モバイルコンピューティングと
ワイヤレス通信 15－7
高度交通システム 3－7
（2000．11．30）

ITS画像処理開発・運転制御開発における ハードウェア・ソフトウェア協調設計方式とその検証

遠藤 祐† 潘 耀東‡ 畠山省四郎† 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に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 its verification

Yu Endo,† Yaodong Pan,‡ Shoshiro Hatakeyama,† and Hisao Koizumi†

In this article, a hardware and software co-design method for the image processing system of Intelligent Transport System (ITS) is proposed. The objective of this method is to optimize the division of functions between hardware and software by the optimum trade-off. Under this method, first, the targeted overall system is modeled by software to adjust functions and structures of the system. Then, the performance of each function of software and hardware in the model is evaluated by simulation, and the optimized division of functions between hardware and software is eventually obtained. Each function divided in hardware and software is designed in detail using development tools and then the results of the design of hardware and software are loaded on the co-design test board to verify and evaluate satisfaction in the design objectives. The ITS safe driving system is chosen as an application example for this method and its effectiveness is verified by evaluating the construction of a safe driving support system.

1．はじめに

筆者らは現在までにモデル結合型によるハードウェア，ソフトウェアのコデザイン方式を提案し[1]，ＩＴＳ(Intelligent Transport Systems：高度道路交通システム)における危険警告システムへのコデザイン方式の適用・検証を行った．危険警告におけるＩＴＳ画像処理システムは，前方の障害物までの距離を画像処理にて測定し，その危険度に応じた警告を運転者に発信するシステムである．コデザイン方式を用いてシステムを構築した結果，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うことができた[2) 3) 4)]．

ＩＴＳ安全運転の支援システムには，自動車専用道路等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)]．

本稿では，表１に示す危険警告，運転補助，自動運転の各機能に対する本コデザイン方式の適用検証を論じる．

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．

表1 コデザイン方式検証対象

危険警告	道路構造等の危険警告
	前後方向の車両の危険警告
	歩行者、障害物の危険警告
	車線変更の危険警告
	車線逸脱警告
	分合流部の危険警告
運転補助	道路構造等の危険性に対する運転補助
	前後方向の車両の危険性に対する運転補助
	歩行者、障害物の危険性に対する運転補助
	緊急一斉停止の運転補助
	車線変更時の運転補助
	車線逸脱時の運転補助
	分合流部の運転補助
自動運転	自動車専用道路等の自動運転

† 東京電機大学大学院理工学研究科
‡ Department of Electrical Engineering
The Ohio State University, USA

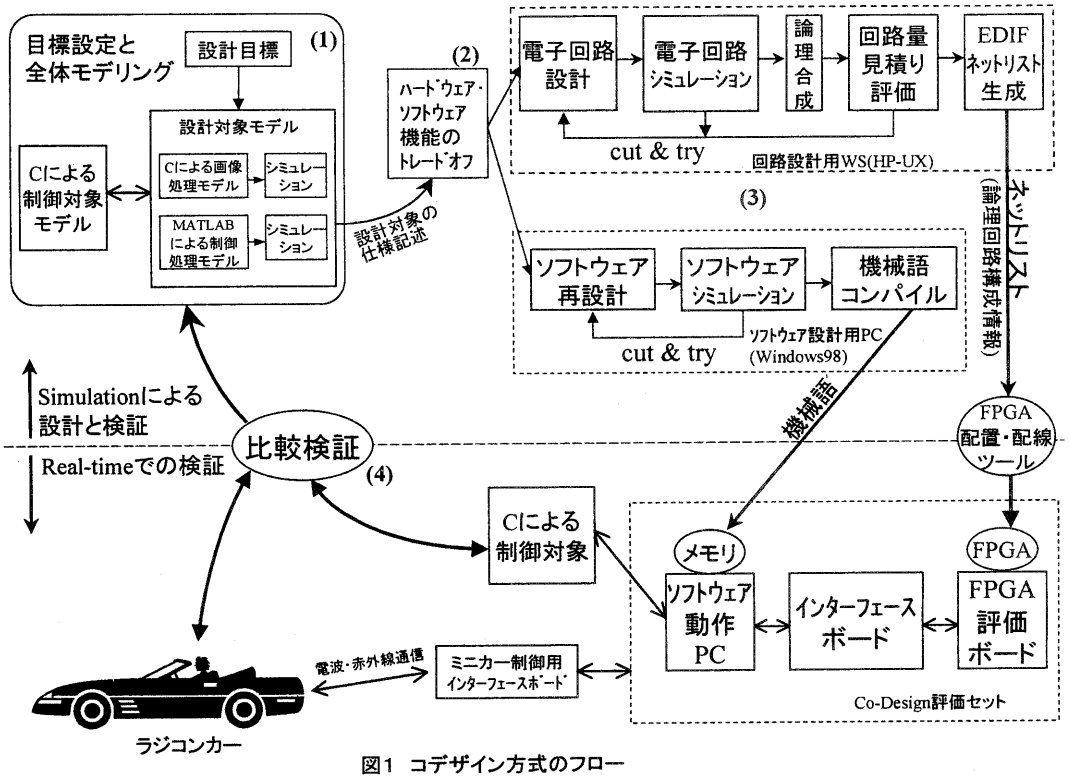

図1 コデザイン方式のフロー

2. コデザイン方式のフロー

本稿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

本方式では，まず，目標システム構築における設計目標及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為に利用可能な設備，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・制約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，設計対象モデルの両者を作成する．ここで，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ＧＵＩ環境で構築しシミュレートする．制御処理以外の部分はＰＣを用いて全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．制御対象モデル，設計対象モデルの両者を合わせ全体モデルとし，コデザイン方式を進めていく上での検証モデルとする．

(2)ハードウェア・ソフトウェア機能のトレードオフ

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．トレードオフでは，まず大分類として，ＳＩＭＵＬＩＮＫで作成した制御処理部分をＣ言語にコンパイルし，設計対象モデルの全ての処理がＣ言語にて仕様記述する．次にそのＣ言語で仕様記述した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．本プロセスではある程度同じ系列の処理は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詳細分類で行う各処理に対する全通りのハードウェア，ソフトウェア割り当て作業回数を減らす．

(3)ハードウェア・ソフトウェアの設計

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，トップダウン論理回路設計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も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・制約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

比較検証では，ＦＰＧＡ評価ボードとソフトウェア動作ＰＣを，インターフェースボード経由にて連動させ，動作結果を制御対象ＰＣに出力する．制御対象ＰＣの出力結果を予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し，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．

次に，ＦＰＧＡ実装キットによる実証で制御対象ＰＣで受け取っていた制御信号や警告信号などを，ラジコンカーによる実証では実際に高速道路のミニチュアコースを走るラジコンカーに入力しスモールスケールでの走行制御を行い検証する．制御結果は，対象とした安全運転の支援のサブサービスが正しく機能しているかどうかを目視にて確認する．

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２，設計環境条件を表３に示す．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．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．

表2 設計目標

設計目標	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停車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

表3 設計環境条件

設計環境条件	実証レベル1，2，3共通	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用ｼﾘｱﾙﾎﾟ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
		ﾊｰﾄﾞｳｪｱ･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ﾎﾟｰﾄを介し9600BPSである
	実証レベル1	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
	実証レベル1，2共通	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
	実証レベル2	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
	実証レベル2，3共通	ハードウェア動作には，50kゲートのFPGAを使用する
		カメラはNTSC方式カラーCCDカメラを使用する
	実証レベル3	ラジコンカー走行による実証ではラジコンカーをハードウェアもしくはソフトウェアからプロポを制御しラジコンカーを走らせる
		ラジコンカー走行による実証ではスモールスケールの高速道路の部分モデルを用いラジコンカーを走行させ実証する
		自車速度は赤外線によりハードウェアもしくはソフトウェアに入力する

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）のカラーＣＣＤ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，運転切替スイッチは(a)に入るので車の運転はドライバーが行う．

運転補助が機能中は，運転切替スイッチは(b)に入るので，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，運転切替スイッチは(c)に入

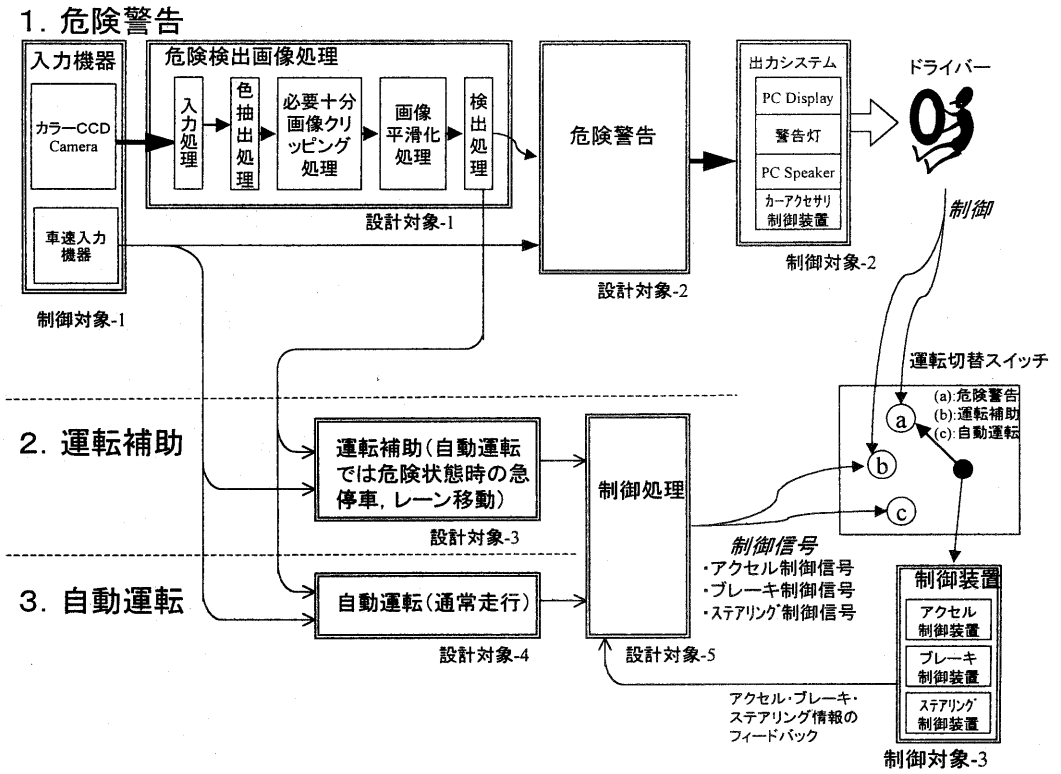

図2 全体モデル

るので，車の運転を全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

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，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３つの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言語にてモデリングを行う．

3. 2. 2 運転補助・自動運転

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運転補助の(7)～(13)の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，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．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 $f_x(t)$，$f_y(t)$，$f_\theta(t)$，$f_s(t)$との誤差を修正しながら新たなアクセル，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

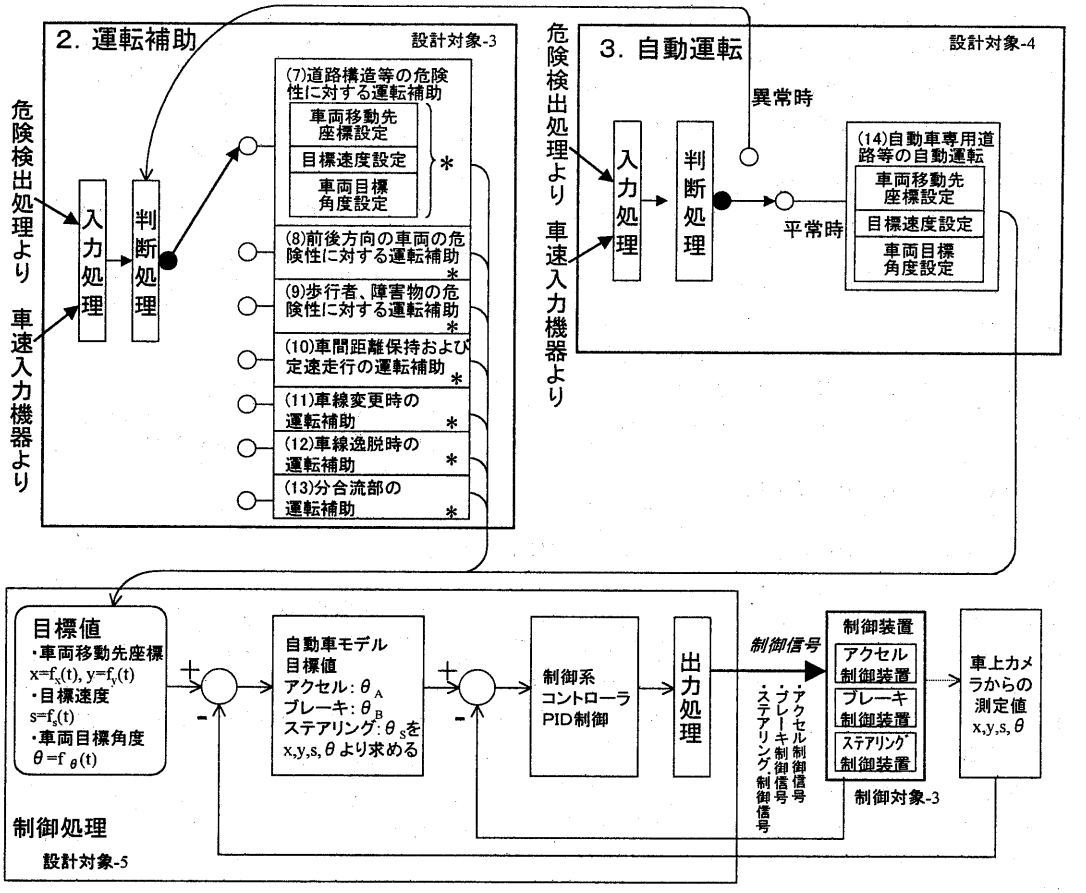

図3 運転補助・自動運転モデル

とに目標値との誤差を修正しながらＰＩＤ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）を例に挙げる．トップ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為である．あ

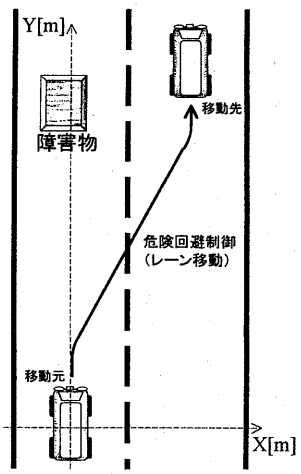

図4 危険回避制御の様子

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，図5 (b)はY座標の時間に対する推移を示している．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を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為，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．

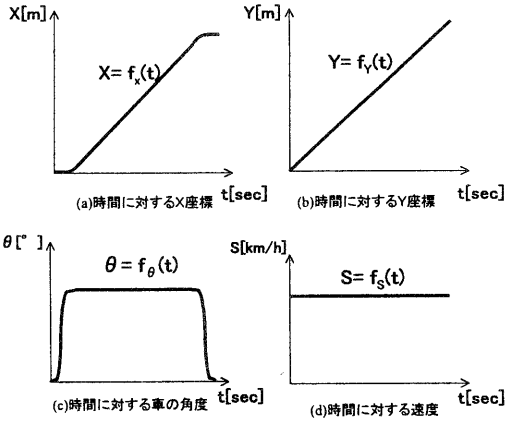

図5 時間に対する制御目標関数

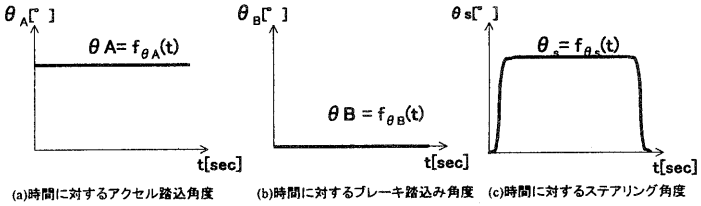

図6 アクセル，ブレーキ，ステアリングを制御する為の制御目標関数

また，これら $f_x(t)$，$f_y(t)$，$f_\theta(t)$，$f_s(t)$から求められるアクセル，ブレーキ，ステアリングを制御する為の制御目標関数 $f_{\theta A}(t)$，$f_{\theta B}(t)$，$f_{\theta S}(t)$を図6 (a) (b) (c)に示す．

図6 (a)は時間に対するアクセルの踏込み角度を示している．このグラフは図5 (d)の時間に対する速度の推移より求まり，アクセルの角度は移動元から移動先まで一定の値を示している．

図6 (b)は時間に対するブレーキの踏込み角度を示している．このグラフは図5 (d)の時間に対する速度の推移より求まり，ブレーキの角度は移動元から移動先まで一定の値（0[°]＝ブレーキを踏んでいない）を示している．

図6 (c)は時間に対するステアリングの角度を示している．このグラフは図5 (c)の時間に対する自動車の角度の推移より求まり，ステアリングの角度は移動元から徐々にその角度が大きくなり，ある角度で値が一定になり，移動先に近づくにつれ角度はもとに戻っている．これら，図6 (a) (b) (c)は，アクセル，ブレーキ，ステアリングを制御する為の制御目標関数 $f_{\theta A}(t)$，$f_{\theta B}(t)$，$f_{\theta S}(t)$として活用する．

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挙げる．

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7に示す．

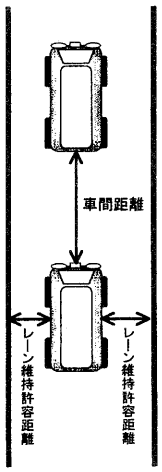

図7 平常時の自動運転の様子

画像処理にてレーンや前方車両を検出して，予め決められた車間距離，レーン維持許容距離内に自車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為の制御目標関数 $f_{\theta A}(t)$，$f_{\theta B}(t)$，$f_{\theta S}(t)$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い，その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取りはファイルベースで行う．また，入力機器のカラーＣＣＤカメラのかわりに，予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，車速入力機器のかわりに，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．これら，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上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９３個の処理コンポーネントからなり，これら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．

ハードウェア・ソフトウェア機能のトレードオフの結果，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や時定数が高速である必要がある制御処理がハードウェアの機能担当となり，ユーザーに警告を行う処理などのユーザーインターフェース部分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トップダウン型論理回路設計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構成を Cut and Try 式に作り上げていくことができる．また，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トップダウン論理回路設計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，トップダウン型論理回路設計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 50k 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 Windows98 搭載ＰＣ上で動作する．処理内容は主にハードウェアの各種検知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．こ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，Windows98 搭載ＰＣにダウンロードする．

4．リアルタイムでの検証と考察

(1) FPGA実装キットによる検証

ＦＰＧＡ実装キットによる検証で用いるコデザイン評価セットを図８に示す．コデザイン評価セットは，ＦＰＧＡ評価ボードとソフトウェア動作用ＰＣ，そして両者をつなぐハードウェア/ソフトウェア・インターフェースボードからなる．

シミュレーションによる実証と同じ入力動画像をコデザイン評価セットに入力しその結果を制御対象ＰＣに出力したところ，危険警告の段階ではその出力結果が目標モデルの出力結果と同じであった．運転補助，自動運転でも目標モデルと同じ結果が得られる見込みである．

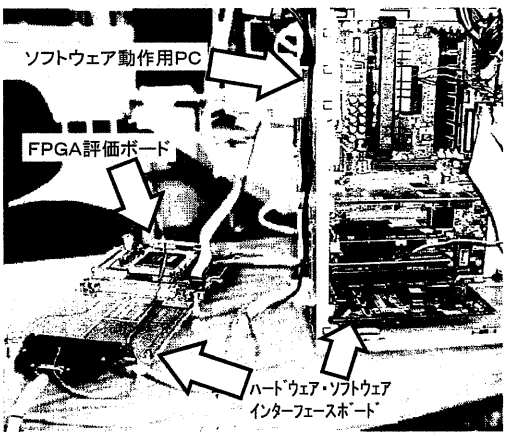

図8 コデザイン評価セット

(2) ラジコンカー走行による検証

本検証では図９のようなシステム構成にてラジコンカーを制御する．入力機器であるカラーＣＣＤカメラと，車速入力機器をラジコンカーに取り付け，無線通信にて危険検出画像処理に信号を送るようにする．また，制御処理から制御信号は，プロポを通してラジコンカーに送る．

(3) 考察

ハードウェアは画像処理部分，制御処理部分の回路を全てＦＰＧＡにマッピングし，現在チューニング中である．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，主にインターフェース処理と警告処理が機能分担され，その全てをＣ言語にて記述し，現在チューニング中である．ＦＰＧＡ実装キットによる検

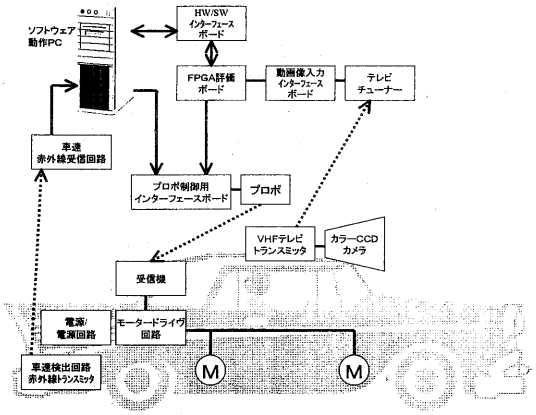

図9 実証レベル3 システム構成

証では，3．4のシミュレーションによる検証と同じ入力画像でを動作させて，その結果とシミュレーションの結果を比較する．

ラジコンカー走行による検証では，構築したミニチュアコースを自動運転し，前方に障害物を発見したら回避制御を行い検証を行う．

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，現段階で0.63[sec]以内であり設計目標を満たしている．このことから作成しているＩＴＳ安全運転の支援システムが，目標としていた全体モデルの機能を満たし動作することの見通しが得られた．

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る．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上でのハードウェア，ソフトウェア開発にスムーズに移行できる．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る．

5. まとめ

ハードウェア・ソフトウェア協調設計方式を提案し，ＩＴＳ画像処理開発・運転制御開発に適用して，安全運転の支援システムを構築し評価を行っている．その結果，目標とするシステムの制御対象・設計対象モデルを先ず作成し，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．

今後の課題は次の通りである．

(1)ＩＴＳ安全運転の支援システムの完成．

・トレードオフでハードウェアに分担された処理をハードウェア記述言語にて構築し動作確認する．

・システムとして必要とされる性能を満たすようにモデルのチューニングを行う．

(2)ラジコンカー走行制御環境におけるリアルタイムでの評価・検証を行う．

(3)本方式の検証の後は，ＩＴＳ安全運転の支援で今回対象としなかったサブサービスについても本コデザイン方式を適用し，本方式の有効性の検証を行う．

参考文献

1) Hisao Koizumi, Katsuhiko Seo, Fumio Suzuki, Yohsuke Ohtsuru, and Hiroto Yasuura, "A Proposal for a Co-design Method in Control System Using Combination of Models", IEICE Trans. on Information and Systems, vol. E78-D No.3, March, 1995, pp.237-247.

2) 遠藤 祐, 小泉 寿男, "ITS 情報処理開発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", 情処研報, Vol.99, No.ITS-1, pp.13-20(1999)

3) Yu Endo, Hisao Koizumi, "A Hardware/Software Co-design Method of ITS Image Processing Development", IEEE Computer Society, Proceedings of the Seventh International Conference on Parallel and Distributed Systems(ICPADS2000), INDAP-4, pp.415-420.

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, T.IEE Japan, 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 (Aug.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, 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,"ﾚｰｻﾞﾚｰﾀﾞ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ﾎﾟｼﾞｳﾑ,SAD-5-6,pp.97-98(1998).

11) 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).