研究速報

海中ロボット研究のための汎用仮想海中環境シミュレータMVS

MVS: The Virtual Underwater Environment Simulator for Underwater Robotics

荒牧浩二*・黒田洋司*・浦　環*・川野　洋*
Koji ARAMAKI, Yoji KURODA, Tamaki URA and Hiroshi KAWANO

1. はじめに

海中ロボットの実験には莫大な金銭的労力的コストがかかるため，海洋と同様に利用者が自由にアクセスし得る仮想的な海を計算機ネットワーク上に恒常的に展開することで，様々な開発局面で海洋実験と同じスタイルで検証を行えるシミュレーションシステムが必要である（図1参照）．そのシミュレータは動的に複数エージェント（ロボットや障害物）に対応可能で，複数の研究者が同じシミュレーション世界を共有できるべきであり，実ロボットとの接続機能によりプログラムの開発や移植の手間が押さえられるべきである．本研究では，これらの条件をふまえ，自律型海中ロボット研究用の汎用システムMulti-Vehicle Simulator（MVS）[1]を開発し提案する．

2. 仮想海中環境シミュレータMVSの概要

MVSでは抽象的なシミュレーション世界をWorldと呼び，同時に複数の独立したシミュレーションを行えるよう複数のWorldが存在し，全Worldの集合をUniverseと呼ぶ．シミュレーションに関与する主体をAgentと総称し，それはUser（研究者）とEntityに分類され，Entityはさらに Obstacle（海底地形や試験水槽などを含む障害物），Vehicle（海中ロボット），Station（通信機能などを備えた海中基地）に分類される．

任意の形状の海底地形や障害物が存在する環境を想定したり，複数ロボットによる協調行動の研究に対応するために，MVSは任意の個数の任意のタイプのEntityを動的にWorldに加えたり削除する機能も備えてある．また，実際の海での実験では複数の研究者が立ち会い，指示や観察をするように，MVSが展開する仮想的な海にも複数のUserが任意にアクセスでき，原理的には地球上のどの研究者も計算機ネットワークにより同一のシミュレーション世界を共有できる．

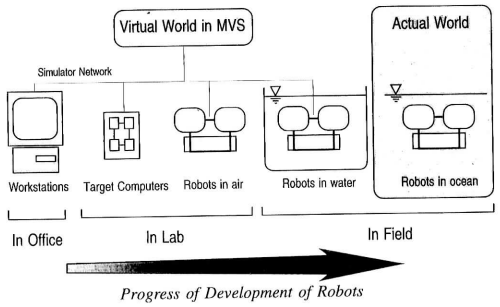

図1　MVS利用による統合的な研究環境

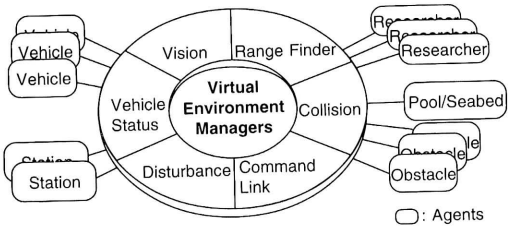

図2　分散システム構造

MVSを構成する主たるプロセス群は，ManagerとPerformerの2種類に大別される（図2参照）．Managerは Agent同士の相互作用関係を仲介し，仮想海中環境を計算機ネットワーク上に実現するための中心的なプロセスで，数種類あり，それぞれの詳細は3章で述べる．PerformerはAgentの役割をするプロセスで，一つのAgentにつき一つの独立したPerformerが実行される．これらのプロセスをネットワーク上の適切な場所で実行することにより，情報処理の負荷を分散させ高い処理速度を確保し，またグラフィクスに適した計算機で図3のようなGraphical User Interface（GUI）を実行するなど個々の計算機の特性を活かせる．

*東京大学生産技術研究所　第2部

研　究　速　報

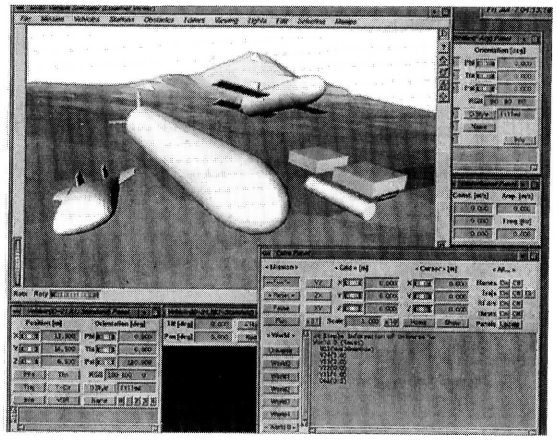

図3　Graphical User Interface

MVSでは，図4のように，各PerformerがManagerを通して他のPerformerと情報のやり取りを繰り返すことにより，全体のシミュレーションが進む．これらのやり取りは，表1に示されるような，イベント駆動型の一連の手続きとして定義されており，ロボットに内蔵されたインターバルタイマや，研究者がUser Interface（UI）を通じて与える操作などをきっかけに実行される．

ロボットの制御アルゴリズムを一旦シミュレータ内で構築，検証した後，実ロボットのCPU用にそれを変換，移植する方法では手間がかかり，シミュレーションと実験を交互に繰り返すという研究スタイルをとりにくい．そこで実ロボットが計算機ネットワークに接続しMVSのAgentの一つとしてシミュレーションに参加できるように，MVSでは実時間動作性と外部システムとの接続機能を設計方針に組み入れた．そのため，研究者は最初から実ロボット用の制御プログラムを組め，能率よく研究を進められる．また，外乱やそれに伴うダイナミクスは試験水槽などで航行中の実ロボットに受け持たせ，ロボットが獲得すべき環境知覚のためのセンサ計測値，つまり実際には存在しない障害物を検知したかのような情報をMVS側が算出しロボットに提供するという新しい実験スタイルが可能である．これは，実験の手間を軽減する効果と，シミュレーションの精度を確保する効果を期待することができ，MVSの大きな特徴である．

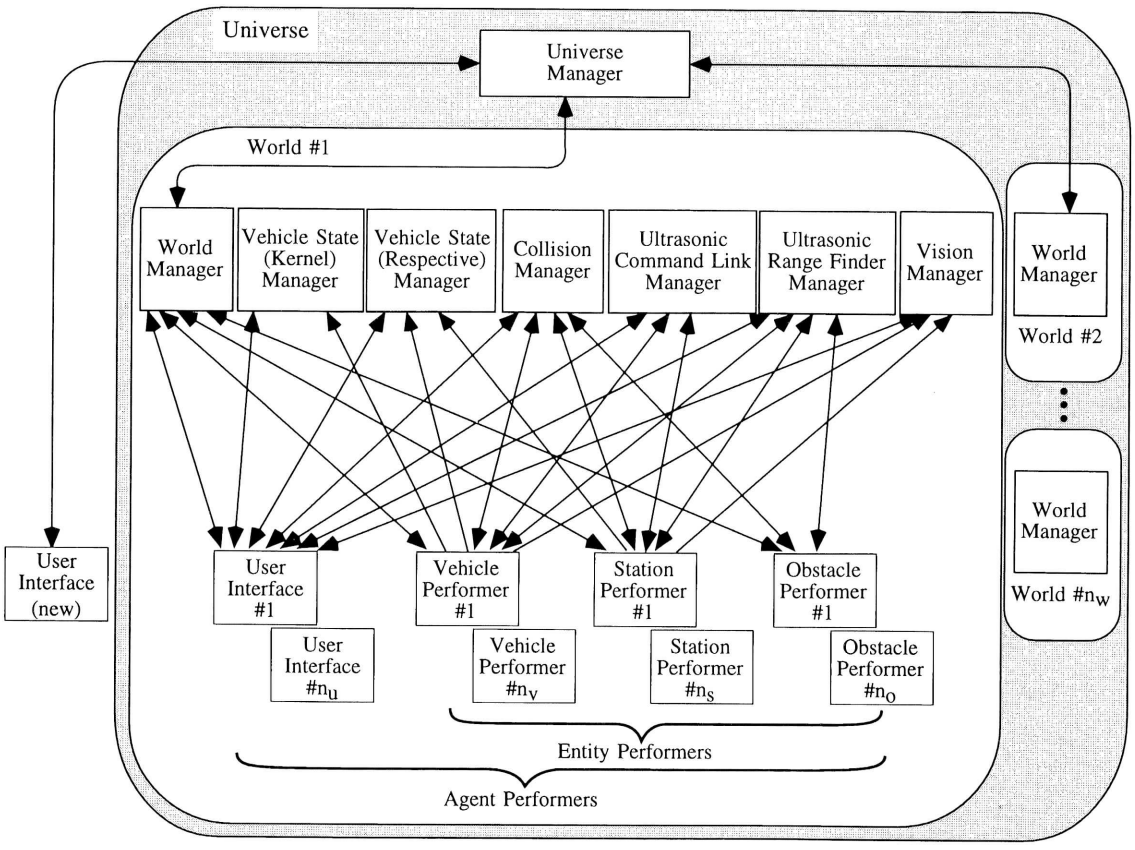

図4　MVSのソフトウエア構造

表1　代表的なイベント

No.	Eventの内容
01	System configurationの登録
02	System configurationの登録
10	UIがSystem configurationを入手
11	UIが全Worldの簡単な情報を入手
12	UIがWorldの詳細な情報を入手
15	UIがWorldに参加
16	UIのパスワードの確認
20	UIが新規UIに対するメッセージを登録
21	UIが他のUIにメッセージを送信
25	Managerが全UIにメッセージを送信
26	Entityが全UIにメッセージを送信
30	新たにEntityが起動
40	UIがEntityの位置・姿勢を変更
41	UIがObstacleの大きさを変更
42	UIがEntityにトリガを送信
50	UIがMission Modeを変更
51	UIが外乱パラメータを変更
60	定期的にVehicleが位置・姿勢などを報告
61	UIがVehicleの位置・姿勢などを入手
63	定期的にVehicle/Stationが内部状態量を報告
64	UIがVehicle/Stationの内部状態量を入手
67	定期的にVehicleが位置・姿勢を報告
68	定期的な衝突判定
70	定期的な超音波測距
75	通信用超音波の発信
90	UIがEntityの削除を指示
91	UI/Entityの終了

UI: User Interface

3. MVSのソフトウェアとハードウェア

MVSでは，クライアント／サーバモデル[2]に基づいたプロセス間通信による分散システム構造を採用している．個々のPerformerがクライアントで，通信関係や超音波測距センサなどの情報をロボットに提供したり，ロボットの状態量をUIに提供するための，機能毎に独立した複数のサーバがManagerである．

Managerには以下の8種類がある．このうち，(2)〜(8)は各Worldに一つずつ存在する．また，(1)〜(4)はシミュレーション時の利便性を，(5)〜(8)は物理的な現象を司るものである．

(1) Universe Manager：主にUserに対して各Worldでのシミュレーション状況の通知サービスを行う．

(2) World Manager：User間の伝言授受やEntity位置の設定などWorld内の非定常的な操作を仲介する．

(3) Vehicle State Kernel Manager：Vehicleタイプに依らない位置，姿勢などの状態量をUserに中継する．

表2　Ultrasonic Range Finder Managerのイベント

No.	Direction	Data
70-1	各Vehicle → URFM	VehicleのType, 位置・姿勢
70-2	URFM → 全Entity	VehicleのType, 位置・姿勢
70-3	各Entity → URFM	当Entityまでの全測距センサからの距離
70-4	URFM → Vehicle	全測距センサの最小化された計測値

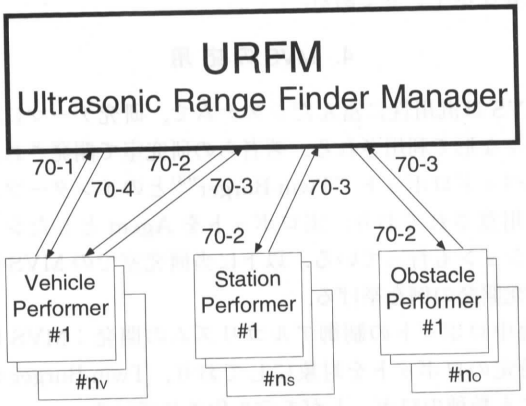

図5　Ultrasonic Range Finder Manager

(4) Vehicle State Respective Manager：Vehicleタイプに固有の内部状態量をUserに中継する．

(5) Collision Manager：Vehicleと他のEntityとの衝突判定を援助するためにVehicleの現在位置を中継する．

(6) Ultrasonic Command Link Manager：VehicleやStationの間の超音波による通信機能を司る．

(7) Ultrasonic Range Finder Manager：ナロービームの超音波測距センサの機能を司り，Vehicleの現在位置や距離データなどの集配と中継を行う（図5，表2参照）．

(8) Vision Manager：VehicleやStationに搭載されたビデオカメラが捉える画像情報を司る．

一方Agentの役割をするPerformerは，4種類がある．

(1) Obstacle Performer：海底地形などを含む障害物の役割をし，幾何学的な形状情報を持つ．衝突判定や超音波測距センサの計測値の実際の計算を行う．

(2) Vehicle Performer：海中ロボットの役割をし，ロボットの制御アルゴリズムとダイナミクス計算機能を合わせ持つ．これがロボット研究者の研究対象を表すので，研究者はシミュレーションに先立ちこのプログラムに自身の研究内容を組み込む．(1)と同様に衝突判定などの計算も行う．

(3) Station Performer：海中基地の役割をし，移動機能がない点以外はVehicle Performerに等しい．基地を含むミッション制御の研究に利用する．

(4) User Interface：シミュレーションに立ち会う研究者の

窓口となる．実行する計算機の特性に合わせてGUIや，文字ベースの端末装置に対応した汎用的なものなど，多様な形態をとる．

MVSが規定するプロトコルに則った情報の伝達がプログラム間で可能であれば，プログラムを実行するためのハードウェアを規定することは基本的には行わない．これはMVSの汎用性および様々なシステムとの接続性を高める重要なポイントである．

4. MVSの応用

MVSは汎用性に富んだシステムで，研究テーマに即して様々な形で利用される．著者らの研究室で開発されたテストベッドロボット“Twin-Burger”[3]とのインターフェースも用意されており，実ロボットをAgentとしたシミュレーションも行っている．以下に当研究室でのMVSによる研究開発の例を挙げる．

(1) 海中ロボットの制御アルゴリズムの開発：MVSは不特定のロボットを対象にしており，Twin-Burgerの他にも数種のロボットがモデル化されている．

(2) マルチロボット研究（ロボット個々とミッション全体の制御）[4]：実験には複数のロボットと大規模な設備，人員が必要であり，シミュレータは欠かせない．

(3) ロボットに搭載された超音波センサによる地形図作成[5]：センサの計測値の算出や，GUIによる障害物認識結果の3次元的な可視化にMVSを役立てている．

(4) 実験データの可視化による検証：航跡や測距センサ計測値などを可視化することで，研究者がロボットの活動中の挙動を把握し，問題点を調査し易くなる（図6参照）．センサの故障などが発見された実績もある．

(5) 実仮想世界の合成による実験[6]：2章で述べた新しい実験スタイル（図7参照）．

5. おわりに

複数エージェント，複数利用者対応型で，複数計算機に負荷を分散し，実時間性を確保し，実ロボットとの接続機能を備えた，シミュレーションシステムを提案，開発した．これは，海中用のみならず，原子力施設内の作業や惑星探査などの様々な分野で，自律型移動ロボットの研究開発に幅広く応用できる．

（1995年9月18日受理）

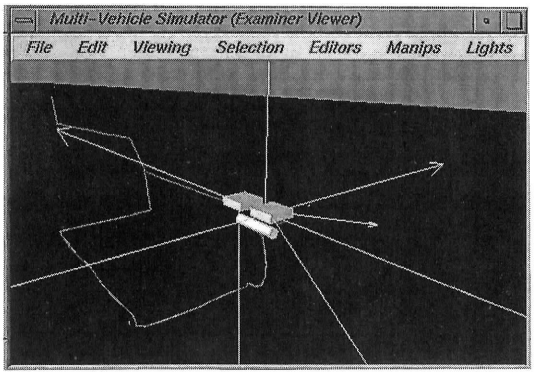

図6　実験データの可視化

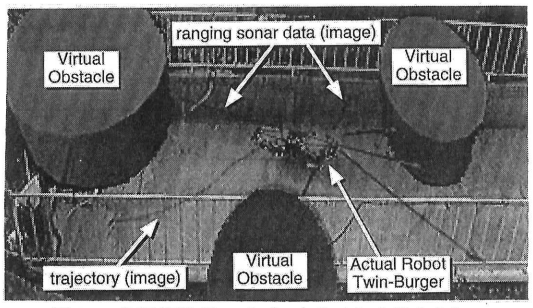

図7　実仮想世界の合成（ビデオ映像より）

参考文献

1) 黒田洋司，荒牧浩二，浦環，菅野崇，大和裕幸：“ネットワーク分散処理による海中ロボット用海中環境シミュレータ”，日本造船学会論文集，Vol. 178（1995）近刊．

2) D. Comer, D. Stevens 著，村井純，楠本博之訳：“TCP/IPによるネットワーク構築，Vol. 3”，共立出版（1995）．

3) 藤井輝夫，浦環，黒田洋司，荒牧浩二，能勢義昭，千葉裕之：“海中ロボットの知能化に関する研究（その1：汎用テストベッドの開発と水槽実験）”，日本造船学会論文集，Vol. 174（1993），pp. 903-916.

4) 黒田洋司，浦環，荒牧浩二：“Vehicle Control Architecture for Operating Multiple Vehicles”, Proc. of IEEE Autonomous Underwater Vehicle Technology '94 (1994), pp. 323-329.

5) 荒牧浩二，浦環：“潜水機による局所海底地形図作成およびパスプラニング”，日本造船学会論文集，Vol. 177（1995），pp. 437-445.

6) 黒田洋司，荒牧浩二，藤井輝夫，浦環：“A Hybrid Environment for the Development of Underwater Mechatronic systems”, Proc. of IECON '95, (1995) 近刊．