高度交通システム　8－14
(2002.　3.　5)

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

吉田 健† 飯田 庸介‡ 井上 聡‡ 小泉 寿男†

†東京電機大学大学院理工学研究科
〒350-0394 埼玉県比企郡鳩山町石坂

‡東京電機大学理工学部
〒350-0394 埼玉県比企郡鳩山町石坂

E-mail: {ken,yothsuke,satoshi}@itlab.k.dendai.ac.jp koizumi@k.dendai.ac.jp

あらまし 本論文では，システムの作成において設計目標を満たすようなハードウェアとソフトウェアのトレードオフを迅速に行う事のできるコデザイン方式を提案する．本方式は，まず設計目標と全体モデリングを行い，続いてハードウェア・ソフトウェアのトレードオフ，再設計を行った後，両者を評価セット上で動作させる．この流れを繰り返しながら，目標を満たすシステムに近づけていく方式である．本方式を，ＩＴＳ(Intelligent Transport Systems)運転支援システム設計に適用し評価を行った．このシステムのリアルタイムでの検証方法として，評価セットを用いたパソコンによる動作確認と，ラジコンカーによる動作確認の両者を用いた．

キーワード コデザイン，協調設計，トレードオフ，ITS，運転支援システム

A hardware/software co-design method in an ITS operation support system, and its verification method

Takeshi YOSHIDA†, Yousuke IIDA‡, ,Satoshi INOUE‡, and Hisao KOIZUMI†

† Graduate School of Science and Engineering, Tokyo Denki University
Ishizaka, Hatoyama-mati, Hiki-gun, Saitama, 350-0394 Japan

‡ Department of Computers and Systems Engineering,Tokyo Denki University
Ishizaka, Hatoyama-mati, Hiki-gun, Saitama, 350-0394 Japan

E-mail: {ken,yothsuke,satoshi}@itlab.k.dendai.ac.jp koizumi@k.dendai.ac.jp

Abstract This paper proposes a co-design technique for system development that support quick trade-off between hardware and software to meet a desig goal.In this way, firstly a user determines a design goal and builds a total model, then make a trade-off decision between hardware and software. After that, the user redesigns the model and run both the previous model and the redesigned model on a evaluation kit.Repeating this sequence, our technique makes a design closer to a system that meets the goal.We applied this process to design a driving support system of ITS(Intelligent Transport System) and evaluated the technique.For verification of the system on real time, we used both demonstrations of a evaluation kit on computer and of a radio-controlled model car.

Key words co-design, trade-off, ITS, operation support system

1. はじめに

近年,組み込み機器の機能が複雑化するに伴なってハードウェア開発，ソフトウェア開発も大規模化,複雑化してきている.そんな中,市場からの要求で開発期間の縮小が求められる,という状況で,従来のハードウェア開発とソフトウェア開発を別々に行うという分業作業では,この要求に答える事が難しくなってきている.そこで,ハードウェアとソフトウェアの混合システムを効率的に設計を行うハードウェア・ソフトウェア　コデザイン（以下コデザイン）方式が注目されている．組み込みシステムを設計するにあたっては，システム仕様の記述方法や，ハードウェアで実現する部分とソフトウェアで実現する部分のトレードオフ（機能分担）が重要になっている.

本論文では,ハードウェアとソフトウェアのトレードオフを定量的，かつ迅速に行うコデザイン方式を提案し,その評価・検証を行う.本方式を ITS（Intelligent Transport Systems）における安全運転の支援システムの危険警告，運転補助，自動運転に適用し，検証を行った.

2. コデザインフロー

提案するコデザイン方式のフローを図１に示す.以下，詳細を述べる.

（1）設計目標と全体モデリング

まず，作成しようとするシステムの目標である設計目標を決定する.これは，後に行うハードウェアとソフトウェアのトレードオフの際の判断要素としても用いる.これをもとにシステムが導入される側である制御対象モデルとそのシステムそのものである設計対象モデルをＣベースによって記述する.制御対象モデル，設計対象モデル両者を連動させ，シミュレーションによってその動作を確認する.制御処理部分についてはＭＡＴＬＡＢのＳＩＭＵＬＩＮＫにてシミュレーションし動作確認を行う.

（2）ハードウェアとソフトウェアのトレードオフ

ハードウェア・ソフトウェア機能のトレードオフによって，ハードウェアで実装する部分とソフトウェアで実装する部分を決定する.この際，設計目標を参考にシステム仕様を満たす機能分割を行う.本研究では，システム性能が設計目標（処理速度）を満たし，かつコストミニマムであることを目標とする.機能分担の詳細に関して，以下に示す.

まず，Ｃ言語にて作成されている設計対象モデルを処理コンポーネント毎に分割し，それぞれのコンポーネント毎のソフトウェア上での処理速度を測定する.その結果から，処理速度的にソフトウェアでは実現不可能であると思われるコンポーネントはハードウェアに機能分担を行う.次に未決定コンポーネ

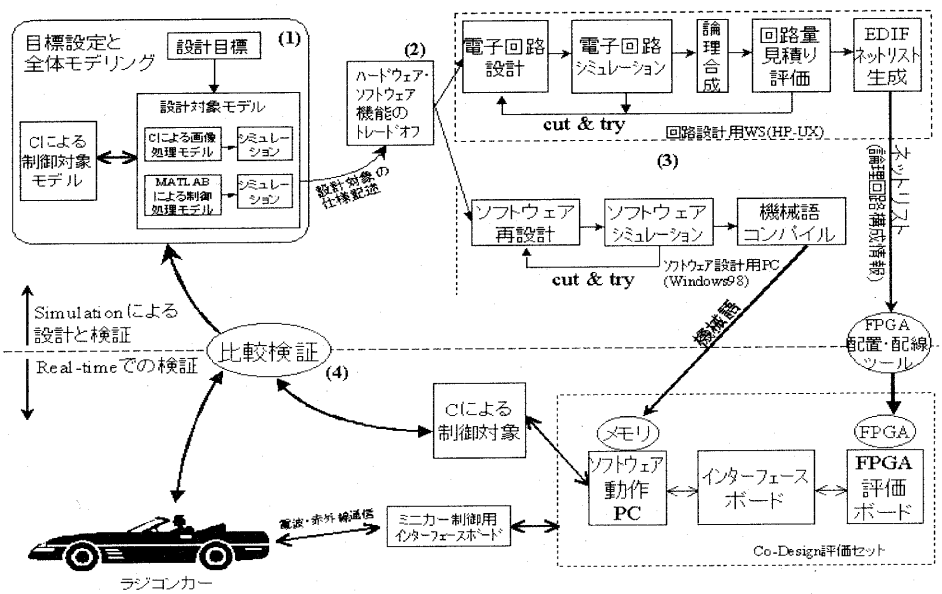

図１　コデザインフロー図

ントを含む全てのコンポーネントに対して同系列処理でのグルーピングを行うことにより，未決定コンポーネントに対する後の割当て作業回数を減少させる．グルーピング作業終了後，再び処理時間の見積もりを行い，処理時間的に実現不可能なコンポーネント群はハードウェアに割当てる．次に詳細分類にて，未決定コンポーネント群に対してハードウェア・ソフトウェアの全通りの組合せをシミュレーションし，処理時間の測定をする．その測定結果，システムの仕様要求を満たし，かつハードウェアに分担される部分の最も少ないものを選び，トレードオフの結果とする．

（3）ハードウェア，ソフトウェアの設計

トレードオフ後，ハードウェア，ソフトウェアに分担されたものをそれぞれ電子回路設計，ソフトウェア設計を行う．ハードウェアに分担されたものは，ハードウェア記述言語であるＶＨＤＬ(VHSIC Hardware Description Language)にて記述し，電子回路に置き換え，シミュレーションや回路量を繰り返しながら詳細設計を行う．設計結果を論理合成し，ネットリストを生成する．生成されたネットリストを用いて，プログラム可能なハードウェアであるＦＰＧＡ（Field Programmable Gate Array）にてハードウェア化を行う．ソフトウェアに分担されたものは再設計した後，Ｃ言語にてプログラミングし，動作結果がモデル作成時のものと同一のものであるか確認した後，設計目標を満たしているか確認しながら設計を行う．

（4）リアルタイムでの検証

ハードウェアに機能分担され、ＦＰＧＡでハードウェア化されたものとソフトウェアに機能分担されコンパイルされたものをインターフェースボード経由にて動作させ，動作結果を制御対象ＰＣに出力する．出力結果とモデル作成時のシミュレーション結果とを比較検証し，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目標を満たすシステムへ近づけていく．制御対象ＰＣに出力していた制御信号や警告信号をラジコンカー制御装置に入力し，ラジコンカーによるスモールスケールでの走行制御を検証する．

3. ITS 安全運転支援システムへの適用

ＩＴＳには９つの開発分野があり，本研究ではその中の１つである安全運転の支援の危険警告，運転補助，自動運転を検証の対象とする．

ドライバーは，通常，前方の道路状況やサイドミラー・バックミラーなどによる後方の状況など様々な状況を認識し，自分の状況を理解，判断しステアリング操作，ブレーキ操作，アクセル操作などを行っている．この人間による動作にはどうしても間違いなどがつきまとってくる．よって，見落としをアラームなどでドライバーに知らせるシステムである危険警告システム，ドライバーの不注意などに対応した運転補助システム，すべてをシステムに任せる自動運転システムなどの運転支援システムの開発が望まれている．

本研究で作成する運転支援システムの概要を図２に示す．ＣＣＤカメラにて前方画像を入力し，画像処理を行い危険を検出する．処理結果をドライバー，または運転支援システムへ出力し車両を操作するものである．

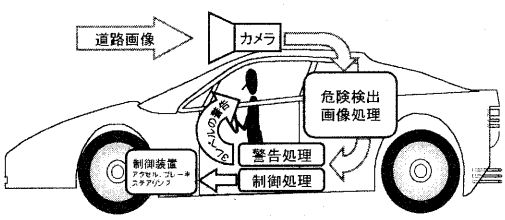

図２　運転支援システムの概要

4. 設計とシミュレーション

4.1 設計目標と全体モデリング

（1）設計目標

作成する安全運転の支援システムの設計目標を表１に示す．共通項目とは，危険警告，運転補助，自動運転の３つの利用者サービスでの目標と条件の事である．制御処理の項目には，運転補助，自動運転

表１ 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

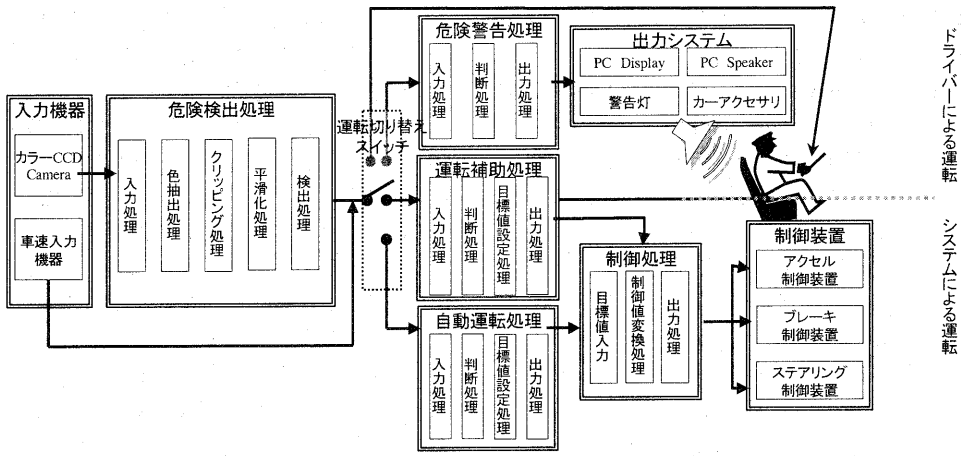

図３　全体モデリング

での目標と条件を示している．ＩＴＳ運転支援システムを設計する際には，表１に示した設計目標に加えて，使用するＣＣＤカメラの性能や制御対象ＰＣの性能などの設計環境条件も十分考慮する．

（２）全体モデリング

続いて，ＩＴＳ安全運転システムの全体モデル図を図３に示す．入力機器のＣＣＤカメラにて前方からの画像を撮影し，その画像を危険検出画像処理の入力処理が受け取る．危険検出処理部では，色抽出処理，必要十分画像クリッピング処理，画像平滑化処理を行い，前方の車両を含む障害物の有無やその障害物までの距離（近距離・中距離），走行位置などの検出を行う．検出結果を出力処理で危険警告処理，運転補助処理，自動運転処理へそれぞれ出力を行う．また，入力機器の車速入力機器からの車速データを危険警告処理，運転補助処理，自動運転処理へそれぞれ出力を行う．

危険警告，運転補助，自動運転のスイッチの切り替えはドライバー自身によって行うものとする．危険警告を選択時では，危険検出処理の結果から判断処理を行い，警告灯，アラーム，カーアクセサリなどの制御出力システムを制御する事により，ドライバーに危険内容を知らせるものとする．運転補助，自動運転モデルでは，危険検出画像処理の結果から判断処理を行い，その結果によって制御の目標値を決定する．自動車モデルの目標値に変換した後，ＰＩＤ制御にてアクセル制御，ブレーキ制御装置，ステアリング制御を行い，車両を制御する．

図３に示した全体モデルから，設計対象モデルと制御対象モデルをＣ言語にて

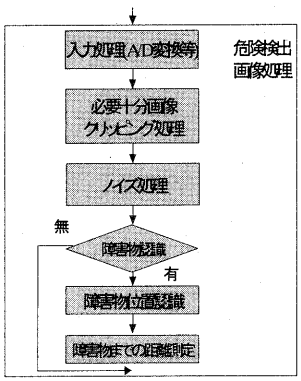

図４　画像処理アルゴリズム

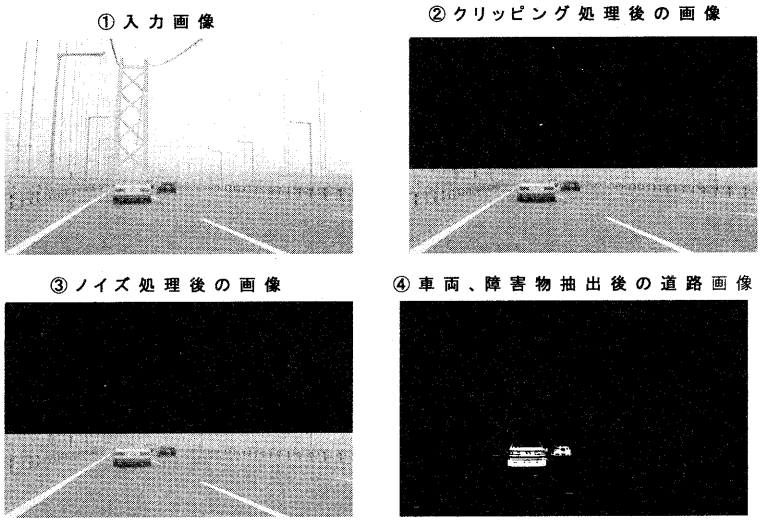

図５　画像処理結果

作成した．

作成した設計対象モデルの危険検出処理部分についての概要を図４に示し，その詳細を以下に述べる．危険検出画像処理の入力処理において，画像処理をデジタル処理で行う為，A/D変換を行う．まず，必要十分画像クリッピング処理にて，入力画像の障害物を認識する際に不必要となる部分を省き，必要な部分だけの抽出を行う．続いて，画像のノイズを取り払う画面平滑化処理（ノイズ処理）を行う．

以上の処理を行った画像から，差分処理などを行い，前方の障害物位置の認識を行う．図５に実際に行った画像処理の様子を示す．認識した障害物を入力画像の画面したからの距離から，自車両との距離を計算し，その距離情報から，危険レベル（遠距離・中距離・近距離）の認識を行う．危険レベルの情報を，危険検出処理,運転補助・自動運転処理へと送る．

続いて，制御処理部分のアルゴリズムについて解説する．作成したアルゴリズムをＳＩMULＩNKにて作成し，制御処理の異物回避動作のシミュレーションを行う．ここでは，前方中距離に障害物がある場合の回避動作を例に挙げる．

自車両が前方に障害物を発見し，レーン移動によって右方向へ回避する様子を図６に示す．自車両の進行方向をＹ軸方向とし，自車両に対して右方向をＸ軸方向とする．これから，時間に対する制御目標関数として，図７に示すような時間に対するＸ座標，Ｙ座標，車の角度，速度を決定する事ができる．この４つのパラメータから，実際，車両制御に必要な関数である，アクセル，ブレーキ，ステアリングを制御する為の関数を求める事ができる．今回の回避動作では，中距離における回避動作の為，速度の変更は行わないので時間に対するアクセル制御，ブレーキ制御は，一定のものとする．自車両の時間に対するＸ座標の推移から時間に対するステアリング角度求め，ＰＩＤ制御によるシミュレーションを行った．また，ＰＩＤ制御を行った時間に対するステアリング角度から，

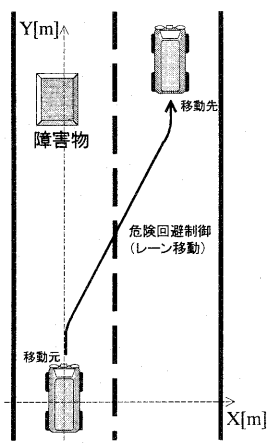

図６　障害物回避の様子

Ｘ座標の推移の様子を求めたところ，目標値との誤差は，許容範囲であることを確認する事ができた．シミュレーションした結果を図８に示す．

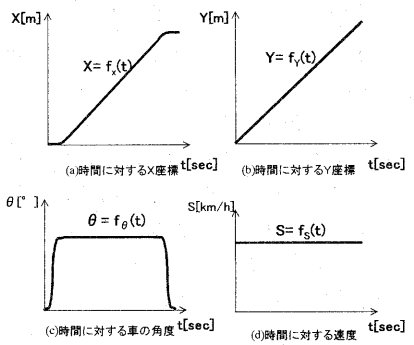

図７　時間に対する制御目標関数

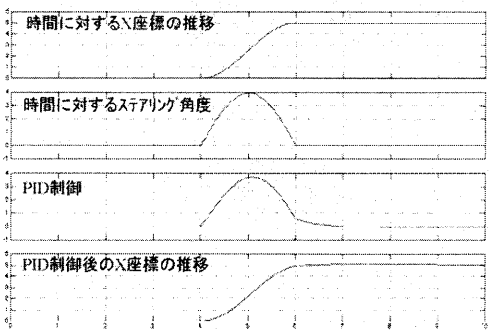

図８　制御処理シミュレーション

4.1 ハードウェア・ソフトウェアの設計

（１）ハードウェア・ソフトウェアのトレードオフ

ハードウェアとソフトウェアのトレードオフの結果，処理データの膨大な画像処理による危険検出処理部などがハードウェアの機能担当となり，制御における判断処理などの部分がソフトウェアの機能担当となった．

（２）ハードウェア・ソフトウェアの設計

①ハードウェアの設計

ハートウェアに機能分担されたコンポーネントは，ハードウェア記述言語であるＶＨＤＬにて電子回路設計，論理合成を行い，ネットリストを生成し，フロアプランを経て，ＦＰＧＡへ書き込みを行う．

②ソフトウェアの設計

ソフトウェアに機能分担されたコンポーネントは，Ｃ言語にて再設計した後，コンパイルし，ソフトウェア動作ＰＣ上で動作させる．

5. リアルタイムでの検証

リアルタイムでの検証環境を図９に示す．ＣＣＤカメラからの画像を受信機で受信し，ＶＤＥＣによってＡ/Ｄ変換しＦＰＧＡでデジタル化された画像情報の画像処理を行う．画像処理によって前方の障害物を認識し，ＩＳＡバス経由にてソフトウェア動作ＰＣにその情報が送られる．ソフトウェア動作ＰＣでは，検証ボード上での検証の場合出力結果を表示し，ラジコンカーによる検証の場合には制御信号をラジコンカーに送り，動作を確認する．以下にコデザイン評価セットによる動作確認とラジコンカーによる動作確認の詳細を示す．

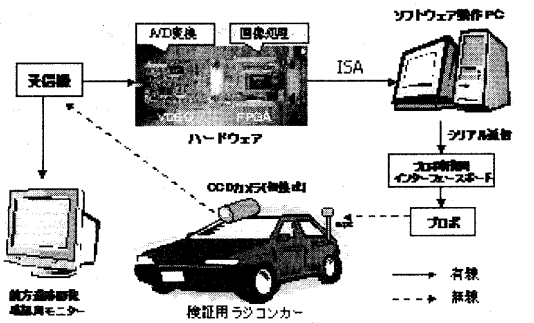

図９　リアルタイムでの検証環境

5.1.1　検証ボード上での検証

コデザイン評価セット上での検証では，“中距離前方左に異物があり”，“中距離前方右に異物があり”，“中距離中央に異物があり”，“近距離に異物あり”など，シミュレーションとほぼ同じ出力結果が得られた．

5.1.2　ラジコンカーによる検証

ラジコンカーによる安全運転支援モデルを図１０に示す．ラジコンカーによる検証では，通常時にはドライバーによる操作を行い，前方障害物発見時にはプロポを通してラジコンカーの制御を行いつつ，

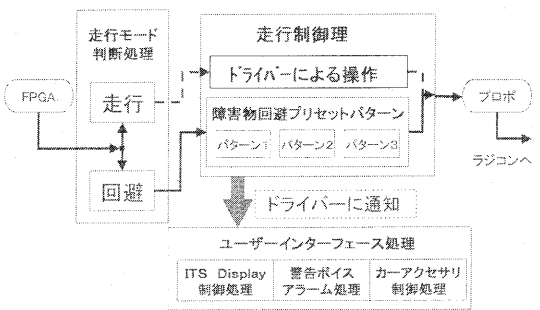

図１０　ラジコンカーによる安全運転支援モデル

カーアクセサリなどのユーザーインターフェース制御を行いドライバーに危険を警告するものとする．ラジコンカーの制御においては，フィードバック制御を用いずに障害物と自車両との距離（中距離，近距離）や障害物の位置（左右，中央）などの情報から理想的な回避動作を行えるように回避パターンをあらかじめ作成しておき，そのパターンを用いる事により回避動作を行う障害物回避プリセットパターンを用いる事とする．前方に異物を発見し，システムによる回避動作中の実行画面を図１１に示す．ラジコンカーによる検証では，障害物の位置によって，左回避動作，右回避動作，停止による障害物回避するなどの動作確認を目視にて確認することができた．

図１１　実行画面

6. 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運転支援システムの設計･開発を行う事でその評価を行った．リアルタイムでの検証（検証ボート，ラジコンカーによる検証）の結果，設計目標を満たすシステムの構築を行えた事を確認した．

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] ITS に係るシステムアーキテクチャ：：http://www.iijnet.or.jp/vertis/j-frame.html

[3] 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,pp1118-1126,2000.

[4] 高木聖和，久野晃，中村哲也，“レーザ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,pp97-98(1998)

[5] 田中宏明，“自動車から見たＩＴＳ-安全な車の実現”，自動車技術 Vol.55,No.11,2001

盛光印刷所