィードバックループを含んだ充足可能性の検証手法を提案した．

Yices は変数，配列，関数などを定義し等式や不等式を列挙することで，それら制約を満たす例を求めることができる．これにより，シミュレーションでは見落としてしまうパラメータの組み合わせを網羅的に求めることができる．

そこで，本研究では設計段階で目標性能を実現できるか確認し，シミュレーションだけでは見落としてしまう問題を発見する方法を考察する．特に，先行研究で提案された Simulink と Yices の連携手法を拡張し，ハードウェア性能の変更による影響分析手法を提案する．SysML モデルに対しハードウェア性能に関する記述を追加し Yices モデルに変換する．Yices モデ

†1 茨城大学
Ibaraki University
†2 国立情報学研究所
National Institute of Informatics

© 2012 Information Processing Society of Japan

ルを Simulink モデルと連携させた検証を行い，検証結果からハードウェア性能の変更が引き起こしたモデルの影響を分析する．

また，ハードウェア性能の変更による影響の分析手法を応用し，ハードウェアが想定していた性能を発揮できない場合の影響分析を行う．本手法の適用可能性の確認のために，二輪型倒立振子ロボットを使用したハードウェア性能変更による影響分析実験を行い，実験結果について考察する．

本手法を用いることにより，ハードウェア性能の選定のために必要な影響分析，そして，分析結果をもとにした影響対策を支援できる．これにより，開発の効率化とハードウェア構成の最適化に貢献できると考える．

2. 関 連 知 識

2.1 SysML

SysML(Systems Modeling Language)[4] は，ソフトウェア，ハードウェア，情報，人，手続き，設備を含む複雑なシステムを定義，分析，設計および検証するための汎用的な図式モデリング言語である．システムの要求，振舞い，構造および制約をモデル化するために必要な図式表現を提供する．

2.1.1 制約ブロックとパラメトリック図

制約ブロックとパラメトリック図は，属性の間に制約を記述したものであり，その制約はシステムの性能や信頼性，整合性を確認するために使用できる．その中でも，制約とパラメータなどの定義を示したものが制約ブロックで，制約ブロックを使う状態 (インスタンス) を示したものがパラメトリック図である．

2.2 SMT ソルバ Yices

SMT(Satisfiability Modulo Theory) は個別理論を取り扱うように拡張した SAT ソルバである[5][6]．線形算術演算，等号と未解釈関数などの個別理論に関する決定手続きを SAT の枠組みに組込んだ充足可能性判定器であり，プログラムやシステムの有界モデル検査などのエンジンとして使われている[7][8][9]．

SMT ソルバの一つである Yices[10] では int，real，nat，bool のプリミティブ型や Function type や Scalar type といった様々な型が用意されている．また Yices 自身で新しい型を定義できる．式 (1) のように define 文を用いて変数を定義する．また式 (2) のような記述で定数を定義することができる．

(define [name] :: [Type]) (1)

(define [name] :: [Type] [expr]) (2)

2.3 SysML モデルの制約妥当性検証手法

本研究の先行研究として，SysML モデル内の制約妥当性検証手法が提案されている[3]．SysML ダイアグラムのひとつであるパラメトリック図はシステムに存在する制約を表現できるため属性の妥当性検証などに役立つと期待されている．しかし，モデル中の制約が複数のダイアグラムに分散してしまうため，システム内の制約が満たされているか確認するのに大きな手間がかかる．そこで，SysML モデルに記述された制約を抽出し SMT ソルバである Yices が解析できる形式に変換し検査する．さらに，Yices が扱う離散系モデルを制御ソフトウェア開発で利用される Simulink の連続系モデルと連携させることでフィードバックループを含んだ制約の充足可能性検査を行う手法を提案している．

3. 提 案 手 法

本章では，SysML・Yices・Matlab/Simulink でのハードウェア性能の記述方法とその影響分析手法について説明する．本手法では，ハードウェア性能の変化による影響分析のために，先行研究で提案された手法で記述された SysML モデルにハードウェア性能に関する記述を追加する．さらにモデルを Simulink・Yices に変換しそれぞれのモデルを実行することでハードウェア性能による影響を分析する．

3.1 ハードウェア性能の変化による影響

ハードウェア性能を変えた場合の影響を分析する．特に，性能を下げた場合でもシステムが問題なく動作するのか確認する．問題が発生する場合はどこに発生するのか，問題をソフトウェア側で対処できるのかを本手法を用いて分析する．本手法の実行手順は以下のようになる．

(1) 対象の SysML モデル (ブロック定義図・パラメトリック図・内部ブロック図) を作成する．
(2) ハードウェア仕様をもとに SysML モデルに変更するハードウェア性能を記述する．
(3) ハードウェア性能を記述した SysML モデルを Yices モデルに変換する．
(4) Yices に初期値を設定する．
(5) Yices で反例があるか調べる．
    5-1 反例がある場合，対象の SysML モデルは必ず充足せず，想定外の振舞いを起こす場合があると考えられる．

© 2012 Information Processing Society of Japan

5-2 反例が出力されなかった場合，検証性質箇所の assert 文の not をとり Yices で充足する例を出力する．

(6) 手順 5 の結果を set_param コマンドで Simulink へ設定しシミュレーションを続行する．

(7) シミュレーション結果が出力されたファイルの値を Yices へ設定し，手順 5 へ

(8) 手順 5 から手順 7 までで得られた結果を分析する．

(9) 分析結果から新たに分かったハードウェア特性を SysML モデルに追記する．

3.2 ハードウェア故障の影響分析

前節で提案した手法を拡張し，ハードウェアが性能を発揮できない場合の影響分析手法を提案する．

故障は下記の 2 つに大別できる．

- ランダムハードウェア故障
  - ハードウェアに関し部品や材料の劣化やばらつきなどにより確率的に発生する故障
- 決定論的原因故障
  - 設計の誤りなどの，特定の原因から決定論的に関係する故障

ランダムハードウェア故障は検出可能なものと不可能なものに分けられ，検出不可能な故障をモデリングすることはできない．そのため，本研究ではハードウェア故障の中でも「検出可能なランダムハードウェア故障」を対象とする．

本手法の実行手順は以下のようになる．

(1) 対象の SysML モデル (ブロック定義図・パラメトリック図・内部ブロック図) を作成する．

(2) SysML モデルにハードウェア故障に関する記述を追加する．

(3) Yices モデルに変換し初期値を設定して検証を行う．

(4) 故障時の制約充足例・非充足例をすべて Yices で出力させ影響と特性を分析する．必要に応じて Simulink の連続系モデルと連携する．

(5) Simulink モデルを用いて Yices の分析結果から想定される故障のシミュレーションを実行する．

(6) 検証結果とシミュレーション結果からモデルに対し故障対策を行う．

3.3 故障の表現方法

故障に関する表現を SysML モデル内に記述する際，通常時と区別する必要があるためステレオタイプ`<<detectionfailure>>`を定義した．故障時に特定のパラメータがステレオタイプ`<<detectionfailure>>`で定義したものに切り替わることを意味する．SysML で記述した Simulink での故障を表現するために，パラメータが故障時の値を取るブロックを追加する．正常時の値と故障時の値はマニュアルスイッチを用いて切り替える．`<<detectionfailure>>` を Yices モデルに変換する際は assert を用いる．

4. 適 用 実 験

本章では 3 章で述べた手法の適用実験を示す．実験には二輪型倒立振子ロボットモデルを利用する．

4.1 二輪型倒立振子ロボット

本研究で扱う二輪型倒立振子ロボットは，ジャイロセンサ，バッテリー，DC モータ 2 つと，モータの走行速度や旋回速度を設定する走行制御，ジャイロセンサから取得した車体の角速度と DC モータから取得した回転角度，走行で設定される前後進/旋回速度から車体が安定するようサーボ制御を行い，DC モータの出力となる PWM の値を決定する倒立制御から構成されている．

二輪型倒立振子で使用されているセンサ・アクチュエータの特性を表 1 に示す．

表 1 二輪型倒立振子ロボットで使用するセンサ・アクチュエータの特性
Table 1 Characteristics of Sensors and Actuators Used in Inverted Pendulum Robot

センサ	出力値	単位	データタイプ	最大サンプル数 [1/sec]
ロータリエンコーダ	回転角度	deg	int32	1000
ジャイロセンサ	傾斜角速度	deg/sec	uint16	300
DC モータ	PWM	%	int8	500

以上の仕様をもつ二輪型倒立振子ロボットをモデリングした．モータの PWM 値に関する制約にステレオタイプ`<<assert>>`を付加し，ブロック定義図において定数となるバリューにステレオタイプ`<<constant>>`を付加した．ステレオタイプを付加したブロック定義図を図 1 に示す．また，内部ブロック図を図 2 に示す．

倒立制御のために実行されるサーボ制御の制約を定義したブロック定義図を図 3 に，パラメトリック図を図 4 に示す．MathWorks 社から提供される NXTway-GS という二輪型倒立振子ロボットに関する関連資料[11] を参考に本事例ではサーボ制御を 6 つの制約ブロックに分割しサーボ制御制約を定義した．

© 2012 Information Processing Society of Japan

Vol.2012-SE-175 No.19
2012/3/16

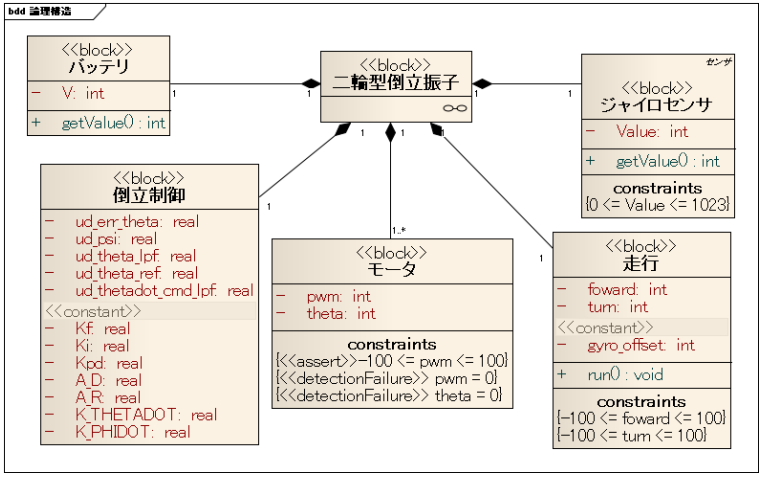

図 1 二輪型倒立振子ロボットのブロック定義図
Fig. 1 Block Define Diagram of Inverted Pendulum Robot

- 状態値制約ブロック (状態値に関する制約を記述)
- 制御目標値制約ブロック
- 目標値制約ブロック
- 入力値 (PWM 制約) 制約ブロック
- 車輪の平均角度
- ローパスフィルタ

それぞれの制約ブロック図において定数となるパラメータにステレオ タイプ<<constant>>を付加した．

Yices から Simulink へ渡すデータは図 2 より外部と通信しているフローポートのうち direction プロパティが out 方向のフローポート l_moter.pwm と r_moter.pwm である．この 2 つのデータが Yices から Simulink へ渡すデータとなる．Simulink から Yices に渡すデータは図 2 で外部と通信している direction プロパティが in 方向の l_moter.theta, r_moter.theta，gyro.value，battery.V である．

SysML モデルから変換した Yices の記述の一部を図 5 に示し，図 6 に使用する Simulink モデルを示す．

Yices からの出力結果は l_moter.pwm と r_moter.pwm という Constant ブロックに入力

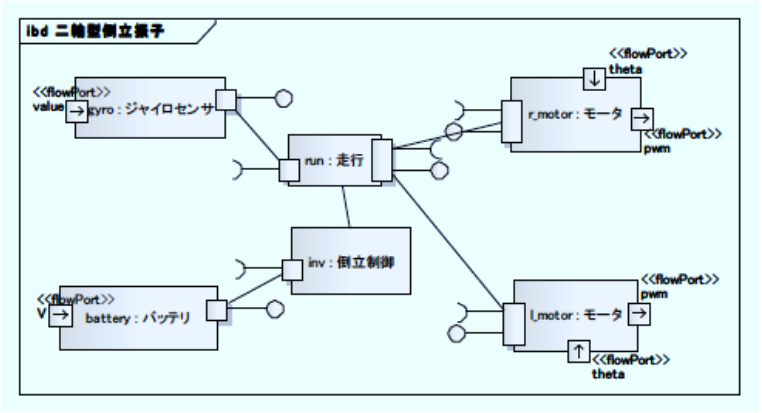

図 2 二輪型倒立振子ロボットの内部ブロック図
Fig. 2 Internal Block Diagram of Inverted Pendulum Robot

する．実行時の出力結果は To Workspace を用いてワークスペースに記録する．また，シミュレーション中に二輪型倒立振子が転倒したことを検知するために Viewer ブロックが接続されている．

4.2 二輪型倒立振子ロボットを用いた適用実験

実験を行うにあたり，ジャイロセンサのオフセットを 634 とした．初期値における Yices での検証を行ったが反例は出力されなかったため，初期条件で充足しない状況が存在しないことをあらかじめ確認した．

静止 (走行しない) 状態の二輪型倒立振子ロボットに対して旋回速度と前進・後進速度については SysML モデルで記述した通り-100 から 100 の間で制御し，左右のモータ出力の分解能を変化させた場合と，片方のモータが故障し動作しない場合の実験を行う．

実験準備：静止状態における限界値分析

実験を行う準備段階として，二輪型倒立振子ロボットが静止状態を保てる傾きの限界値を求める．モータの PWM が-100 から 100 の値の間で左右いずれかのモータの回転角度を 0 に維持できるジャイロセンサの値を Yices により求める．Yices モデルの l_moter.theta と r_moter.theta の値を 0 にして検証を行う．検証結果から静止時にモータの回転角度を 0 に維持できるジャイロセンサの限界値は 469 から 800 の範囲になることが分かった．

また，静止状態での傾きの限界値での制御可能なモータの回転角度を求めると表 2 のよ

© 2012 Information Processing Society of Japan

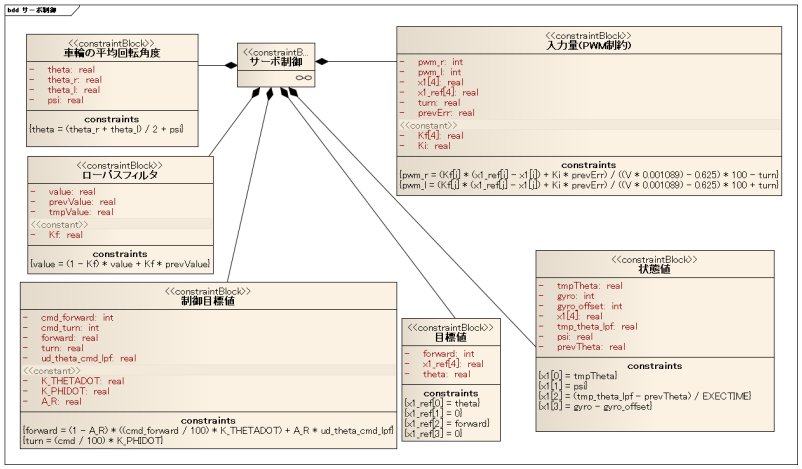

図 3 ステレオタイプを付加した制約のブロック定義図
Fig. 3 Constraint Block Define Diagram with Stereo Type

表 2 制御可能限界でのジャイロセンサ値とモータ PWM
Table 2 Motor PWM and Gyro Sensor in the Controllable Limit Value

ジャイロセンサ値	最小回転角 (モータ PWM)
469	0(-100)
800	-15(-92)
ジャイロセンサ値	最大回転角 (モータ PWM)
469	15(92)
800	0(100)

表 3 ジャイロ初期状態 (634) での制御可能モータ回転角度とモータ PWM
Table 3 Controllable Motor Rotation Angle and PWM in Initial State of Gyro Sensor

最小回転角	最小回転角におけるモータ PWM
-7	-90
最大回転角	最大回転角におけるモータ PWM
7	90

うになった．さらに，初期状態のジャイロセンサ値 (634) における制御可能なモータの回転角度は表 3 のようになることが分かった．

表 2 および表 3 より，初期状態のロボットは左右モータ PWM を-90 から 90 の範囲内で制御しなければならず，倒立状態を保持すべきモータの回転角は-15 から 15 の範囲になることが分かる．

実験 1：左右のモータ出力の分解能を変化させた場合

モータの PWM 値は-100 から 100 の 201 段階に設定可能であり，これを 8bit の整数型で表現する．本実験ではモータ出力の分解能を下げて 8bit の整数型を 7bit 以下にした場合

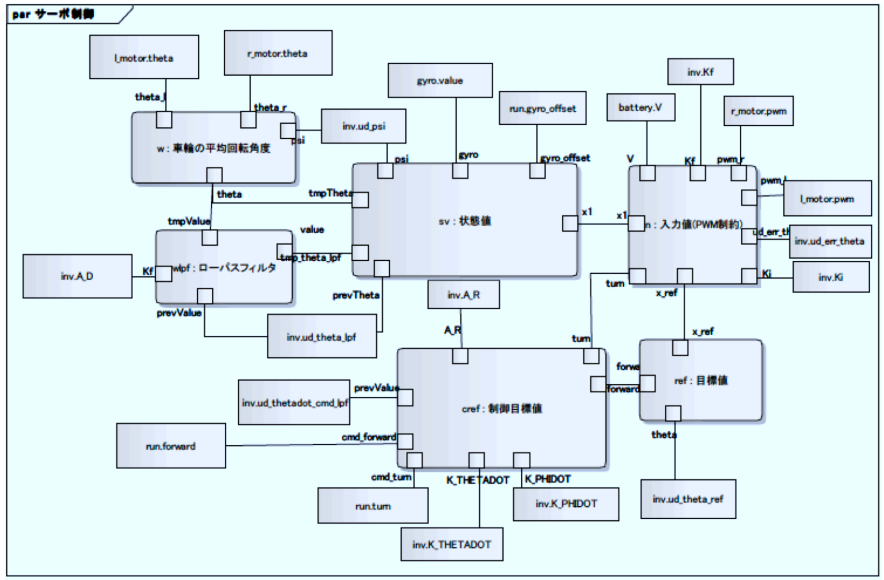

図 4 サーボ制御のパラメトリック図
Fig. 4 Parametric Diagram of Servo Control

の影響を分析する．

まず，101 段階に設定した場合で実験を行った．あらかじめ通常の Simulink モデルを利用して PWM 値を偶数値のみに限定した場合のシミュレーションを行った．このときの結果をワークスペースに保存されている値で確認したが制約を満たさなくなるようなパラメータはなく，また，シミュレーション結果も転倒するようなことはなかった．

次に，Yices を用いて検証を行った．まず，図 1 のモータブロックに以下の制約を追加する．

$$\texttt{<<assert>>}\ \mathrm{pwm}\bmod 2 = 0 \tag{3}$$

SysML モデルに追加した制約を Yices に記述すると次のようになる．

$$(\mathrm{assert}\ (=\ 0\ (\mathrm{mod}\ \mathrm{r\_moter.pwm}\ 2))) \tag{4}$$

$$(\mathrm{assert}\ (=\ 0\ (\mathrm{mod}\ \mathrm{l\_moter.pwm}\ 2))) \tag{5}$$

これを図 5 に追加した Yices モデルを実行し反例の有無を調べる．反例がない場合は充足例を出力させ Simulink に入力しシミュレーションを実行する．実行で得られた l_moter.theta,

© 2012 Information Processing Society of Japan

Vol.2012-SE-175 No.19
2012/3/16

```
;; プロパティからの抽出
;; 走行
(define *run.forward::int)
(define *run.turn::int)
(define *run.*gyro_offset::int 600)
(assert (and (>= *run.forward -100) (<= *run.forward 100)))
(assert (and (>= *run.turn -100) (<= *run.turn 100)))
;; ジャイロ
(define *gyro.value::int)
(assert (and (>= *gyro.value 0) (<= *gyro.value 1024)))
…
;; 制約ブロックからの抽出
;; 車輪の回転角度
(define w.theta::real)
(define w.theta_r::int)
(define w.theta_l::int)
(define w.psi::real)
(assert (= w.theta (+ (/ (+ (* DEG2RAG w.theta_l) (* DEG2RAG w.theta_r)) 2) w.psi)))
;; 制御目標地
(define cref.cmd_forward::int)
(define cref.cmd_turn::int)
(define cref.forward::real)
(define cref.turn::real)
(define cref.K_THETADOT::real *inv.K_THETADOT)
(define cref.K_PHIDOT::real *inv.K_PHIDOT)
(define cref.A_R::real *inv.A_R)
(define cref.prevTheta::real)
(assert (= cref.forward
         (+ (* (- 1 cref.A_R) (* (/ cref.cmd_forward 100) cref.K_THETADOT)) (* cref.A_R cref.prevTheta))))
(assert (= cref.turn (* (/ cref.cmd_turn 100) cref.K_PHIDOT)))
…
;; バインディングコネクタ
;; 回転角度
(assert (= *l_moter.theta w.theta_l))
(assert (= *r_moter.theta w.theta_r))
(assert (= *inv.ud_psi w.psi))
;; 制御目標
(assert (= *run.forward cref.cmd_forward))
(assert (= *run.turn cref.cmd_turn))
(assert (= *inv.ud_thetadot_cmd_lpf cref.prevTheta))
;; 制御目標-目標
(assert (= cref.forward ref.forward))
;; 制御目標-pwm
(assert (= cref.turn in.turn))
;; 目標-pwm
(assert (= ref.x1_ref[0] in.x1_ref[0]))
(assert (= ref.x1_ref[1] in.x1_ref[1]))
(assert (= ref.x1_ref[2] in.x1_ref[2]))
(assert (= ref.x1_ref[3] in.x1_ref[3]))
…
;; 評価式の否定
(assert (not (and (>= *l_moter.pwm -100) (<= *l_moter.pwm 100) (>= *r_moter.pwm -100) (<= *r_moter.pwm 100))))
```

図 5　変換した Yices 記述
Fig. 5　Converted Yices Description

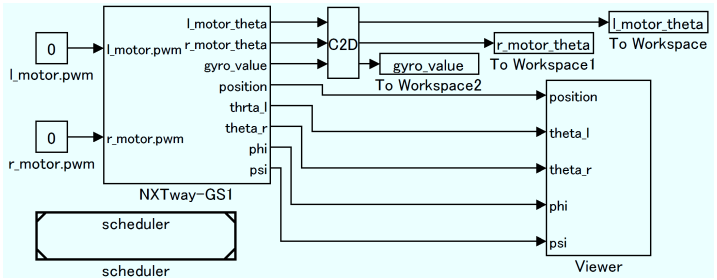

図 6　二輪型倒立振子の Simulink モデル
Fig. 6　Simulink Model of Inverted Pendulum Robot

```
(= run.forward -50)
(= run.turn 95)
(= gyro.value 634)
…
(= sv.gyro_offset 634)
…
(= l_moter.theta -75)
(= r_moter.theta -103)
(= l_moter.pwm -1116)
(= r_moter.pwm -1164)
```

図 7　実験 1 における反例 (一部抜粋)
Fig. 7　Counterexample in the Experiment 1

```
(= run.forward 71)
(= run.turn -3)
(= gyro.value 653)
…
(= sv.gyro_offset 634)
…
(= l_moter.theta 0)
(= r_moter.theta 0)
(= l_moter.pwm 10)
(= r_moter.pwm 12)
```

図 8　左右モータ PWM が異なる充足例 (一部抜粋)
Fig. 8　Satisfy Example with Different Left and Right Motor PWM

r_moter.theta，gyro.value を Yices に入力し反例の有無を調べる．

得られた結果をモータ出力が通常の分解能の場合と比較したが，どちらも実験中に反例は出力されなかった．実験で出力されたジャイロセンサ値の最大値と最小値であるが，どちらも最大値は 669，最小値は 603 であった．また，左右モータの PWM 値に関しても，最大値はどちらも 58 で，最小値は通常状態で-69，本実験結果では-68 とほとんど違いがなかった．追実験としてモータの分解能をさらに粗くした場合反例が出力されるようになった．6bit 以下の分解能では図 7 に示すような反例が出力され初期状態でも制約を充足できなかった．

図 7 は 6bit 分解能での初期状態における反例である．左右モータの PWM 値が限界値を大きく超えていることが確認できる．

実験２：左モータを故障させた場合

本実験では左側のモータを故障させた場合の影響を分析する．静止状態からスタートさ

© 2012 Information Processing Society of Japan

せ，スタート時から左モータが動作しない場合を考える．

まず，図 1 のモータブロックに以下の制約を追加する．

<<detectionfailure>> pwm = 0 (6)

これは，モータは故障時は PWM 値が 0 になることを示す．theta に関しては故障時もタイヤは回転することを考慮し制約は追加しない．SysML モデルに追加した制約を Yices に記述すると次のようになる．

(assert (= l_moter.pwm 0)) (7)

この記述を図 5 に追加した Yices モデルを実行する．得られた充足例・非充足例のパラメータの組み合わせを求めるために，それぞれの場合について，既に得られた例以外のパラメータの組み合わせを Yices を繰り返し実行することで得る．繰り返し実行した結果，左モータの PWM 値が 0 の場合でも右モータの PWM 値が-50 から 50 の範囲であれば制約を満たすことが分かった．

追実験としてタイヤも回転しない場合の影響も分析した．SysML と Yices には次の記述を追加した．

<<detectionfailure>> theta = 0 (8)

(assert (= l_moter.theta 0)) (9)

この場合でも右モータの PWM 値が-50 から 50 の範囲であれば制約を満たすことが分かった．

実 験 考 察

実験 1 に関して，モータの PWM 出力値の分解能を 101 段階までなら粗くすることは可能であると確認できた．ただし，今回の実験では旋回値に制約をつけていないため，図 8 に示すように制約を満たしていても左右モータの PWM を等しくすることができない場合がある．よって，通常の分解能であれば前後運動だけで倒立制御が可能であるが，分解能を粗くしてしまうと車体を旋回させなければ倒立状態を維持することはできないことがわかる．

実験 2 に関して，故障時の PWM 値が-50 から 50 の範囲になる原因を分析したところ，旋回値が-100 から 100 までという制約が原因であることが分かった．静止・走行中に限らず片方のモータ PWM 値が-51 以下または 51 以上のときに，もう片方のモータが故障すると旋回限界値を超えてしまい転倒する可能性がある．この結果から，片方のモータが故障した場合でももう片方のモータの PWM 値を-50 から 50 の範囲に制御すれば転倒対策が可能であるということが分かった．

5. 考　　察

本章では，Yices の適用可能性について考察する．

5.1 ハードウェア性能決定のための Yices の適用

ハードウェア性能の変更自体は Simulink モデルで容易に実現できる．しかし，Simulink によるシミュレーションだけでは網羅性に欠け影響を分析することは難しい．そこで，Yices と Simulink を連携させ各ハードウェア性能ごとに制約の充足可能性を検証することで，網羅的に影響の出るパラメータを求めることができる．

これにより，センサやアクチュエータの精度を変更した場合のシステムへの影響の有無を調べることができる．また，分析結果からソフトウェア側で対策が可能であれば，想定していたよりもハードウェアの性能を下げることが可能である．

分析によってハードウェア性能を変更する場合，その変更を SysML モデル内に反映するならば，ハードウェア性能が検証済みであることを示すためのステレオタイプが必要であると考える．

5.2 ハードウェア故障影響分析のための Yices の適用

前述したハードウェア性能決定のための Yices の利用方法を応用することで，ハードウェアが想定していた性能を発揮できない場合の影響分析を行うことができる．本研究ではその実験として，ハードウェアが故障した場合の影響分析を行った．

危険側故障に対してその影響を分析し故障対策を行うことで危険側故障を少なくできれば安全度水準を引き上げることができる．

故障による制約充足可能性への影響を分析することで，影響の出るパラメータの組み合わせを網羅できるだけでなく，影響の出ないパラメータとの境界値を求めることができるため，危険側故障を安全側故障に落とし込むための対策を支援できる．

6. 関 連 研 究

各産業の安全基準で複雑化するシステム中のソフトウェアの安全性の確認のために形式手法の使用が強く勧められている．D Evrot らの研究[12] では，システムの振舞いが要求を満たすか確認するために SysML 要求図とブロック定義図および形式手法を組み合わせた方法を提案している．SysML 要求図を使用した複雑な制御システムの安全要求を確認するために UPPAAL モデル検査ツールを使用している．

© 2012 Information Processing Society of Japan

Yosr Jarraya らの研究[13] では時間と確率的側面に関して SysML で表現された設計モデルに対する検証手法として，SysML アクティビティ図として記述されたシステム設計の性能解析を提案している．解析には PRISM モデル検査ツールを使用している．

これらの研究のように SysML のシーケンス図やステートマシン図，アクティビティ図を用いた振舞い仕様の検証/シミュレーションに関する研究は数多くされている．しかし，SysML で新規追加されたパラメトリック図などの制約の充足可能性に関する研究はされていない．さらに，本研究では SMT ソルバを用いてパラメトリック図をはじめ構造図にある静的な論理，数式制約に関する充足可能性検証を，Simulink によるシミュレーションと連携させることによりフィードバックループを考慮しながら行っている．また，検証だけでなくハードウェアの変更や故障といった要因に対する性能解析と設計段階におけるパラメータ設定を支援できる．

7. ま と め

本研究では SysML モデルの制約充足可能性検証手法を拡張したハードウェアのシステムへの影響分析手法を検討し，ハードウェア性能及び故障の影響分析とその対策支援手法の提案した．手法適用実験として二輪型倒立振子ロボットモデルを用いて考察を行った．本手法を用いることにより，ハードウェア性能の選定のために必要な影響分析を支援できるようになり，開発の効率化とハードウェア構成の最適化に貢献できると考える．

今後の課題として手法の自動化が必要である．本手法では離散系モデルと連続系モデル間におけるデータ交換の必要であるが，受け渡すデータ量が増えると手動で行うことが困難である．そのため提案手法のサポートツールを現在開発中である．また，本論文では手法の適用例として二輪型倒立振子ロボットを取り上げたが，組込みシステム開発は様々な分野があり開発方法も様々である．したがって，本手法の妥当性や汎用性を評価するには適用事例を増やす必要がある．

参 考 文 献

1) The MathWorks:MATLAB
http://www.mathworks.com/products/matlab/ .
2) The MathWorks:Simulink,
http://www.mathworks.co.jp/products/simulink/ .
3) 加藤 秀明, 上田 賀一, 中島 震:SysML モデルの制約妥当性検証に関する考察, 情報処理学会, 組込みシステムシンポジウム 2010 論文集 (ESS2010), Vol.2010 (2010)
4) Object Management Group, OMG Systems Modeling Language (SysML),
http://www.omgsysml.org/
5) S Ranise, C Tinelli:Satisfiability modulo theories,Trends and Controversies-IEEE Magazine on Intelligent Systems,Vol.21,No.6,pp.71-81(2006)
6) 岩沼宏治, 鍋島英知: SMT:個別理論を取り扱う SAT 技術 (<特集>最近の SAT 技術の発展), 人工知能学会誌,Vol.25,No.1, pp.86-95 (2010)
7) Prasad, M. R., Biere, A., and Gupta, A.: A survey of recent advances in SAT-based formal verication, STTT, Vol. 7, No. 2, pp. 156-173 (2005)
8) 中島震:形式手法の潮流—アーキテクチャへの関心—(『システム設計のための形式手法の基礎と応用』特集号), システム/制御/情報, Vol.9, No.52 , pp.1-6 (2008)
9) 土屋達弘, 菊野亨: モデル検査入門 (ミニ特集安心・安全システム構築のための形式検証技術), 計測と制御,Vol.48, No.11, pp.797-802 (2009)
10) Yices: An SMT Solver,
http://yices.csl.sri.com/
11) NXTwayGS のモデルベース開発 LEGO Mindstorm NXT を用いた二輪型倒立振子ロボットの制御
http://www.mathworks.com/tagteam/56683_TA060_Model_Based_Development_of_NXTway_GS.pdf
12) D Evrot, JF Pétin, G Morel: Combining SysML and formal models for safety requirements verification, 22nd ICSSEA Conference (2010)
13) Yosr Jarraya, Andrei Soeanu, Mourad Debbabi, Fawzi Hassaine : Automatic Verification and Performance Analysis of Time-Constrained SysML Activity Diagrams, 14th Annual IEEE International Conference and Workshops on the Engineering of Computer-Based Systems (ECBS'07), pp.515-522 (2007)

© 2012 Information Processing Society of Japan