ITS 画像処理系・制御系開発における ハードウェア・ソフトウェア協調設計方式

遠藤 祐[†] 吉田 健[†] 井上 聡[†]
飯田庸介[†] 小泉寿男[†]

本論文は，ITS（Intelligent Transport Systems）における画像処理開発・運転制御開発のハードウェア・ソフトウェア協調設計方式を提案する．本方式は，ハードウェアとソフトウェア間のトレードオフによって両者の機能分担を最適にすることを目的としている．本方式では，まず，目標システム全体をソフトウェアでモデリングし，機能やバランスの調整を行う．次にモデルに含まれている各処理のハードウェア・ソフトウェアそれぞれの性能をシミュレーションにより評価しハードウェア・ソフトウェアに的確にトレードオフする．機能分担した各処理をハードウェア・ソフトウェアそれぞれの開発ツールで設計・チューニングしなおし，コデザイン評価セットやラジコンカーを用いて，実時間での動作確認を行い，目標を満足するシステムを迅速に構築する設計を目指す．本方式の適用例として，ITS 画像処理開発・運転制御開発分野を取り上げ，安全運転の支援システムを構築し評価を行い，その有効性を確認した．

A Hardware/Software Co-design Method in Development of ITS Information Processing and Control System

Yu Endo,[†] Takeshi Yoshida,[†] Satoshi Inoue,[†] Yosuke Iida[†] and Hisao Koizumi[†]

The authors propose a method for achieving the right balance between hardware and software when developing image processing and drive control in intelligent transport systems (ITS). The aim of this method is to optimize the division of functions between hardware and software in view of the tradeoffs between the two. The proposed method starts by modeling the target system as a whole in software, adjusting the functions and the balance between them. Next, the performance of each processing block in the model is evaluated by simulating both hardware and software implementations to determine the tradeoffs. Using hardware and software tools, the design of each of the processing blocks after functional division is revised and fine-tuned as necessary, and the operations in real time are checked by means of co-design evaluation sets and radio- controlled cars, with the aim of designing a system that can be built quickly satisfying the objectives. Taking a sample application from the field of ITS image-processing development and drive-control development, a safe driving assistance system was designed and evaluated, demonstrating the validity of this approach.

1. はじめに

組み込みシステム（Embedded System）は，情報通信，情報家電，交通制御などの機器に組み込まれ，産業分野で重要な役割を果たしている．マイクロプロセッサと LSI は組み込みシステムの主構成要素であり，マイクロプロセッサ上で動作するソフトウェア部分と LSI 化されるハードウェア部分との機能分担を行うトレードオフ作業が重要視されてきている[1)~3)]．特に，近年，マイクロプロセッサの性能が大幅に向上し，ソフトウェア部分によってカバーされる範囲が増加しつつあり，機能の性能を満足しつつ，ハードウェアとソフトウェアの適切な機能分担を実現する協調設計方式が望まれる．

筆者らは，制御対象モデルと設計対象モデルを結合させたシミュレーションを行い，これをもとにして設計仕様をハードウェアとソフトウェアに機能分担させ，それぞれを段階的に詳細設計していく協調設計方式を提案した[4)]．この方式を ITS（Intelligent Transport System）の画像処理設計に用いて，危険警告システムへの適用可能性を見出した[4)]．しかしながら，この

† 東京電機大学大学院理工学研究科
Graduate School of Science and Engineering, Tokyo Denki University

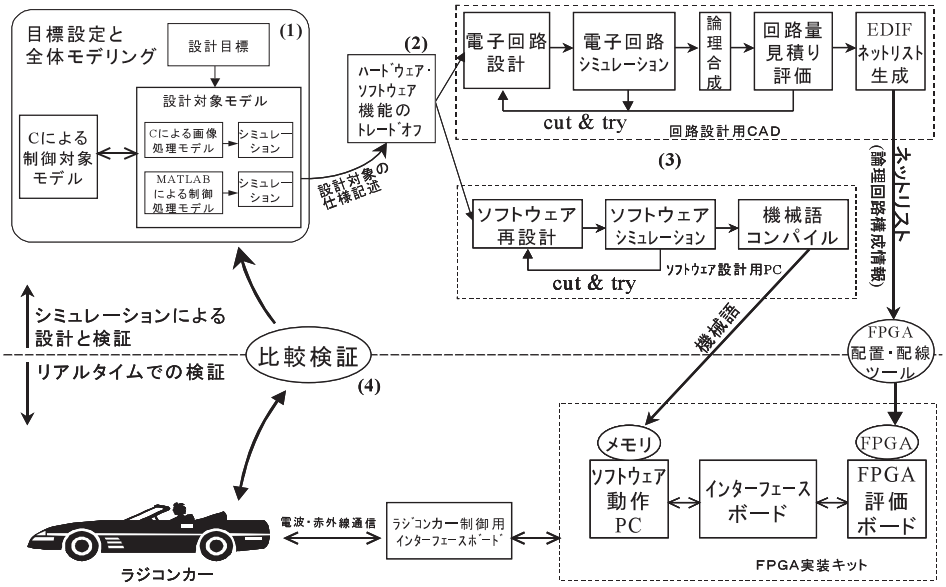

図 1 コデザイン方式のフロー
Fig. 1 The flow of co-design method.

範囲は，画像処理設計と危険警告システムにとどまっており，動作の検証もシミュレーションによる目視の範囲にとどまり，ITS における制御系の電子設計を含む範囲までの協調設計方式とそのリアルタイムでの検証は残された課題となっていた．

ITS 安全運転の支援システムには，自動車専用道路などでの自動運転とそれを実現するまでのマイルストーン上に存在する危険警告や運転補助の機能があり，ITS システムアーキテクチャで危険警告，運転補助，自動運転の 3 つの利用者サービスが定義されている[5),6)]．

運転補助，自動運転においては，車両の制御という動的なモデルを構築しコデザイン方式に組み入れる必要がある[8)]．

本論文では，筆者らが提案するハードウェア・ソフトウェア協調設計を ITS 画像処理および制御系を含めた開発に適用できるようモデルを拡張した方式を提案する．この方式を ITS のシステムアーキテクチャ[5),7),9)]で示されている安全運転支援の危険警告，運転補助，自動運転までへの範囲の適用性について論じる．さらに，本方式の実現性を検証する手段としてラジコンカーを用い，低速の範囲ではあるがリアルタイムの動作確認を行った．

本論文では，2 章に協調設計方式のフローを参考文献 4) に述べた方式を拡張して述べ，3 章で ITS 画像処理，運転制御システムへの適用を論じる．4 章で FPGA 実装キットとラジコンカーによる検証を論じ，5 章で結果の説明と考察を行う．

2. コデザイン方式のフロー

本論文で提案するコデザイン方式のフローを**図 1** に示す．以下，具体的な方法について述べる．

(1) 目標設定と全体モデリング

図 1 の (1) 部分に目標設定と全体モデリングの内容を示す．ここでは，目標システム構築における設計目標および設計環境条件を決める．設計環境条件は，目標システム実現のために利用可能な設備，部品，開発ツールの制限・制約から生じる条件の記述である．これらの設計目標・設計環境条件は後のモデリング段階でのアルゴリズム作成，また，その後のハードウェア・ソフトウェア機能のトレードオフのための判断要素として用いられる．次に，目標システムが導入される側の状態を表す制御対象モデルと，システムそのものである設計対象モデルの両者を作成する．制御系を含むシステムの開発のため，提案するコデザイン方式で，運転補助，自動運転の動的モデルを構築する必要がある．その手段として設計対象モデルの制御処理部分は MATLAB 上で動作する動的システムのモデリングや非線形システムのシミュレーションなどを行うブロックダイアグラムシミュレータ SIMULINK を用いて制御処理のブロック線図を構築しシミュレートする．制御処理以外の部分は PC を用いてすべてソフトウェアにてプログラミングし動作確認する．

次に設計対象モデル，制御対象モデルの両者を結合

し，その動作をシミュレーションによって計算し，相方のモデルのチューニングにより設計対象モデルを確定する．この全体モデルを，コデザイン方式を進めていくうえでの機能確認モデルとする．

(2) ハードウェア・ソフトウェア機能のトレードオフ

図1の(2)部分に位置づけされるハードウェア・ソフトウェア機能のトレードオフでは，設計対象モデルの仕様記述をもとにシステム性能（処理速度）を満足する範囲でのコストミニマムを目標とする．ここでは，LSI化する電子回路のゲート数ミニマムをトレードオフの条件とする．このため，処理性能上ソフトウェアで実現可能な部分はソフトウェアで機能分担し，それ以外をハードウェアで機能分担させる．本方式のトレードオフでは，まず大分類として，C言語で仕様記述されている設計対象モデルをいくつかの処理コンポーネントに分割し，各処理コンポーネントに対してソフトウェア速度の見積りを行う．このとき，制御系への拡張のため，SIMULINKでモデリングした制御処理部分は，あらかじめC言語にコンパイルし，設計対象モデルのすべての処理をC言語の仕様記述にする．次にそのC言語で仕様記述されている設計対象モデルをいくつかの処理コンポーネントに分割し，各処理コンポーネントに対してソフトウェア速度の見積りを行う．見積りの結果，設計目標・設計環境条件からみてソフトウェアの処理速度では遅すぎて実現困難と考えられる処理はハードウェアに機能分担し，処理速度をそんなに必要としない処理をソフトウェアに機能分担する．

次に同系列処理コンポーネントのグルーピングを行う．本プロセスでは同じ系列の処理は1つにグルーピングし，トレードオフ対象のコンポーネント数を減らして次に行う詳細分類の各処理に対する全通りのハードウェア，ソフトウェア割当て作業回数を減らす．

詳細分類では，ハードウェア，ソフトウェア全通りの組合せとそのシステム全体の処理時間をシミュレーションによって算出する．このシミュレーションのとき，実際は処理内容によってハードウェア，ソフトウェアの性能差は異なるが，ハードウェア処理時間はソフトウェア処理時間にくらべきわめて小さいため，シミュレーションでのハードウェア処理時間は0 [sec]とする．シミュレーションの結果，設計目標・設計環境条件の処理時間を満たしている組合せをすべてピックアップし，その中でも最小のハードウェア化作業ですむものを選び機能分担する．設計目標・設計環境条件を満たしていなければ，システムのアルゴリズムの改善や設計目標・設計環境条件の見直しを行う．

表1 設計目標
Table 1 Design targets of the ITS safe driving system.

共通	自動車専用の高速道路を対象とする
	専用レーン内を自律走行するシステムとする
	一般車両との混合交通とする
	システムの具体化にあたっては，車載自律型システムを基本とし，インフラの支援は必要最小限とする
	160m先の異物を検出できなくてはならない
	入力画像は256*256画素24bitカラーとする
	画像処理は単眼式ビジョンシステムとする
制御処理（運転補助，自動運転）	高速道路で時速100km走行時160m先の異物を回避できなくてはならない
	レーン移動にて回避するのか停車して回避するのか判断できなくてはならない
	異物が静止体なのか移動体なのか判断できなければならない
	移動体であればその速度と方向がわからねばならない
	異物に衝突しない停車制御目標を立てられねばならない
	障害物手前10mで停車する
	異物に衝突しないレーン移動制御目標を立てられねばならない
	停車制御目標にそった停車制御が出来なくてはならない
	レーン移動制御目標にそったレーン移動制御が出来なくてはならない
	常に道路状態を把握し，制御目標を逐次更新できなくてはならない。
	通常走行時は車速・車間・レーンを維持して走行できなくてはならない

(3) ハードウェア・ソフトウェアの設計

図1の(3)部分では，分担されたハードウェア，ソフトウェアそれぞれに対し，電子回路設計とソフトウェア設計を行う．ハードウェアは，論理回路設計CADを使用して電子回路に置き換え，構成しなおし，シミュレーションや回路量見積りを繰り返しながら詳細設計を行う．次に設計結果を論理合成しネットリストを生成する．ネットリストの結線情報を用い，FPGA（Field Programmable Gate Array）によってハードウェア化する．

ソフトウェア部分はC言語で組まれたプログラムをコンパイラで機械語に変換しPC上で実際に動作させ，その結果がモデル実行時の動作結果と同じかどうかを確認する．さらに処理速度が設計目標・設計環境条件を満たしているかどうかを繰り返し評価，検証しながらソフトウェアのプログラムを作成していく．

(4) 比較検証

車両制御系開発における比較検証をする際，本方式では制御対象モデルにラジコンカーを用いた．図1の(4)部分の比較検証では，FPGA評価ボードとソフトウェア動作PCを，インタフェースボード経由で連動させる．その動作結果をラジコンカー制御用インタフェースボードに入力し，ラジコンカーによる走行制御結果をあらかじめ作成してある制御対象モデルのシミュレーション結果と目視によって比較検証する[12]．その結果，ハードウェア・ソフトウェア機能のトレードオフやアルゴリズムの改善を繰り返しながら最適な設計に近づけていく．

3. ITS画像処理，運転制御システムへの適用

3.1 コデザイン方式の設計目標・設計環境条件

ITS安全運転の支援システムの設計目標を**表1**，設計環境条件を**表2**に示す[10),11)]．表1の共通項目は，危険警告，運転補助，自動運転の3つの利用者サービ

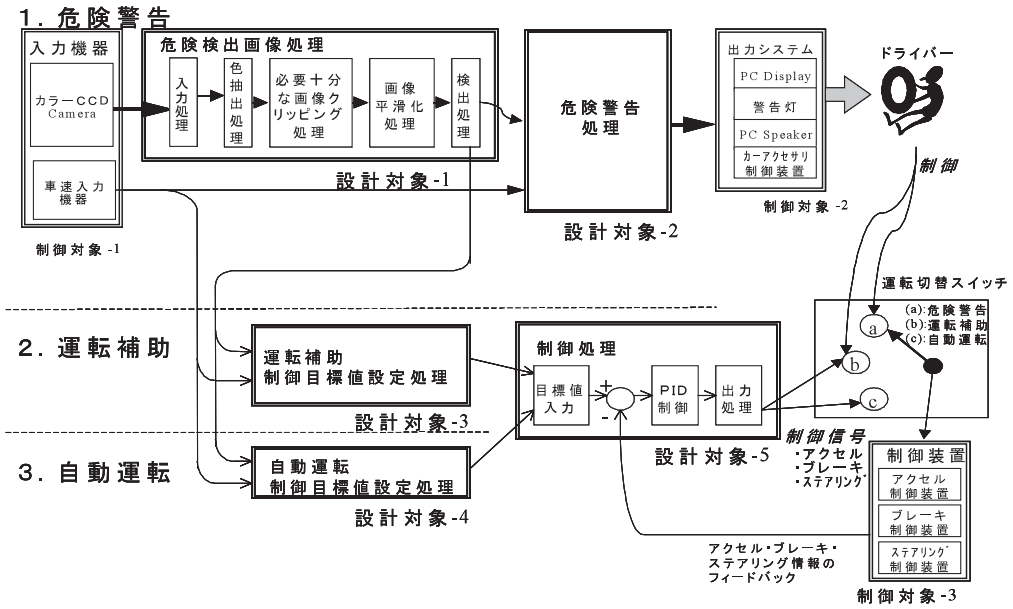

図 2 全体モデル
Fig. 2 The total model of the co-design method.

表 2 設計環境条件
Table 2 Conditions of the ITS co-design.

シミュレーションによる検証	制御処理はMATLABにてモデリングしシミュレートする。
	前方画像はハードディスクに記録されている動画情報を用いる。
	画像処理，警告処理部分はハードウェア・ソフトウェア両者ともC言語にて作成しシミュレートする。
FPGA実装キットによる検証	前方動画像はビデオデッキより再生したものを動画像入力インターフェースボードに入力する。
	実証は，動作結果を制御対象PC上に出力し自動車の制御状態や警告状態の確認を行う。
	ハードウェア動作には，50kゲートのFPGAを使用する。
共通項目	ボイスアラームにはPC用スピーカを使用する。
	ITS Displayには PC用モニターを使用する。
	自車ステータスはPC用シリアルポートを介して入力する。
	ソフトウェアの設計・動作には汎用PCを使用する。
	ハードウェア・ソフトウェア間の通信速度は4MBPSである。
	外部制御入出力はシリアルポートを介し9600BPSである。
	自車速度は実験者が逐次シミュレーション時に入力する。
	実証は，入力画像に対する制御信号・警告信号の出力結果のログをとり行う。

スで共通の目標や条件である．制御処理項目は運転補助，自動運転での目標や条件である．表 2 の設計環境条件は各検証ごとに異なる利用可能設備や部品，開発ツールなどの条件である．この設計目標より，危険警告における条件処理時間は 0.68 [sec] 未満となる．また，制御処理は，車両が 100 [km/h] 走行時，20 [cm] 進むごとに 1 回処理を行うようにする．よって，制御処理の条件処理時間は 7.2 [ms] 以内となる．

3.2 全体モデリング

ITS 安全運転の支援システムの全体モデルを**図 2** に示す．入力機器（制御対象-1）として，自動車上部に設置したカラー CCD カメラ（車上カメラ）から前方道路動画像を危険検出画像処理（設計対象-1）の入力処理が受け取り，その動画像を用いて色抽出処理がアスファルト，車線，遠・中・近距離異物，道路線形を抽出する．その後，必要十分な画像クリッピング処理や画像平滑化処理を経て，検出処理が前方の道路構造や車両，障害物，レーン，分合流部などを検出する．検出された情報と車速入力機器の速度情報は，危険警告（設計対象-2），運転補助（設計対象-3），自動運転（設計対象-4）それぞれの処理に入力される．

危険警告処理が機能中に危険が検出されれば，出力システム（制御対象-2）を用いてドライバーに警告する．また，危険警告では，車の運転はドライバーが行い，運転補助が機能中は，車はドライバーと運転補助処理の両者で制御される．通常走行時はドライバーが車を運転するが，危険検出画像処理が危険を検出すれば運転補助処理が制御処理（設計対象-5）に制御目標を送り制御装置（制御対象-3）を動かし危険回避制御を行う．自動運転が機能中は，車の運転をすべて自動運転処理が行う．危険検出画像処理により危険が検出されたら運転補助と同様に危険回避制御を行う．平常時はレーン維持や車間維持，速度維持などの自動運転制御を行う．

3.2.1 危険警告[4)]

「危険警告」については，ドライバーに前方異物による危険を警告するものである[4)]．すなわち危険警告が機能中の場合，危険検出画像処理の検出結果は危険警告の各警告処理に入力される．これら警告処理では，道路構造や前方向の車両，歩行者・障害物，車線変更，

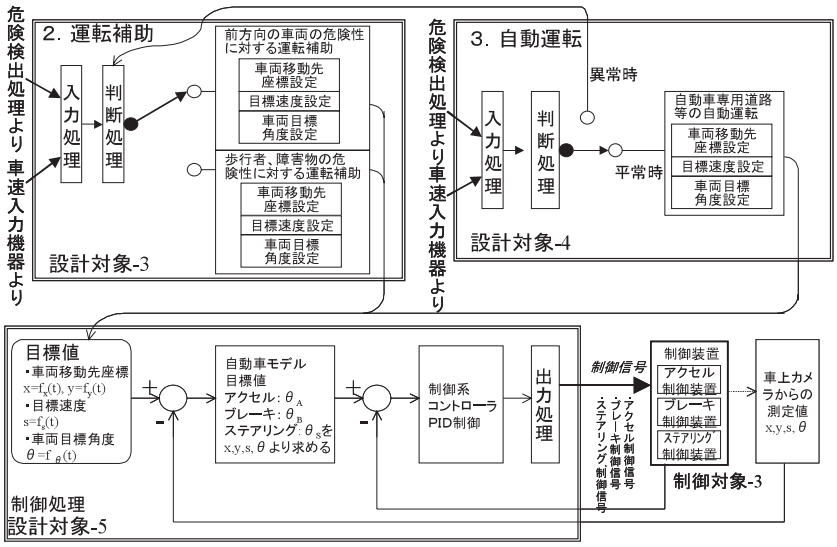

図 3 運転補助・自動運転モデル
Fig. 3 The model of driving support & automatic driving.

車線逸脱，分合流部などについて危険があるかを危険検出画像処理の各検出結果より判断し，危険があればその危険度から弱・中・強の 3 レベルの警告を使い分け，出力システムからドライバーに警告を発信する．危険警告処理や危険検出画像処理は制御処理ではないので C 言語でモデリングを行う．

3.2.2 運転補助・自動運転

本論文における動的モデルへの拡張部分のモデルである運転補助・自動運転モデルを**図 3** に示す．

（1）運転補助

運転補助が機能中の場合，危険検出画像処理の検出結果は各運転補助処理に入力され，道路構造や，前方向の車両，歩行者・障害物などについて危険が検出されれば車両移動先座標や目標速度，目標車両角度の制御用目標値をリアルタイムに設定し，その目標値よりアクセル，ブレーキ，ステアリングの制御目標値を求める．制御系コントローラは制御装置からのフィードバックをもとに目標値との誤差を修正しながら制御を行う．車両の制御は，PID 制御を用いて行う．アクセル，ブレーキ，ステアリングの制御信号は出力装置をとおして制御装置に送られ自動車の運転補助制御を行う．

また，車載カメラからの前方映像をもとに，自動車の X，Y 座標，速度，角度を求めそれらをフィードバックすることにより制御目標値 $f_x(t)$，$f_y(t)$，$f_\theta(t)$，$f_s(t)$ との誤差を修正し新たなアクセル，ブレーキ，ステアリングの制御目標値をたてる．

（2）自動運転

自動運転が機能中の場合，危険検出画像処理の検出結果は自動運転処理に入力される．

その結果，道路構造や，前方向の車両，歩行者・障害物，車間距離，車線変更，車線逸脱，分合流部などについて危険が検出されれば運転補助同様に車両移動先座標や目標速度，目標車両角度などの制御用目標値をリアルタイムに設定し，その目標値よりアクセル，ブレーキ，ステアリングの制御目標値を求める．制御系コントローラは制御装置からのフィードバックをもとに目標値との誤差を修正しながら制御を行いアクセル，ブレーキ，ステアリングの出力装置をとおして制御装置に制御信号を送り自動車の危険回避制御を行う．

平常時は，レーン維持や車間維持などの制御用目標値をリアルタイムに設定し，ブレーキ制御処理，アクセル制御処理，ステアリング制御処理が出力システムをとおして制御装置に制御信号を送り自動車のレーン維持，車間維持，速度維持制御を行う．

また，車上カメラから自動車の X，Y 座標，速度，角度を求めそれらをフィードバックすることにより制御目標値 $f_x(t)$，$f_y(t)$，$f_\theta(t)$，$f_s(t)$ との誤差を修正し新たなアクセル，ブレーキ，ステアリングの制御目標値をたてる．

3.3 運転補助・自動運転の制御アルゴリズム

運転補助，自動運転のモデリングを行う前に制御アルゴリズムを決める．運転補助は危険検出画像処理が危険を検出したらドライバーに代わり自動車の危険回避制御を行う．ここでは，前方に異物を発見したときの危険回避制御（レーン移動）を例にあげる．トップ

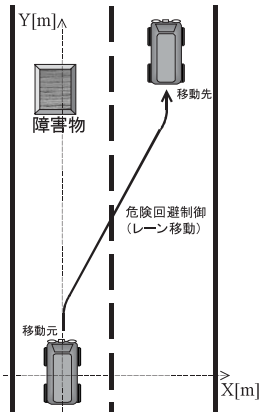

図 4 危険回避制御の様子
Fig. 4 Control to avoid danger.

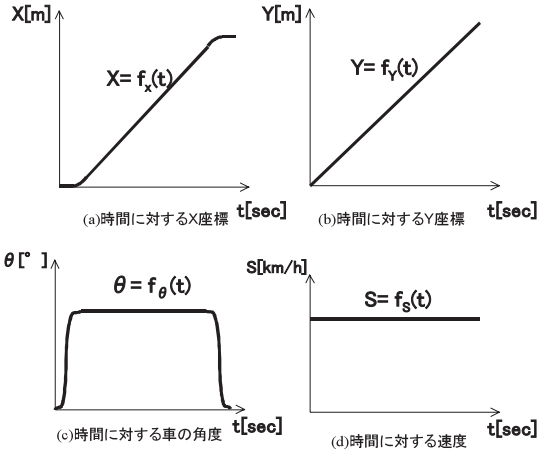

図 5 時間に対する制御目標関数
Fig. 5 Control target functions for time variable.

ビューからの視点で危険回避制御の様子を表したものを**図 4** に示す．図 4 の下方から走行してきた車両が前方に障害物を検出し，衝突を避けるために移動先まで運転補助制御で移動している．

このとき，時間に対する X 座標（水平運動）の推移を**図 5** (a) に示す．図 5 (a) より，レーンの移動し始めたばかりは X 座標の移動量が少ない．これは，急激に横移動して横転やスリップをしないようにするためである．ある程度したら X 座標の移動量は徐々に大きくなりしばらくすると安定して一定の移動量で X 座標を移動している．また移動先に近づくにつれ X 座標の移動量は徐々に少なくなり，X 座標は無理なく移動先ポイントに到着する．

また，Y 座標の時間に対する推移を図 5 (b) に示す．グラフから時間に関係なく移動量は一定で移動先座標まで進んでいる．

図 5 (c) は，時間に対する自動車の角度を示している．このグラフは図 5 (a) の時間に対する X 座標の推移より求まり，移動元から曲がり始めは徐々に車の角度を傾けていき，ある程度で角度の変化は安定している．また，移動先に近づくにつれ車の角度は元の角度まで戻っている．

図 5 (d) は，時間に対する自動車の速度を示している．このグラフは図 5 (b) の時間に対する Y 座標の推移より求まり，Y 座標の移動量が一定であるため，速度も移動元から移動先まで一定である．

これら図 5 (a)，(b)，(c)，(d) の関数，$f_x(t)$，$f_y(t)$，$f_\theta(t)$，$f_s(t)$ を制御目標関数とする．

また，これら $f_x(t)$，$f_y(t)$，$f_\theta(t)$，$f_s(t)$ からアクセル，ブレーキ，ステアリングを制御するための制御目標関数も求める．

自動運転では，平常時はレーン維持，車間維持，速度維持などの制御を行い，危険検出画像処理が危険を検出したら運転補助と同様に危険回避制御を行う．ここでは，自動運転の平常時のレーン維持，車間維持の様子を例にあげる．

トップビューからの視点で平常時の自動運転制御の様子を表したものを**図 6** に示す．

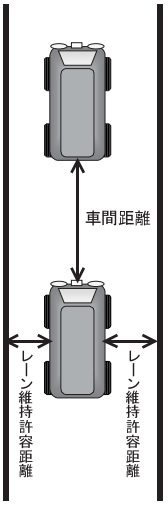

図 6 平常時の自動運転の様子
Fig. 6 Automatic driving in the normal condition.

画像処理でレーンや前方車両を検出して，あらかじめ決められた車間距離，レーン維持許容距離内に自車両が入るように制御目標関数 $f_x(t)$，$f_y(t)$，$f_\theta(t)$，$f_s(t)$ を逐次更新する．それら制御目標関数から，アクセル，ブレーキ，ステアリングを制御するための制御目標関数を求め運転補助制御と同様に自動運転制御を行う．

3.4 シミュレーションによる検証

以上，作成してきたモデルのシミュレーションを行

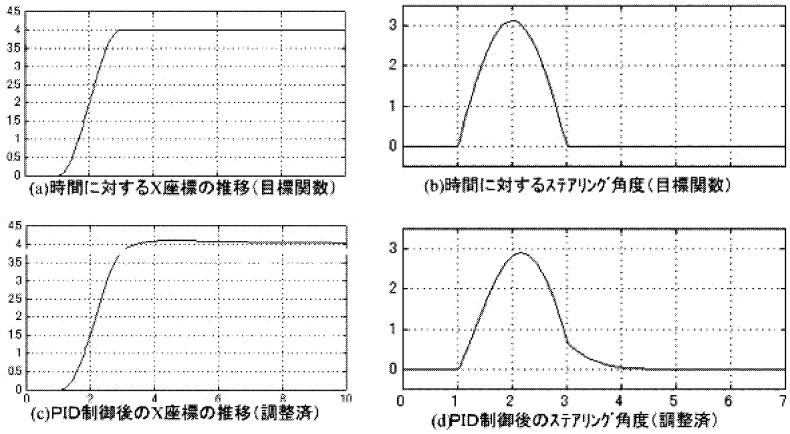

図 7 制御処理部分のシミュレーション結果
Fig. 7 The simulation results of the control processing.

い，機能が正しく動作しているか確認する．

制御処理部分は MATLAB 上の SIMULINK で動作させ，制御処理以外の部分は C 言語をコンパイルし実行する．両者のデータのやりとりはファイルベースで行う．また，入力機器のカラー CCD カメラの代わりに，あらかじめハードディスクに保存しておいた前方道路動画像ファイルを使用し，車速入力機器の代わりに，前方道路動画像ファイルと時定数が同じ車両速度ファイルを使用する．

3.3 節で示した危険回避制御のステアリングの制御部分のシミュレーション結果を**図 7** に示す．図 7 (a), (b) がそれぞれ車両の X 座標とステアリング舵角度の時間に対する推移で制御用目標関数である．ここでは，シミュレーション開始から 1 秒後に 2 秒間かけて 4 m 水平方向に移動することを目標としている．PID 制御用の係数をシミュレーションを繰り返し行い調整したものが，(c)，(d) である．(d) のステアリング角度は目標関数に比べ若干遅れが生じているため (c) の X 座標の推移は目標とした座標を若干オーバーしたが，許容範囲内であった．

MATLAB とソフトウェアで構成されている全体モデルを今後システムを構築していくうえでの目標モデルとして活用する．

3.5 ハードウェア・ソフトウェア機能のトレードオフ

ITS 安全運転の支援システムの設計対象は 78 個の処理コンポーネントからなり，これらの処理コンポーネントをシステムの性能（処理速度）を満たす範囲でのコストミニマムを目標としハードウェア・ソフトウェアに分類する．ハードウェア・ソフトウェア機能のトレードオフは大分類，並列（同系列）処理コンポーネントのグルーピング，詳細分類の 3 つのステップで行う．

大分類では，設計対象モデルの各処理コンポーネントのソフトウェア処理時間を見積もり，システム全体の条件処理時間を上回っていれば，その処理コンポーネントはハードウェアに機能分担する．また，設計目標・設計環境条件でハードウェア・ソフトウェアが決定してしまうものはそのまま機能分担する．

本方式の評価システムでは，ソフトウェア動作用 PC として用いる Windows98 搭載 PC（CPU: Pentium II 233 MHz，RAM: 192 MB）はモデル動作 PC と同じものであるため，モデルの実測でソフトウェア処理時間を得ることができる．その結果，システム全体での条件処理時間 0.68 [sec] を満たしていない処理は，危険検出画像処理の 29.62 [sec] であり，これはハードウェアに機能分担する．

また，制御処理の条件処理時間である 7.2 [ms] を PID 制御処理は満たしているのでソフトウェアに分担する．また，設計目標・設計環境条件より，危険警告処理はソフトウェアに機能分担する．

次に，機能分担されなかった処理のグルーピングを行う．機能分担されなかった残りの処理コンポーネントをそのまま詳細分類するには，トレードオフ対象の数に対して指数関数的なハードウェア・ソフトウェアの組合せを試さなければならない．現段階では，全通りのシステム処理時間をシミュレーションによって算出する詳細分類プロセスへ移行することは実際上困難であるため，同系列の処理は 1 つにグルーピングし，後の詳細分類での作業負担を軽減する．

表 3 ハードウェア・ソフトウェアトレードオフ結果
Table 3 The result of the hardware/software trade-off.

処理内容	機能分担
危険検出画像処理	ハードウェア
危険警告処理	ソフトウェア
運転補助制御目標設定処理	ソフトウェア
制御処理	ソフトウェア
自動運転制御目標値設定処理	ソフトウェア

次に詳細分類を行う．グルーピングされた処理に対してすべてのハードウェア・ソフトウェアの組合せで，設計目標のシステム全体での条件処理時間 0.68 [sec] や制御処理の条件処理時間 7.2 [ms] を上回っていない組合せの中からハードウェア化個数が最小である組合せを詳細分類での結果とする．

ハードウェア・ソフトウェア機能のトレードオフの結果を**表 3** に示す．特に現れた特徴として，画像処理を行う必要がある検知処理がハードウェアの機能担当となり，ユーザに警告を行う処理などのユーザインタフェース部分や制御目標関数を求める処理，制御処理がソフトウェアの機能担当となった．

3.6 ハードウェア・ソフトウェア設計

(1) 設計対象ハードウェア部の設計

設計対象ハードウェア部分の電子回路設計を論理回路設計 CAD を用いて作成した．このツールを用いることにより画像処理の設計においては，設計者が設計結果のシミュレーション画面を見ながら電子回路の詳細設計を Cut&Try 式に作り上げていく．この過程で，ITS 安全運転の支援システムの全体モデルを参照して，各処理コンポーネントの入力に対する出力が判明しているため，これらの入出力データと論理回路設計 CAD を駆使し，現在設計中の処理コンポーネント単位での最適設計を行う．また，論理回路設計 CAD はマッピングレポートにより回路量を知ることができるので，設計環境条件にある 50 k ゲートの回路量を満たしていなければ，シミュレーションと回路設計を繰り返し行う．

(2) 設計対象ソフトウェア部の設計

設計対象のソフトウェア部分は Windows 搭載の PC 上で動作確認する．処理内容は主にハードウェアの危険検出画像処理から受け取った検知信号から状況を判断し，運転者への警告や制御用目標値の設定，車両の制御を行う．警告には警告レベル弱・中・強の 3 レベルがあり，制御対象モデルを組み合わせて警告を機能させる．

制御処理部分は SIMULINK で作成した制御ブロックを MATLAB の C 言語出力機能を用いて C 言語ソースに変換する．さらに，実装するうえで必要なデータ入出力部分の処理を C 言語で記述する．

以上のような機能を織り込んだソフトウェアをコンパイルし機械語にした後，Windows 搭載 PC にダウンロードする．

4. FPGA 実装キットとラジコンカーによる検証

以上述べたコデザイン方式による画像処理系・制御処理系の設計結果をリアルタイムの動作で実証するために，FPGA 実装キット，ソフトウェア動作 PC および，車両制御系開発における制御対象にラジコンカーを用いた検証を行った．検証の構成図を**図 8** に示す．

(1) FPGA 実装キット

図 8 上部 (a) の FPGA 実装キットは，FPGA 評価ボードとソフトウェア動作 PC，そして両者をつなぐハードウェア/ソフトウェア・インタフェースボードからなる．ハードウェアの設計結果である NetList は FPGA にダウンロードし，ソフトウェアの設計結果である C 言語ソースは機械語にコンパイルしたものをソフトウェア動作 PC 上のメモリにロードし実行する．

3.4 節のシミュレーションによる検証と同じ入力動画像で危険警告部分を動作させたところ，モデルのシミュレーションと同等の結果が得られた．

(2) ラジコンカーによる検証

ラジコンカーによる検証の構成を図 8 下部 (b) に示す．ラジコンカーによる検証では，運転補助，自動運転機能の部分を，FPGA 実装キットにラジコンカーの制御機器部分や前方道路画像の撮影，動画像送受信，動画像 AD 変換機器が追加して検証を行った．

ラジコンカーに取り付けた CCD カメラからの前方動画像を，VHF テレビトランスミッタにて電波として送信し，テレビチューナーが受信したアナログカラー映像を動画像入力インタフェースボード経由で 24 ビットカラーデジタル画像に変換し FPGA 実装キットに入力する．FPGA 評価ボードは入力された前方道路画像を処理し異物などを検出する．その検出結果をハードウェア/ソフトウェア・インタフェースボード経由でソフトウェア動作用 PC に送り，ソフトウェア動作用 PC は制御用目標値などを計算し，その結果からプロポ制御用インタフェースボード経由でラジコンカーに制御信号を送信しラジコンカーの制御を行う．

ラジコンカーの構造がフィードバック制御が行えるクローズループな構成になっていないことから，ラジコンカーの制御は**図 9** に示すプリセットパターン方式を用いる．プリセットパターン方式は，制御処理部分のシミュレーション結果から得られた情報をもとに，

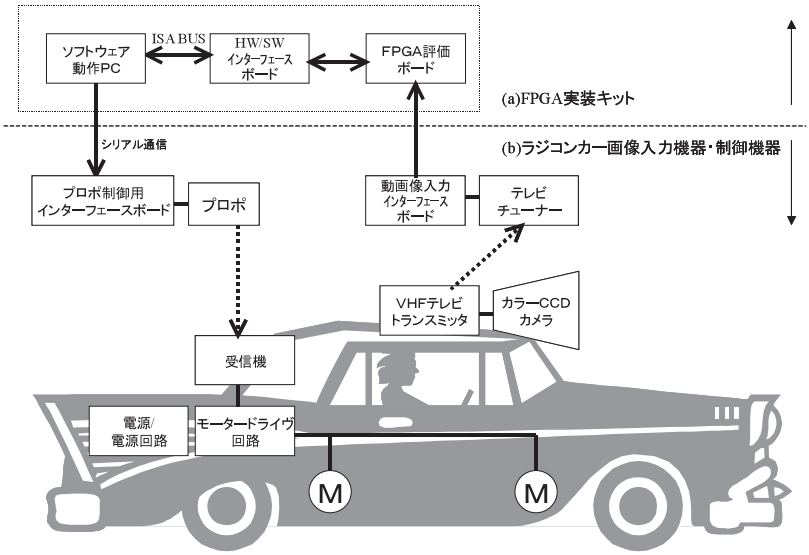

図 8　ラジコンカーによる検証構成図
Fig. 8　Verification by the radio control car.

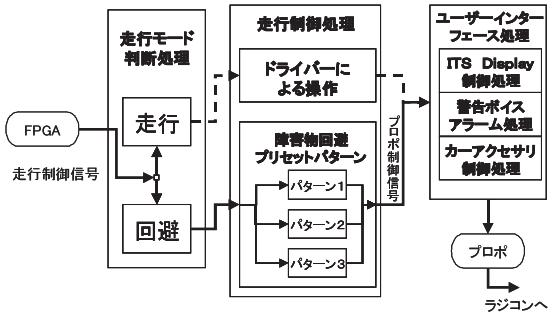

図 9　プリセットパターン制御方式
Fig. 9　Preset pattern control methd.

自車両に対する障害物までの距離や方角，障害物の大きさなどの様々な状況に対応した理想的な回避動作を行ういくつものプリセットパターンをあらかじめ作成しておき，それらを用いて障害物回避の車両制御を行う．また，ラジコンカーの最高速度はモータ性能上の制限で最高時速 10 km だが，本検証では，実車の時速 10 km と同等と見なし，その範囲でリアルタイム検証を行った．

様々な障害物の位置や距離，大きさの組合せでパターンを数種類設定し，それぞれの状況下における，左右回避，緊急停止などの障害物回避パターンが選択され，いずれの場合も回避パターンの目標値に移動することが確認できた．

5. 結果と考察

制御処理部分のモデリングでは，MATLAB 上で動作するブロックダイアグラムシミュレータ SIMULINK で作成したため，シミュレーションを行いながら制御処理を目標とする結果が得られるまで，Cut&Try 式に作り上げることができた．また，トレードオフの際には，SIMULINK の制御ブロック図を C 言語にコンパイルすることにより，制御処理以外の部分と仕様記述を統一することができ，処理性能の比較をスムーズに行い最適なハードウェア・ソフトウェアの機能分担ができた．

この SIMULINK を用いることにより，仮想的なプロトタイプを早く構築しテストすることが可能になり，その結果を直接 C 言語出力することで，より速く，より効果的な設計プロセスを得ることができるようになった．

ハードウェアは，危険検出画像処理の回路をすべて FPGA にマッピングし，約 20 k ゲートであった．電子回路設計時のアルゴリズムの評価修正は，シミュレーション画像処理を見ながら実施できた．ソフトウェアは，主にインタフェース処理と警告処理，制御目標値設定処理，制御処理が機能分担され，そのすべてを C 言語で記述し，約 15,000 ステップであった．

FPGA 実装キットによる検証では，3.4 節のシミュレーションによる検証と同じ入力動画像で動作させたところ，モデルのシミュレーションと同等の結果が得られた．

ラジコンカーによる検証では，様々な障害物の位置や距離，大きさの組合せで検証したところ，制御処理部分のシミュレーション結果とほぼ同じ理想的な回避

パターンで障害物の回避制御が行われていることを確認することができた．これは，モデルを正確に構築することができ，また，そのモデルを自車両に対する障害物までの距離や方角，障害物の大きさなどの様々な回避動作を行うプリセットパターンにスムーズに変換することができたためである．

実行速度については，1画面分の入力に対し，モデルをシミュレートしている段階では，600 [sec] の処理時間に対し，実時間上での実行は，画像処理が 100 [msec] 以内で終了し設計目標を満たした．以上のことにより，本設計手法を用いることで，シミュレーションの段階では，設計目標を満たし，ラジコンカー検証では，時速 10 km の範囲内での設計目標を満たすことを確認することができた．そのことにより，本設計手法が，動的なモデルを含むシステムの設計にも有効であることが分かった．

ターゲットとなるシステムのモデルを最初にコンピュータ上で完成させることにより，制御対象，設計対象の役割や両者間の入出力に必要なデータとそのフォーマットが明確となった．また，目標システムのモデルを確定させていたため，ハードウェアとソフトウェアに各処理を最適割付分担することができ，実時間で動くシステムを開発するうえでのハードウェア，ソフトウェア開発にスムーズに移行できた．そのハードウェア，ソフトウェアの各開発過程では，作成しなければならない処理とその機能が決定しているのでモデル以上のアルゴリズムの最適化や，さらには人間の感性を折りこむような設計も可能であった．

6. ま と め

筆者らは以前，協調設計方式を ITS 画像処理系開発に適用し，構築・評価を行っている[4]．本論文では，ハードウェア・ソフトウェア協調設計方式において，ITS 画像処理系・制御系開発まで含めた協調設計方式を提案し，安全運転の支援システムを構築し評価を行った．その結果，目標とするシステムの制御対象・設計対象モデルをまず作成し，そのシミュレーション結果を参考に機能のトレードオフを行うコデザイン方式を用いることにより，目標とするシステムのハードウェア・ソフトウェアの機能分担を行えるようにした．運転補助，自動運転においては，車両の制御という動的なモデルを構築し，ラジコンカーを用いて低速の範囲ではあるがリアルタイムでの検証を行った．様々な障害物の位置や距離，大きさの組合せで検証したところ，制御処理部分のシミュレーション結果とほぼ同じ理想的な回避パターンで障害物の回避制御が行われていることを確認した．以上のことにより，本設計手法を用いることで，ITS 画像処理系・制御系開発に有効である可能性を見いだした．

ただし，モータ性能やラジコンカーの構造上からくる制限で，ラジコンカーによる検証では最高時速 10 km での検証となった．また，本論文では動的モデルのシミュレーションに PID 制御を用いたが，制御モデルの充実化として，自動車の動的モデルとしてスライディングモード制御を用いるなどの制御モデルを充実させた場合の検証を行う必要がある．

今後の課題は次のとおりである．

(1) ラジコンカーによる検証では，モータ性能やラジコンカーの構造上からくる制限で，最高時速 10 km でフィードバック制御でないリアルタイム検証となったが，今後は，より実車両に近い環境での検証を行う．

(2) 本コデザイン方式のハードウェア/ソフトウェア設計部分に再利用のためのリポジトリデータベースとその選択・組み込みプロセスを追加し，設計者の作業工数をさらに減少できる設計方式を目指す．

(3) 現在のハードウェア・ソフトウェア機能のトレードオフでは処理性能を満たす範囲でのコストミニマムとし，なるべくハードウェアのゲート数を少なく抑えることを目標としているが，今後は，開発工数や量産したときのコストなども考慮にいれてトレードオフ方式を充実させる．

(4) 車両の制御に今回は PID 制御方式を用いたが，今後は各種制御方式を ITS 安全運転の支援システムに導入し，制御方式の違いによるトレードオフ結果やハードウェア・ソフトウェアの設計結果を確認し，コデザイン方式の検証を行う．

(5) 制御処理部分のシミュレーションを MATLAB で行うとき，制御対象である車両モデルが正確であればあるほど，そのシミュレーション結果は正確になる．今後はより車両モデルの正確さを追及する．

(6) 現在は前方画像と車速情報の 2 つの情報を用いている．今後，車車間通信や路車間通信 DSRC（狭域通信）など利用し，様々な情報を取り入れそれらを用いることより，画像処理や制御処理の精度向上を追求する．

(7) 本論文では，回避パターンの検証は目視により行われているが，制御目標関数に対する定量的な回避パターンの検証を行う．

参 考 文 献

1) 中本幸一，高田広章，田丸喜一郎：組込みシステム技術の現状と動向，情報処理，Vol.38, No.10,

pp.871–878 (1997).
2) Koizumi, H., Seo, K., Suzuki, F., Ohtsuru, Y. and Yasuura, H.: A Proposal for a Co-design Method in Control System Using Combination of Models, *IEICE Trans. Inf. Syst.*, Vol.E78-D, No.3, pp.237–247 (1955).
3) Thomas, D.E., Adams, J.K. and Schmit, H.: A Method and Methodology for Hardware-Software Codesign, *IEEE Design and Test of Computers*, Vol.10, No.3, pp.6–15 (1993).
4) 遠藤 祐, 小泉寿男, 清尾克彦：ハードウェア・ソフトウェア協調設計方式と ITS 画像処理開発への適用検証, 電気学会論文誌 D, Vol.120-D, No.10, pp.1118–1126 (2000).
5) 高度道路交通システム（ITS）に関わるシステムアーキテクチャ概論編素案 (1999). http://www.vertis.or.jp/j-SA/
6) 松下 温, 屋代智之：ITS の実現に向けて, 情報処理学会誌, Vol.40, No.10, pp.960–963 (1999).
7) 小泉寿男：ITS と情報処理技術, 情報処理学会誌, Vol.40, No.10, pp.978–981 (1999).
8) AHS 研究組合のホームページ. http://www.ahsra.or.jp/
9) 高木聖和, 久野 晃, 中村哲也：レーザレーダを使用した路面状況検出システム, 電子情報通信学会 1998 年 ITS に関する情報通信シンポジウム, SAD-5-6, pp.97–98 (1998).
10) 内藤貴志, 山田啓一, 山本 新：撮像位置にロバストなナンバープレート認識方式, 電子情報通信学会論文誌 A, Vol.J81-A, No.4, pp.536–545 (1998).
11) 梶原康也, 楠 秀樹, 山田勝規, 溝口正典, 大中慎一, 高野和朗, 黒田浩司：知的状況認識システム開発の方向, 電気学会道路交通研究会資料, RTA-95-28, pp.19–27 (1995).
12) 吉田 健, 飯田庸介, 井上 聡, 小泉寿男：ITS 運転支援システムにおけるハードウェア・ソフトウェア協調設計方式とその検証方式, 情報処理学会誌, Vol.2002, No.21, pp.87–92 (2002).

(平成 14 年 3 月 25 日受付)
(平成 14 年 10 月 7 日採録)

遠藤 祐

昭和 51 年生．平成 11 年 3 月東京電機大学理工学部経営工学科（現情報システム工学科）卒業．平成 13 年 3 月同大学大学院理工学研究科システム工学専攻修士課程修了．同年，株式会社ズーム入社．現在東京電機大学大学院理工学研究科応用システム工学専攻博士後期課程在籍．

吉田 健

昭和 52 年生．平成 13 年東京電機大学理工学部経営工学科（現情報システム工学科）卒業．現在同大学大学院理工学研究科システム工学専攻博士前期課程在籍．

井上 聡

昭和 54 年生．平成 14 年東京電機大学理工学部経営工学科（現情報システム工学科）卒業．現在，同大学大学院理工学研究科情報システム工学専攻博士前期課程在籍．

飯田 庸介

昭和 54 年生．平成 14 年東京電機大学理工学部経営工学科（現情報システム工学科）卒業．現在同大学大学院理工学研究科情報システム工学専攻博士前期課程在籍．

小泉 寿男（正会員）

昭和 36 年東北大学通信工学科卒業．同年三菱電機株式会社入社．昭和 45 年より主として O/S，ソフトウェア生産技術，コンピュータ通信制御に関する研究・開発に従事，平成 10 年 4 月より東京電機大学理工学部教授，博士（情報科学）．電子情報通信学会，電気学会，IEEE 各会員．情報処理学会フェロー．