東海大学紀要情報通信学部
vol.4,No1,2011,pp.26-33

論文

ETロボコン2010による組込みシステムPBLの実施と評価

細川 達也[*1] 上陰 敏弘[*1] 今井 紘士[*1] 小沢 滋紀[*1] 山崎 亮太[*1]
塩満 博[*1] 萩沢 浩貴[*1] 樋口 拓哉[*2] 大金 淳一郎[*2] 清水 尚彦[*3]

The Operation and valuation of Embedded System PBL in ET-robot contest 2010

by

Tatsuya HOSOKAWA[*1], Toshihiro KAMIKAGE[*1], Hiroshi IMAI[*1], Shigenori OZAWA[*1], Ryota YAMAZAKI[*1], Hiro SHIOMITSU[*1], Hiroki HAGISAWA[*1], Takuya HIGUCHI[*2], Junichiro OGANE[*2] and Naohiko SHIMIZU[*3]

(Received on October 2, 2010 & Accepted on September 12, 2011)

Abstract

Usually students study embedded system in lectures and adopt Project Based Learning(PBL). Therefore, we did practical learning for developing embedded system with using PBL. The project members were 7 senior students who had basic knowledge of embedded system and 2 graduate students as instructors. For practical learning, we participated the ET-robot contest 2010. As a result, students could have practical learning with PBL and get essential knowledge of embedded system. Therefore, we verified the efficiency of our method to learn embedded system comprehensively.

Keyword: ET-robot contest, Project-Based Learning, Difference between simulation and real, Schedule managements

1. はじめに

本論文ではETソフトウェアデザインロボットコンテスト（ETロボコン）[1)]2010による組込みシステムのProject-Based Learning(PBL)実施を評価する．開発に必要な要素，開発工程，及び実装機能をあげ，プロジェクトにおける学習成果，反省点を明確にすることで全体的にPBLがどのような効果をもたらしたかを評価する．

組込みシステムに関しての学習では，実際の開発環境で具体的な成果物を作成するのが効果的である．そのため多くの大学，企業でPBLが採用されている．そこで私たちはPBLを用いて試行錯誤しながら実際に開発をすることで，実践的な学習を行った．PBLを用いた学習の例としては[2)]をはじめ，様々な例がある．

2. ETロボコンとは

ETロボコンは組込みシステムについて企業の新人研修から個人まで多様な参加者の教育の場として提供される．題材としてLEGO MINDSTORM NXT(NXT)[4)]にFig.1のような各種センサを取り付けたものを使用し，実行環境としてnxtOSEK[5)]と拡張ファームウェアを用いる．競技はNXTに独自のプログラムを組込み，コースを周回させたタイムの結果と，モデル図の評価の２つの総合点数を競う. そのため周回タイムの短縮が非常に重要な要素となる. 走行体はラインをトレースして走行するが，ラインのないところを走行することも認められている．また，コースにはインコースとアウトコースが設けられ，同時に二台が走行する．それに加え，コースやルールも各年度によって異なる．例としてFIg.2とFig.3に前年度と今年度とでのコースの差異を示す．主催者側が提供したサンプルコードを利用し，これをもとに参加者が独自の技術を加える方法をとることで，開発を進めやすくしている．ライブラリとしてセンサ類から値を取得するAPIや倒立振子制御関数が開発の敷居を低くし，教育に適した提供がされている．

サンプルプログラムは光センサ値によるモータのon/off制御によりライントレースをする．サンプルコードは，旋回量と，速度制御変数の値を調整するだけでコースの周回速度を向上できる．

競技では他にモデル図の提出が義務となる．モデル図は様々な観点から評価され，採点される．モデル図の評価は走行体の周回タイムと同様に総合順位を決める重要な要素となる．そこで，走行体の周回タイムを短縮させ，モデル図に設計手法を明確に記載することが上位入賞への必須条件である．

input	output
Light sensor	Sonor
Gyro sensor	Motor
Touch sensor	
Sonor	
Motor	

Fig.1　List of sensors

[*1] 情報理工学部 ソフトウェア開発工学科４年生
[*2] 工学研究科　情報通信制御システム工学専攻１年生
[*3] 情報理工学部 組込みソフトウェア工学科 教授

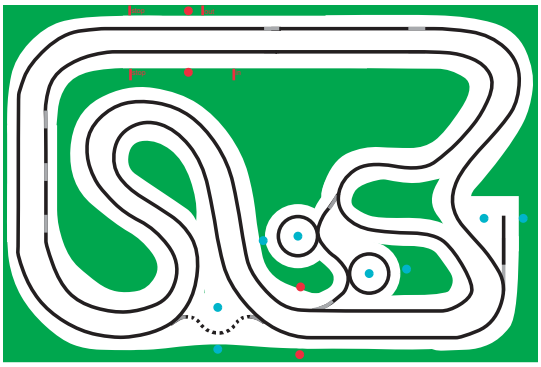


Fig.2　ETrobo2009 course

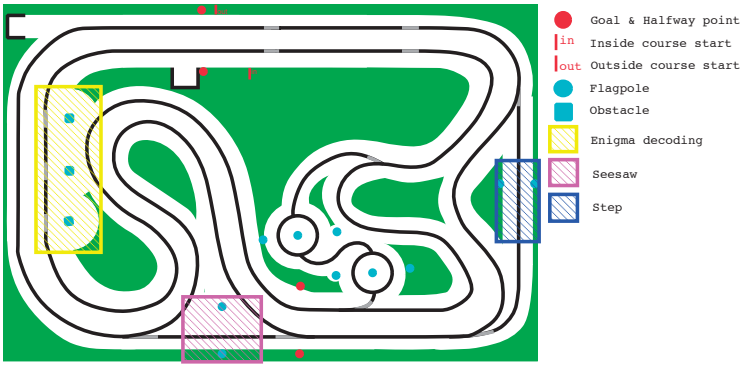


Fig.3　ETrobo2010 course

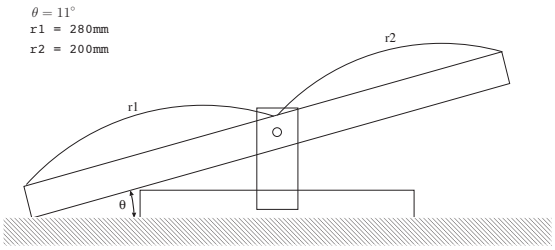


Fig.4　Seesaw model

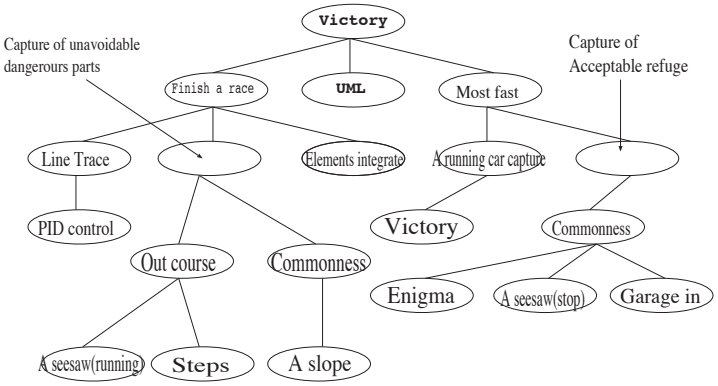


Fig.5　Goal oriented model

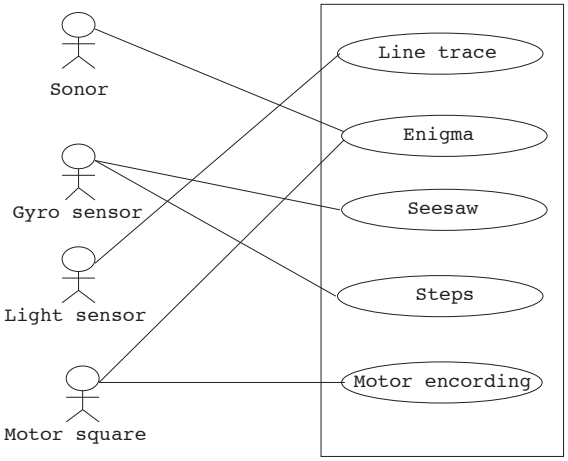


Fig.6　Usecase model

3.　PBL 実施内容

この章では開発開始にあたってのスケジュール組みと実際に用いた設計手法，及び各機能についての解説を述べる．

ET ロボコン参加にあたって，６月から９月までを開発期間とし，組込み開発を専門的に学び，基礎知識を十分に持つ学部生７人とそれを指導する大学院生２人との計９人のメンバーでプロジェクトを開始した．PBL のため，学部生からプロジェクトマネージャを立て，スケジュール作成を一任した．開発については最初にゴール図を作成し，それを元に機能分解するためのユースケース図を作成した．そのためユースケース図には主な機能に限って記載した．

また，実装手法についての学習を同時進行で行った．そのため必要な技術，手法は全て開発と同時進行で学んだ成果を用いた．

3.1　スケジュール

ET ロボコン南関東大会では試走会が計４回開催された．そこで試走会をマイルストーンとして開発スケジュールを作成した．Fig.7 に予定スケジュールと実際のスケジュールの差異と説明を示す．

3.1.1　予定スケジュール

一回目の試走会までは Fig.5 のゴール図，Fig.6 のユースケース図を参考に各機能毎の UML 製作のために時間を取った．これはユースケース中のアクターで示された各機能の位置づけや初期設計を明確にする目的を持つ．また，同時進行で数学モデルを立て，機能実装のための基礎データ取得を行う．

その初期設計と基礎データを用いて二回目の試走会までに走行に必要な要素を実装する．試走会の間が一ヶ月半程空き，十分な時間が取れると判断し，この時期までに基本的な機能を実装させ，試走会でのデータ収集を行える環境を整える．また，走行に必須ではない機能は三回目の試走会での完成を目標とした．これらの機能は実装されなくとも走行は可能であり，技術的にも高いレベルが要求されるため，時間に余裕を持たせている．

各機能は実装されしだい，機能毎の UML 拡張作業に入る．新たに作成した機能や変更を加えた機能を UML に表す．各機能の UML 拡張が終わったところから総合 UML 作成担当へデータを提出し，担当がそれらをまとめる．

UML 提出後は機能拡張期間とした．新たな改善点などを改良し，走行戦略に沿った走行をするための機能調整などをこの期間に行う．この期間は大会当日まで設け，当日まで機能拡張できるスケジュールとした．

3.1.2　実際のスケジュール

一回目の試走会まではどの機能も基本 UML 作成と基礎データ取得に徹し，完成した機能からシミュレーション上最適と思われるアプローチで機能実装に取りかかった．この際，シミュレーションと実際の動作の差異が激しく，基本実装は UML 提出締切り日まで長引いた．そのため機能実装と UML 拡張を同時進行で行なわねばならず，プロジェクトメンバーに非常な負担を強いた．その後も機能実装に取り掛かり，各機能を総合して実装できたのは大会当日の朝となってしまった．この時点でも時間の関係上実装できない機能があり，完全な総合実装とは言えない状況であった．

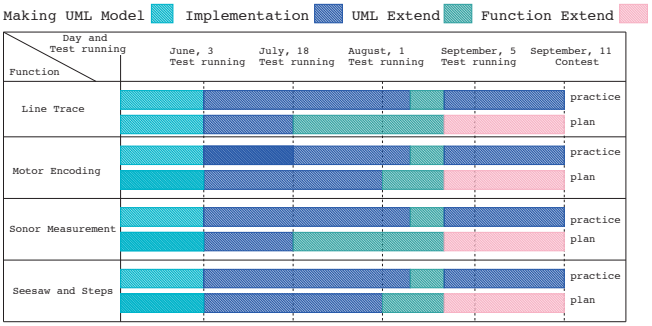


Fig.7　Schedule of plan and practice

3.2　設計手法

機能間の円滑なコミュニケーションを図り，かつ各自の負担を軽減するため，各機能の担当者をメインとサブに分け，各自がどれかのメインと他のサブを担当した．複数の担当者により不測の事態によって機能開発の中断を未然に防いだ．

数学モデルを逐次取り入れることによって効率的に基礎データの計測を行い，設計ミスのない実装を目指した．

3.3　設計，開発

各機能についての設計，開発過程を示す．今年の ET ロボコンで採用されている主な難所はシーソーと階段，エニグマデコーディングである．このうちシーソーと階段はアウトコースに，エニグマデコーディングはインコースに設置されている．各難所の詳細はそれぞれの項目で紹介する．全体的にシミュレーションと実際の動作にかなりの差が生まれた結果となった．

3.3.1　UML

Fig.8 の UML は MARTE[7] を用いて記述したものである．MARTE とは組込み開発用の UML であり，UML 拡張プロファイルとして定義される．MARTE ではソフトウェアとハードウェアを同じ言語で分離して記述できる．ソフトウェアは機能を実現するプログラムを，ハードウェアはセンサから得られる物理量を定義し明確に区別した．今回は機能毎の基本的な UML を作成したのちに機能拡張し，UML を拡張するが，このような手法が可能であるのは MARTE が非常に拡張性，分析性に特化しているからである．また，モデル記述ではソフトウェアとハードウェアの関連性が明示できるよう気を配った．特にどのハードウェアがどのソフトウェアと繋がっているかを強調した．そしてソフトウェア側では記述単位をタスクとした．これによってモデルの肥大化を防ぎ，タスクの値のやりとりと，関連性を明確にした．ここで，MARTE ではクラス図の中に変数を記述することができるため，これを用いて全体の簡略化を目指した．しかしタスク毎に変数の数にばらつきがあったため，グローバル変数をパッケージとして扱い，各タスクがローカル変数をインポートする記述がこの問題を解決した．

今回，NXT 外部に運動として力を与える倒立振子制御はハードウェアとして定義，全体の可読性を向上させた．

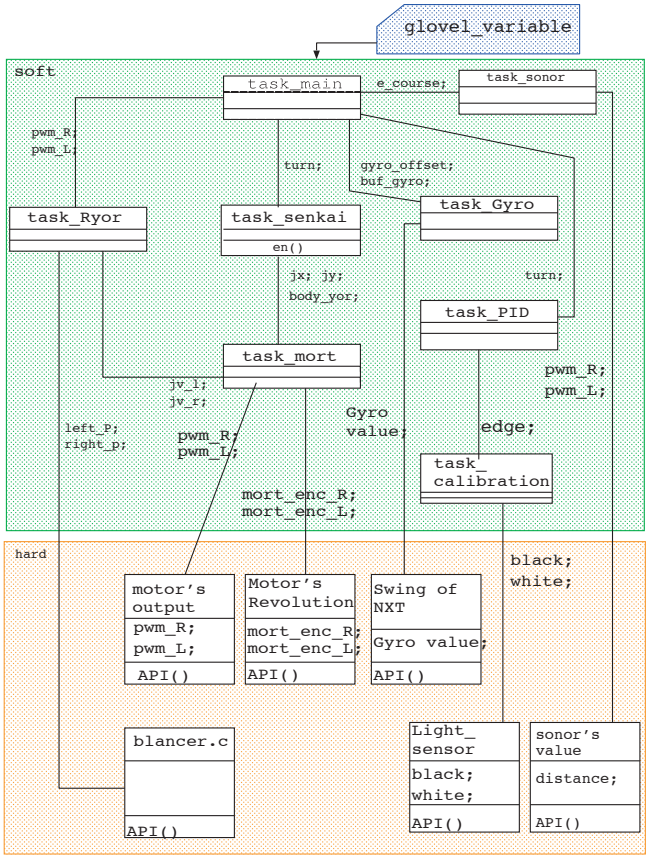
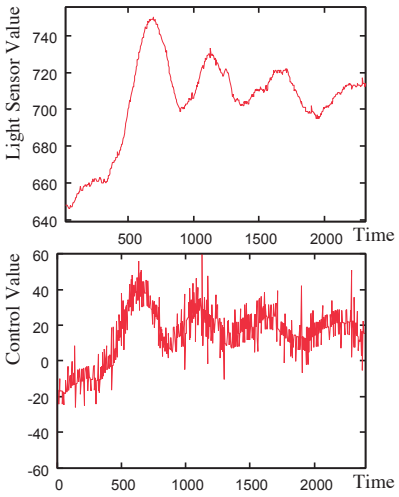


Fig.8　Inverted pendulum control in class diagram

3.3.2　ライントレース

サンプルプログラムの ON-OFF 制御は，光センサから返ってきた値が黒色と白色を測定した値の中間値より小さければ黒方向へ旋回，大きければし白方向へ旋回という制御をする方法である．しかし，この方法では走行体は常に蛇行していることになり，黒線が直線であっても直進することはできず，速い速度でライントレースすることはできない．

この問題に対し，過去の優秀チームのアプローチから，光センサを入力値とし，走行体の旋回量を制御量とし Fig.9 に示す PID 制御 [6] を設計した．その結果，もっとも滑らかに制御され，速い速度でのライントレースを実装することができた．



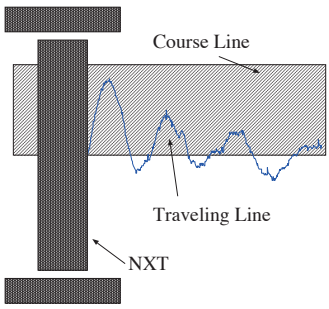


Fig.9　Relation between light sensor value and control value on the PID controls

3.3.3　モータエンコーディング

走行体をラインに頼らない走行をさせたい場合に有効な方法としてモータエンコーディング走行がある．NXT のサーボモータはタイヤの回転角を返すため，タイヤの回転角 ω とタイヤの半径 l から本体速度 v は $v = 2l\pi\frac{\omega}{360^\circ}$ となる．本体速度 v と走行時間の積から走行距離，左右の走行距離の差から旋回度を算出することができる．これらを用いて走行する方法がモータエンコーディング走行である．Fig.10 に旋回度を算出するときの模式図を示す．

最初のアプローチとして現在位置の座標を求め，それを目標座標と比較し，それに応じた制御を行う方法を選んだ．これは走行体の速度を x 軸と y 軸について成分分解し，両軸の移動距離を求めることで現在座標を求める計測部分と，走行体の向きと垂直方向の座標成分を現在座標と目標座標で比較し，この差に応じて左右に旋回する制御部分からなる．Fig.11 に計測部分の計算模式図を示す．

しかし，実際に走行させてみたところ計測部分では，タイヤの空転によって実際のタイヤの回転数とサーボモータが返す回転数に± 2% 程の誤差が生じ，計測部分が正確な座標を算出できなかった．また，目標座標に正確に移動しないと次の処理が行われないため，次の処理に移行できなかった．制御部分では，走行体に慣性がはたらき，10〜15cm 程オーバーランするため制御切り替えが遅くなるという問題が浮上した．

そのため，計測部分は，目標走行距離を算出するようにした．現在座標と目標座標の２点を円周上にもつ円としてとらえ，その円の半径と２点が描く中心角から左右のタイヤの目標走行距離を算出している．目標走行距離を算出するための模式図を Fig.12 に示す．それにより，目標座標に正確に移動しないと次の処理が行われない問題を解決した．タイヤの空転に関しては，試走会とテストコースの誤差が小さかったため，テストコースに合わせた補正値を用いて解決した．制御部分は走行体の移動速度から走行距離を計測し，左右の目標走行距離の差と直進方向の移動量の積から旋回量を算出するように改良した．それにより，オーバーランを 5〜10cm に抑えた．改良前と後のフローチャートを Fig.13 に示す．

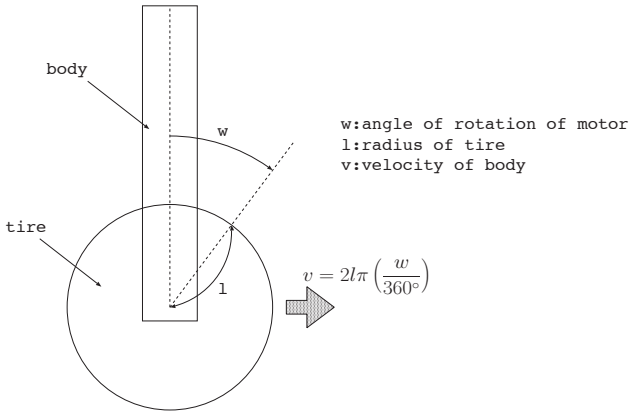


Fig.10　Velocity calculation of NXT

3.3.4　シーソー・階段

アウトコースには難所としてシーソーと階段がある．シーソーは進入方向へ傾いており，階段には二段の段差がある．両者とも通り抜けるためにはそれぞれに細やかな制御が必要である．

この難所は走行体のバランスを保つことが最重要であるため，搭載されたジャイロセンサを用いて段差やシーソーの傾きの検知を試みた．ジャイロセンサは角速度を返す仕様であるため，こ

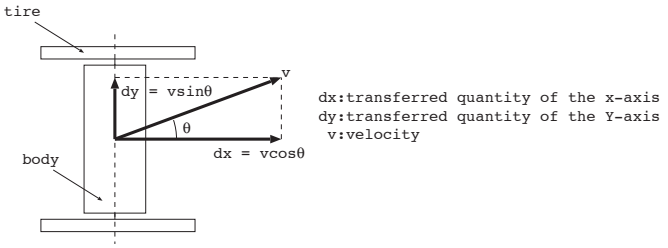


Fig.11　Coordinate calculation of NXT

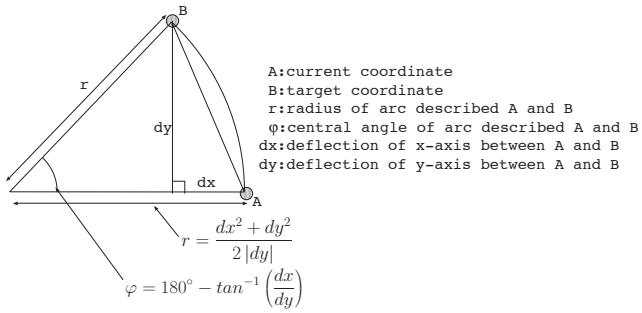


Fig.12　Method of calculating travel distance

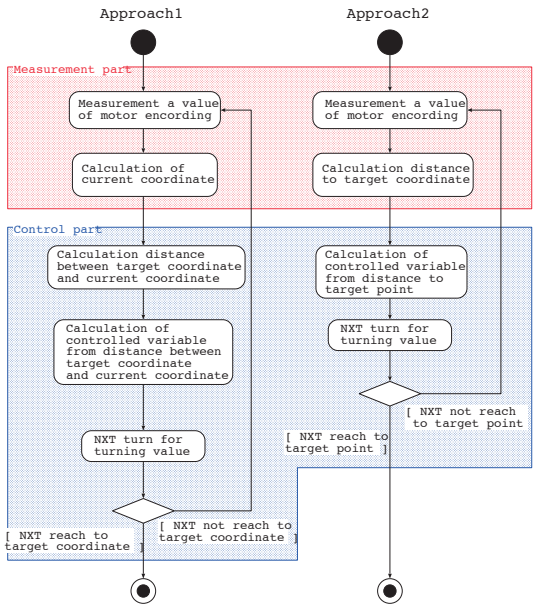


Fig.13　Motor encoding flowchart

れを時間積分して角度を求めた．シーソー初期の傾きは 11 度であり，これをしきい値としてシーソー検知を試みた．

しかし，通常走行中に誤検知が多発した．これは走行体自身が Fig.14 のように倒立振子制御によって常に揺れているため，± 10 度ほどの値が常に返されていることが原因であった．そのため，しきい値を角度から角速度へと変更し，大きく傾いたかどうかを検出する仕様へ変更した．

次に，シーソーを下る際に転倒する問題に着手した．シーソーを下る際は走行体の速度が上昇しすぎていることが原因だと考え，シーソーの板が下り方向に傾いたことを検知すると速度を大幅に落とす制御を行っていたが，カメラにより 1ms 間隔で連続撮影を行い，走行体の本体角度の推移を観察した．これによってシーソーを下る際の走行体は Fig.15 のようにタイヤよりも上部が坂道によって激しく前傾し，それによって転倒しているという動作が判明した．そのため，Fig.16 のように今までとは逆に下る制御では加速するプログラムを実装した．

階段でも基本的にはシーソーと同じ処理をしている．シーソーと同じ方法で段差を検知し，速度を 0 にしている．階段では段差が 2 段続いているため，1 段目でバランスを崩した走行体が 2 段目を迎えてしまうとさらにバランスを崩してしまい転倒してしまう．よって，段差を検知し，速度を 0 にすることを 1 段目，2 段目と最後の下りの合計 3 回繰り返す設計にし，実装を行った．

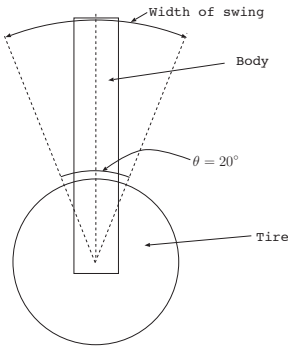


Fig.14　Swing figure of NXT

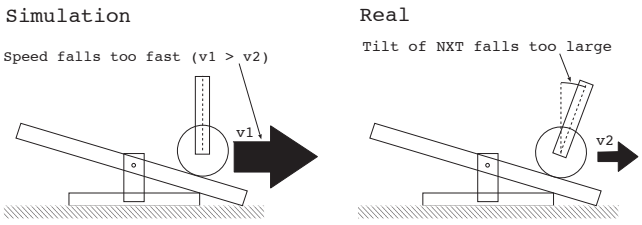


Fig.15　Defference moving NXT between simulation and real

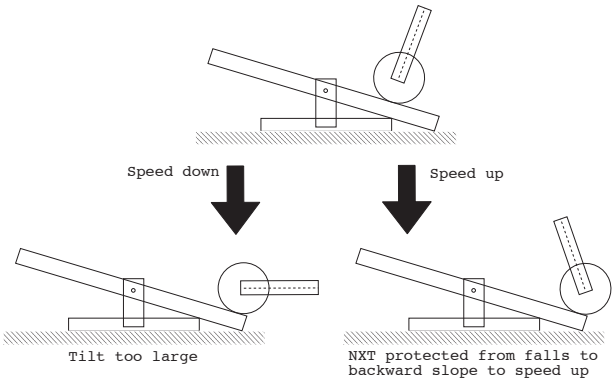


Fig.16　Action when running seesaw of NXT

3.3.5　超音波計測（エニグマデコーディング）

インコースには難所としてエニグマデコーディングがある．これはコース上の特定の位置にある障害物を超音波センサを用いて計測し，そのパターンに応じたコースを走行するというものである．ここで間違ったコースを走行してしまうとリタイアになるため，超音波センサの特性を理解し，いかに障害物の計測率を向上させるかが重要である．また，この難所は超音波計測に加えてモーターエンコーディングが重要な要素となるが，モーターエンコーディングについては前項で詳細を記しているため割愛する．

超音波センサは 50ms 毎に前方に超音波を発射し，障害物に当たって反射してきた超音波を計測することで障害物の有無と距離を算出している．仕様上，距離は cm 単位で 255cm までの計測が可能である．最初のアプローチとして障害物がセンサの正面に位置したときに計測を開始し，障害物の有無を確認する方法を試みた．このとき，モーターエンコーディングを用いた走行を行い，計測を試みたが誤検知の確率が非常に大きく，距離計測も必ずしも正確ではないという問題が判明した．

超音波計測を見直した結果，Fig.17 のようにセンサの正面に対して障害物の面が垂直でなければ超音波が正しく返ってこないことが判明した．また，計測では 10cm から 30cm の距離が最も計測率が高いことが実験で確認されたため，これらの条件を満たす地点で計測を開始することにした．

この方法では，アウトコースを走行している走行体を誤検知してしまう可能性がある．そのため，Fig.18 の中で矢印で示すように計測地点をアウトコースからより遠ざけ，かつ計測方向をコースと平行にして超音波の計測範囲にアウトコースの走行体が入らないよう工夫した．

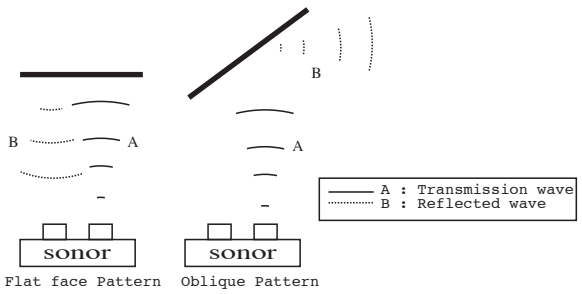


Fig.17　Sonor’s reflection

3.4　ET ロボコン参加結果

大会では，アウトコースはインコース側の走行体に衝突して失格となり，インコースではコースアウトしてリタイアとなった．また，モデル作成ではスケジュールが大幅に狂ってしまったため，十分に情報を盛り込んだモデル図を作成できなかった．

アウトコースでの失敗要因は予期しない状態遷移が発生したためである．走行直前のソースコード調整において人為的ミスが発生したことが原因である．

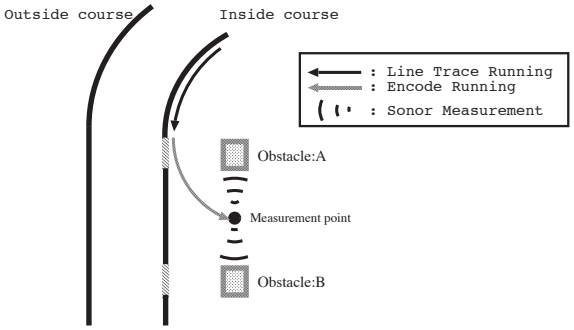


Fig.18　Measurement of barrier

インコースでの失敗要因は目標地点で走行方法を切り替える際にモーターエンコーディングの環境依存により予定しない動作が起こったことである．機能作成の際に補正値を環境依存する値で取ってしまったことが原因である．

これらの失敗はシミュレーションと実際の動作に大きな差があったことが原因である．この事実は ET ロボコンの結果として如実に表れた．この事から，組込み開発にはこのような差異があることを理解しておかねばならない．

4.　PBL 評価

ここで PBL の評価を示す．特に開発結果，学習効果についてを評価する．

4.1　開発結果

各機能で開発を進めた結果，一部の機能においてシミュレーションと実際の動作の差異が大きくなった．そのため何度も機能設計を見直す必要に迫られ，開発期間が大きく延びた．その結果スケジュールにも大きな矛盾が生じてしまい，急ぎスケジュールの見直しを迫られたが，プロジェクトが動いている中での大幅な調整のため，その場しのぎのスケジュールになり，それに新たな矛盾が生じ，また大幅な調整をするというスパイラルに陥った．

しかし，各機能とも大会前日までには完成し，開発を終えた結果，アウトコース走行プログラムではライントレースを主体として走行し，難所をクリアする毎に状態遷移を行い，インコース走行プログラムではライントレースとモーターエンコーディングを適所で切り替えるプログラムが完成した．だが，機能を統一する関係上他の機能との矛盾が生じるために実装を諦めねばならない機能があり，全てを実装することはできなかった．

4.2　学習効果について

プロジェクト開始時点ではメンバーは組込みシステムに関する基礎知識は有していたものの，机上での知識のみであり，実際の開発における手法，ノウハウなどは皆無であった．しかし，結果的にメンバーの知識，技術が開発経過とともに向上していった．このことから，学習と開発の両立はどちらにとっても非常に

有用であるといえる．具体的には，開発を進めるためのアプローチとして PID 制御などの技術を理解するために，自ら知識を取り入れ，その技術の応用力を高めていった．また，プログラムの再利用性を考慮したコーディングも自然に身についた．

ET ロボコンでは昨年とコースが異なり，シーソーやエニグマと言った新たな難所が設けられるため，その難所を克服するために自分たちで新たなアプローチを生み出した．また，機能間での技術交流を積極的に行うことによってどの機能にどんな技術が必要なのかを理解することができたので，全員の学習結果を共有することができた．そしてマネージメントという観点でもお互いの進捗状況が把握できたため，全体を俯瞰的に把握することができた．

モータエンコーディング走行をはじめとして，シミュレーションと実際の動作の差異を体験することができ，実際の組込み開発におけるテスト環境と実際の動作環境が異なることを疑似体験することができた．

4.3 メンバーアンケート

今回実施した PBL についての感想を学部生 7 人にアンケートを取った．それをまとめた図を Fig.19 に示す．この図は各 6 項目において自分が学習前と比べて向上したと思う項目にチェックしてもらったものである．また，図中の数字はチェックをいれたメンバーの人数を表している．

1 では今までの授業などと比べてより深く理解する努力やそのための積極的な取り組みについて
2 では問題を論理的に分析できる能力について
3 では問題を速やかに解決できる能力について
4 では障害となる問題を速やかに発見できる能力について
5 では組込みに関する技術全般の向上について
6 ではメンバーどうしでの意見交換の頻度，内容について
自分の能力が向上したと思っている場合，チェックを入れてもらった．

この図で示されるとおり，2 つの項目であてはまると答えたメンバーが 100 %となり，最も低い項目でも約 70 %のメンバーがあてはまると回答している．総合的に見ると全てのメンバーがこの項目全てに約 90 %弱の割合であてはまると考えているということがわかる．この結果から全てのメンバーがなにかしらの能力が向上したと考えているということが言える．

5. 考察

私たちは PBL を用いて試行錯誤しながら組込みシステムの開発をした．題材として ET ロボコン 2010 を用いることで開発スケジュールや仕様の変更といった実践的な学習を行うことができた．また，実践的な学習だけでなく PID 制御などの技術の根幹知識も身につけることができた．これは私たちが実践に必要な技術を使用する為に根本からの知識が必要であったため，自発的に基礎知識を見直し，必要な知識を身につけた結果である．

このように，PBL を用いることは実践的な知識を身につけるために非常に有用である．その反面自発的な行動が求められるため，個々に差がついてしまう．そのため，技術交流を積極的に行うことができるディスカッションの場を定期的に設けるといったリカバリをすることが重要である．

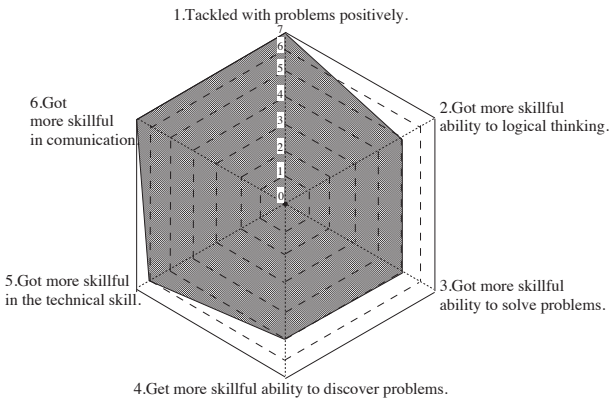


Fig.19 A result of questionnaire

6. まとめ

今回は PBL を用いて組込みシステムの学習を行なった．成果物を作る過程で様々な技術を身につけ，メンバー全員が自分の能力向上を実感している．そのため PBL の効果としては十分なものが得られたといえる．しかし，題材となった ET ロボコンの結果が非常に残念なものとなってしまったため，今回の PBL での成果を生かしてよい結果を得られるように鋭意努力したい．

参考文献

1) ET ロボコン 2010 公式サイト（オンライン），入手先
<http://www.etrobo.jp/2010/>
(参照 2010-06-09)
2) 沢田篤史，小林隆史，金子伸幸，中道上，大久保弘崇，山本晋一郎:飛行船制御を題材としたプロジェクト型ソフトウェア開発実習,
情報処理学会論文誌 (参照 2010-06-09)
3) ソフトウェアの品質と開発効率とは（オンライン），入手先
<http://www.jstage.jst.go.jp/article/secjournal/5/5/275/_pdf/-char/ja/>
(参照 2010-09-30)
4) LEGO MINDSTORMS 公式サイト（オンライン），入手先
<http://www.legoeducation.jp/mindstorms/>
(参照 2010-06-09)
5) nxtOSEK（オンライン），入手先
<http://lejos-osek.sourceforge.net/jp/index.htm>
(参照 2010-06-09)
6) 山本重彦，加藤尚武:PID 制御の基礎と応用 [第 2 版],
朝倉書店 (2005)
7) UML Profile for MARTE: Modeling and Analysis of Real-Time Embedded Systems（オンライン），入手先
<http://www.omg.org/spec/MARTE/1.0>

(参照 2010-06-09)