

栗本直人*1 丹波信夫*2 西ヶ谷浩史*3
奥村稔*4 大平育代*5 鎌田敏之*6 布施泉*7

代表者勤務校：*1 滝高等学校・中学校 *2 半田市立雁宿小学校 *3 静岡大学教育学部附属島田中学校 *4 北海道札幌北高等学校 *5 北海道札幌北稜高等学校 *6 愛知教育大学 *7 北海道大学

1. 要約

本研究では、2年間で「二足歩行ロボット」のキットの開発とそれを活用した小学生・中学生・高校生向けのSTEM (Science Technology Engineering Mathematics) の理解を深める学習カリキュラムの構築を進めた。

「二足歩行ロボット」の制御を学習する中で、小学生・中学生・高校生は、発達段階に応じたSTEMのいろいろな学習ができるようになった。たとえば、生物・物理学的には、「二足歩行」するためには、どのような条件が必要かを学習する。数学的には、「ロボットアーム」による物体の移動で、空間座標の概念を学習する。技術・工学的には、サーボモータのいろいろなノウハウを学習する。

また、小学生・中学生・高校生の発達段階に応じたカリキュラムでは、小学生は楽しくロボットを「WEB制御」し、中学生は楽しくロボットを「ものづくり」し、高校生は楽しくロボットを「プログラミング」した。

なお、学習する時間は、小学生は総合的な学習の時間の中で、中学生は技術家庭科の時間の中で、高校生は情報の時間の中で行った。

共同研究者には、東海スクールネット研究会所属の企業関係者、現場教師・大学関係者が担当した。ロボットの設計・制御基盤の開発から発達段階別カリキュラムの開発までをそのコラボレーションで行った。

2. 研究目的

小学生から高校生までに、STEMを意識させて、発達段階に応じ、体系的に学習するカリキュラムはない。そこで、今回は、「二足歩行ロボット」及びロボットアームの制御を学習テーマとして、発達段階に応じ、体系的な学習カリキュラムを作成し、そのためのロボットキット及びプログラムのプロトタイプを開発して、その評価を行うのが目的である。

3. 仮説

(ア) 2009年度の仮説

- ① 仮説1：二足歩行ロボットのキットを活用して、小学生・中学生が『二足歩行する条件』が何であるかを学習する中で、生物学的あるいは物理学的に効果的な学習環境を提供できる。
- ② 仮説2：ロボットアームのキットを活用して、中学生あるいは高校生が『サーボモータ』あるいは『空間の概念』を学習する中で、技術・工学的あるいは数学的に効果的な学習環境を提供できる。
- ③ 仮説3：二足歩行ロボット及びロボットアームのキットを活用して、高校生あるいは大学生が目で見える形でのプログラミング学習あるいは問題解決型学習を効果的に展開できる。

(イ) 2010年度の仮説

- ① 仮説 4：STEM を理解させるために、小学生・中学生・高校生の発達段階に応じ、体系的なカリキュラムが効果的である。発達段階に応じた体系的なカリキュラムとは、小学生にはWEB制御、中学生にはものづくりと制御、高校生にはプログラミングである。
- ② 仮説 5：小学生も中学生も高校生も、ロボット制御学習を行う際には、PDCA や役割分担・スケジュール等を意識させた Project Based Learning を入れた問題解決型学習をすることが効果的である。

4. 研究計画

(ア) 2009 年度の研究計画<1 年目>

- ① 「二足歩行ロボット」の開発を行う。
 1. 制御基盤の開発、ハードウェアの設計は、東海スクールネット研究会のワーキンググループ及びその関連の企業のメンバーが、現場教師の意見をいれながら担当する。
 2. この制御基盤の設計・製作及び二足方向ロボットのハードウェア（主に、サーボモータとそれをつなぐ金具）の開発費用にこの助成金をあてる。
- ② 「二足歩行ロボット」を制御するプログラムの開発を行う。
 1. プログラムのプロトタイプの開発は、東海スクールネット研究会のワーキンググループ及びその関連の企業のメンバーが、現場教師の意見をいれながら担当する。
 2. このプログラム開発費用にこの助成金をあてる。
- ③ 小学生・中学生・高校生に合わせたカリキュラムのプロトタイプを作成するために、研究協力をいただいたいろいろな小学校・中学校・高等学校で開発中のロボット及びプログラムの試作に対するご意見をいただく。
- ④ 実施可能な体制を模索しながら、カリキュラム内容をチェックして、課題を模索する。
 1. ③～④については、カリキュラム開発を現場教師が行い、プログラム開発者・大学関係者から助言をいただく。
 2. なお、4月に最初に打ち合わせ（最初意思統一）と12月に中間の打ち合わせ（途中での意見調整と次年度への課題調整）を名古屋で行う。

(イ) 2010 年度の研究計画<2 年目>

- ① 1 年目の反省を踏まえて、「二足歩行ロボット」の改良を行う。1 年目の基盤及びハードの修正開発費用にこの助成金をあてる。
- ② 1 年目の反省を踏まえて、小学生、中学生、高校生に合うカリキュラムの改良を行う。1 年目のプログラムの修正開発費用にこの助成金をあてる。
- ③ 2 年間の研究で、「二足歩行ロボット」及びロボットアームのキットとそのカリキュラムを完成させる。（その内容の概略は研究結果に掲載した。その授業資料も参考資料に掲載した。）
 1. 小学生では、「WEB 制御」によるプログラミング学習をする中で、STEM の概念の理解を図る。
 2. 中学生では、「ものづくり」を意識してサーボモータと金具の組み立てを行い、自分で製作したロボットでの制御を行う学習をする中でSTEM の概念の理解を図る。
 3. 高校生では、「プログラミング」を意識して、問題解決型学習の中に、物理学的要素、生物学的要素、数学的要素を入れて、Ruby 言語によるプログラミングを行う。
- ④ この調整をテレビ会議で行いながら、そのまとめを行う。そのまとめは、プログラミング学習を、発達段階に応じ、体系的にまとめた授業内容（「WEB」⇒「ものづくり」⇒「プログラミング」及び「問題解決」のカリキュラム開発を行う。

5. 研究内容

(ア) 2009 年度の研究内容

- ① 作成したキット及びカリキュラム内容

1. 4月～8月に、以下の写真のような11個のサーボを同時にコントロールできる制御基板の試作品（写真1）の開発を行った。



写真1. 11軸制御可能な制御基盤の試作品

2. その上で、9月～10月に、8軸で動く二足歩行ロボット1号機（写真2）、10月～11月に、10軸で動く二足歩行ロボット2号機（写真3）をそれぞれ完成した。また、その間に、二足歩行ロボットのプログラム（言語はRuby）も開発を行った。
3. ここまでの問題点としては、二足歩行ロボット1号機では、サーボモーター（8個）の回転方向をすべて進行方向前後のみにしたため、二足歩行ができない二足歩行ロボットになってしまった。その改良のため、二足歩行ロボット2号機では、足首のところに、進行方向と直角の左右方向にサーボモータを組み込んだ。

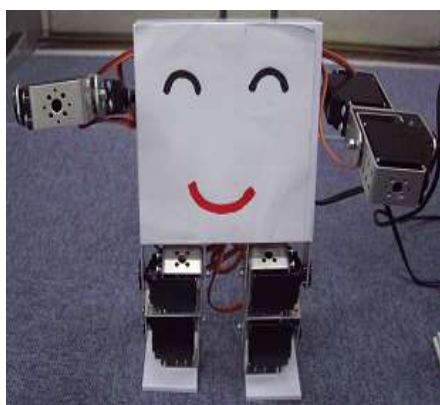
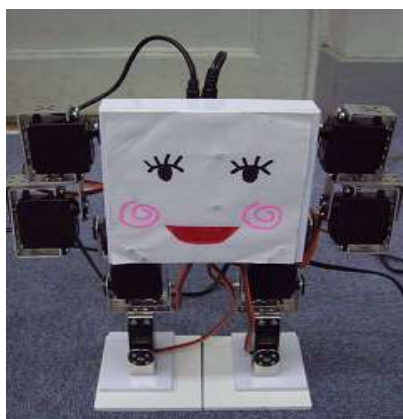


写真2. 二足歩行ロボット1号



足首が違う

写真3. 二足歩行ロボット2号

② 実践した学校とその授業内容

1. 仮説1として、小学生・中学生に『二足歩行する条件』が何であるかを学習する授業を行った。その上で、中学生が自主的に作成した2分間ビデオはなかなかの出来である（写真4及び5）。この授業の試行錯誤により、生物学的あるいは物理学的に『どうしたら、二足歩行できるのか？』をするのかを理解させるのに効果的な教材と学習キットがある程度完成した。なお、授業の実施は、小学校では、愛知県半田市立雁宿小学校、中学校では、愛知県私立滝中学校で実施した。
2. 仮説2として、中学生・高校生が『サーボモータの扱い方』あるいは『空間の概念』を学習する授業を行った（写真6）。具体的には、サーボモータと金具を調整しながらロボットアームへの組み立て作業の中に技術・工学的な概念を、あるいは、ロボットアームによって空間の一点から一転へ物体を移動させる作業の中に数学的な概念を理解させるものであった。この授業の試行錯誤により、技術・工学的に、あるいは、数学的に効果的な教材と学習キットがある程度完成した。なお、授業の実施は、中学校では、帯広市立八千代中学校と愛知県私立滝中学校で

実施した。

3. 仮説3として、高校生・大学生が『二足歩行ロボットあるいはロボットアームのプログラミング制御』を学習する授業を行った(写真7)。具体的には、Rubyプログラミングの結果が目で見え、その中に問題解決型の学習を効果的に展開できた。この授業の試行錯誤により、プログラミング的に、あるいは、問題解決型的に効果的な教材と学習キットがある程度完成した。なお、授業の実施は、高等学校では、愛知県私立滝高等学校と北海道北陵高等学校、大学では、北海道大学で実施した。



写真4. 二足歩行ロボットの1本足立ち



写真5. 二足歩行について議論する生徒



写真6. サーボモータの調整する生徒

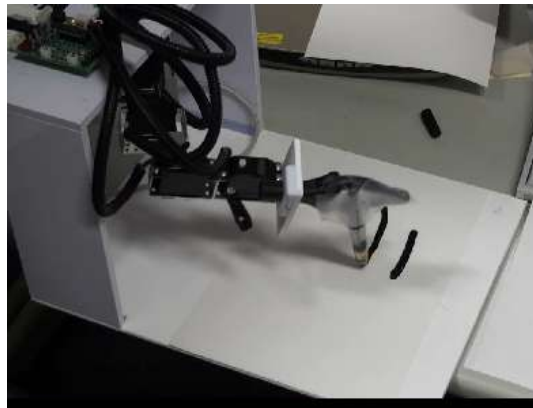


写真7. ロボットアームが習字する様子

③ アンケート内容と結果

1. アンケート内容

(ア) 小学生向けアンケート

- ① 授業の内容は理解できましたか？
 1. 充分できた。
 2. まあまあできた。
 3. あまりできなかった。
 4. ほとんどできなかった。
- ② 授業はおもしろかったですか？
 1. たいへんおもしろかった。
 2. まあまあおもしろかった。
 3. あまりおもしろくなかった。
 4. まったくおもしろくなかった。
- ③ 『二足方向するための条件』が何であるか？理解できましたか？
 1. 充分できた。
 2. まあまあできた。
 3. あまりできなかった。

4. ほとんどできなかった。
- ④ 理科・算数に興味がありましたか？
1. たいへんわいた。
 2. まあまあわいた。
 3. あまりわかなかった。
 4. ほとんどわかなかった。
- (イ) 中学生向けアンケート
- ① 授業の内容は理解できましたか？
1. 充分できた。
 2. まあまあできた。
 3. あまりできなかった。
 4. ほとんどできなかった。
- ② 授業はおもしろかったですか？
1. たいへんおもしろかった。
 2. まあまあおもしろかった。
 3. あまりおもしろくなかった。
 4. まったくおもしろくなかった。
- ③ 『サーボモータの扱い方』あるいは『空間の概念』は理解できましたか？
1. 充分できた。
 2. まあまあできた。
 3. あまりできなかった。
 4. ほとんどできなかった。
- ④ 理科・数学に興味がありましたか？
1. たいへんわいた。
 2. まあまあわいた。
 3. あまりわかなかった。
 4. ほとんどわかなかった。
- (ウ) 高校生向けアンケート
- ① 授業の内容は理解できましたか？
1. 充分できた。
 2. まあまあできた。
 3. あまりできなかった。
 4. ほとんどできなかった。
- ② 授業はおもしろかったですか？
1. たいへんおもしろかった。
 2. まあまあおもしろかった。
 3. あまりおもしろくなかった。
 4. まったくおもしろくなかった。
- ③ 『二足歩行ロボットあるいはロボットアームのプログラミング制御』は理解できましたか？
1. 充分できた。
 2. まあまあできた。
 3. あまりできなかった。
 4. ほとんどできなかった。
- ④ 理科・数学に興味がありましたか？
1. たいへんわいた。
 2. まあまあわいた。
 3. あまりわかなかった。
 4. ほとんどわかなかった。

2. アンケート結果

(ア) 小学生向けアンケート結果 (表 1)

小学生 (25名)	1	2	3	4
授業理解	充分	まあまあ	あまりない	ほとんどない
人数	10	7	5	0
%	45.5	31.8	22.7	0
授業面白い	たいへん	まあまあ	あまりない	ほとんどない
人数	11	6	5	0
%	50	27.3	22.7	0
二足歩行の条件理解	充分	まあまあ	あまりない	ほとんどない
人数	8	8	6	0
%	36.4	36.4	27.2	0
理科算数への興味	たいへん	まあまあ	あまりない	ほとんどない
人数	9	5	5	3
%	40.9	22.7	22.7	13.7

(イ) 中学生向けアンケート結果 (表 2)

中学生 (51名)	1	2	3	4
授業理解	充分	まあまあ	あまりない	ほとんどない
人数	15	16	13	7
%	29.4	31.4	25.5	13.7
授業面白い	たいへん	まあまあ	あまりない	ほとんどない
人数	17	20	8	6
%	33.3	39.2	15.7	11.8
サーボ・空間理解	充分	まあまあ	あまりない	ほとんどない
人数	12	19	15	5
%	23.5	37.3	29.4	9.8
理科数学への興味	たいへん	まあまあ	あまりない	ほとんどない
人数	18	21	9	3
%	35.3	41.2	17.6	5.9

(ウ) 高校生向けアンケート結果 (表 3)

高校生 (85名)	1	2	3	4
授業理解	充分	まあまあ	あまりない	ほとんどない
人数	20	39	21	5
%	23.5	45.9	24.7	5.9
授業面白い	たいへん	まあまあ	あまりない	ほとんどない
人数	27	38	15	5

	%	31.8	44.7	17.6	5.9
プログラミング理解		充分	まあまあ	あまりない	ほとんどない
	人数	26	37	17	5
	%	30.6	43.5	20	5.9
理科学への興味		たいへん	まあまあ	あまりない	ほとんどない
	人数	26	38	16	5
	%	30.6	44.7	18.8	5.9

小学生では、『二足歩行の条件』はほぼ（73%）理解された。
 中学生でも、『サーボモータの扱い方・空間の概念』はほぼ（61%）理解された。
 高校生では、『プログラミング』はほぼ（74%）理解された。
 以上のアンケートの結果から各段階において仮説1～3はほぼ証明された。

(イ) 2010年度の研究内容

① 作成したカリキュラム内容

- 2009年度の結果をもとに、表4のようなカリキュラム内容を構築した。発達段階、学習時間、学習内容、主な授業案からなる。

表 4

発達段階	学習時間	学習内容	主な授業案
小学生	総合的な学習	WEB から制御	① 歩く条件を考えてみよう。 ② ムーンウォークするロボットはなぜだろうか？ ③ 一本足直立をさせてみよう。 ④ 二足歩行をさせてみよう。
中学生	技術	組み立てと制御	① サーボモータセンター出しをしてみよう ② サーボモータの組み立て、制御基盤とサーボモータの動作確認をしよう。 ③ ロボットアームで物体を移動させてみよう。 ④ 二足歩行をさせてみよう。
高校生	情報	プログラミングでの制御	① 二足歩行（Ruby のプログラミング）させてみよう。 ② ロボットアームで物体を移動させてみよう。 ③ ロボットアームに木琴の演奏をさせよう。 ④ ロボットアームに習字をさせよう。

② 実践した学校とその授業内容

- 仮説4（2010年度）の証明をするために、小学生・中学生・高校生の発達段階に応じたカリキュラム（表4）を協力いただいた小学校・中学校・高等学校で実践した。その中で児童・生徒は、気づかないうちにSTEMを理解しながら、小学生にはWEB制御、中学生にはものづくりと制御、高校生にはプログラミングを楽しく学習できるような改良（後述）を加えた。
- 仮説5（2010年度）の証明をするために、小学生にも中学生にも高校生にも、この授業を行う際には、グループで作業する中にPDCAや役割分担・スケジュール等を意識さ、Project Based Learningの概念も取り入れた。すべての授業は表4のとおり問題解決型である。

3. 具体的には、小学生・中学生が、iPod (WEB) からでもロボット制御できる以下のシステム (図1) を作成した。図2は iPod の制御画面である。

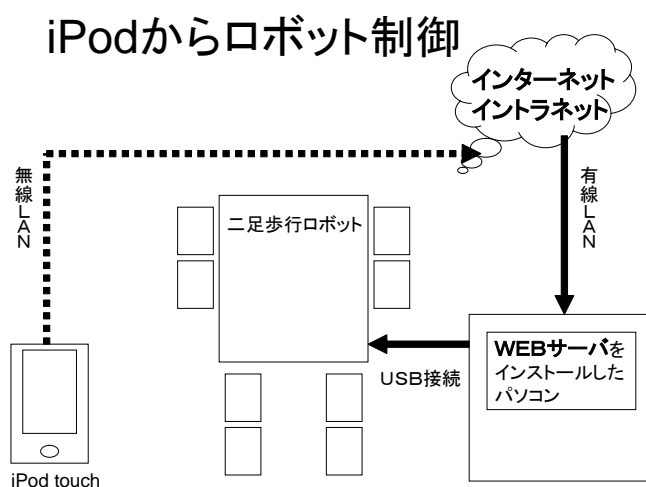


図1. iPod からロボット制御



図2. iPod からの制御画面

4. 中学生が、「ものづくり」として、①サーボモータのセンター出しをしてみよう。②サーボモータを組み立てて動作確認をしよう。③ロボットアームで物体を移動させてみよう。(具的な内容は、参考資料に提示してある。) のカリキュラムをグループ学習として、取り組んだ。
5. 高校生が、「プログラミング」として、①二足歩行 (Ruby のプログラミング) をさせてみよう。②ロボットアームで物体を移動させてみよう。③ロボットアームに木琴の演奏をさせよう。④ロボットアームに習字をさせよう。 のカリキュラムもグループ学習として、取り組んだ。

③ アンケート内容と結果

1. アンケートの内容は 2009 年度と同じで、結果は以下のとおりである。

(ア) 小学生向けアンケート結果 (表5)

小学生 (25 名)	1	2	3	4
授業理解	充分	まあまあ	あまりない	ほとんどない
人数	15	8	2	0
%	60	32	8	0
授業面白い	たいへん	まあまあ	あまりない	ほとんどない
人数	19	5	1	0
%	76	20	4	0
二足歩行の条件理解	充分	まあまあ	あまりない	ほとんどない
人数	10	12	2	1
%	40	48	8	4
理科算数への興味	たいへん	まあまあ	あまりない	ほとんどない
人数	8	12	4	1
%	32	48	16	4

(イ) 中学生向けアンケート結果 (表 6)

中学生 (53名)	1	2	3	4
授業理解	充分	まあまあ	あまりない	ほとんどない
人数	15	27	9	2
%	28	51	17	4
授業面白い	たいへん	まあまあ	あまりない	ほとんどない
人数	21	28	4	0
%	40	53	7	0
サーボ・空間理解	充分	まあまあ	あまりない	ほとんどない
人数	13	29	9	2
%	25	55	16	4
理科数学への興味	たいへん	まあまあ	あまりない	ほとんどない
人数	21	29	3	0
%	40	55	5	0

(ウ) 高校生向けアンケート結果 (表 7)

高校生 (84名)	1	2	3	4
授業理解	充分	まあまあ	あまりない	ほとんどない
人数	22	45	17	0
%	26	54	20	0
授業面白い	たいへん	まあまあ	あまりない	ほとんどない
人数	35	42	7	0
%	42	50	8	0
プログラミング理解	充分	まあまあ	あまりない	ほとんどない
人数	23	44	16	1
%	27	52	20	1
理科数学への興味	たいへん	まあまあ	あまりない	ほとんどない
人数	36	39	9	0
%	43	46	11	0

小学生では、『二足歩行の条件』は (73%⇒88%) 理解された。
 中学生でも、『サーボモータの扱い方・空間の概念』は (61%⇒80%) 理解された。
 高校生では、『プログラミング』はほぼ (74%⇒79%) 理解された。
 以上のアンケートの結果から各段階において2010年度も仮説1~3は証明された。
 また、各段階・各テーマとも改善が見られ、2010年度の仮説4・5も証明された。

(ウ) 2年間のまとめ

- ① 大まかにいえば、1年目で、制御基盤の開発、二足方向ロボットキット及びロボットアームキットの開発を終え、2年目で、それを活用するSTEMを意識したカリキュラムの整備が行われた。アンケート結果から見ても、それは一目瞭然である。
- ② さらに、各段階で同じ結果が出たが、グループ学習の中にPDCAや役割分担・スケジュー

ール等を意識させ、なおかつ問題解決型の学習を行うことは、児童生徒の学習には非常に効果があるものと思われる。

6. 研究結果及び成果

- (ア) 2年間の研究で、「二足歩行ロボット」とロボットアームのキットとそのカリキュラムを完成させることができた。
- (イ) 小学生では、二足歩行ロボットのハードウェア（10軸制御）とそれを簡易にしたロボットアームのハードウェア（5軸制御）とそれを制御する「WEB」ベースのソフトウェアのキットとして開発した。それにより、二足歩行するための重心移動のための一本足立ち（物理的な素養）の課題、ロボットアームによる空間の一点から一点への物体の移動（数学的・プログラミング的な素養）などの課題解決型の学習カリキュラムが開発された。また、二足歩行できるロボットと二足歩行できないロボットを見せることで、歩行するという条件（生物的な素養）を理解させる問題解決型の学習カリキュラムが開発された。
- (ウ) 中学生では、より「ものづくり」にこだわったカリキュラム開発を行った。①サーボモータのセンター出し、②サーボモータと接続金具の組み立て、③二足歩行ロボット及びロボットアームの組み立てと動作確認などでロボットを自作するカリキュラム開発を行うことができた。
- (エ) 高校生では、周りの環境とハード部分の課題の「問題解決」を物理学的、生物学的、数学的な学習概念を入れた Ruby 言語による本格的な『プログラミング』をしながら細かい動きを表現するカリキュラム開発を行った。
- (オ) プログラミング学習を発達段階別の視点を入れながら、かつ発達段階別の授業形態・テーマも「WEB」⇒「ものづくり」⇒「問題解決」と変えていくカリキュラム開発を行うことができた。これを行うことで、小学生から高校生まで学習の中に、以下の点に関する効果が見られた。
- ① 理科・数学への興味関心を喚起した。
 - ② ロボットへの興味をいだかせ、科学技術教育への興味関心を喚起した。
 - ③ ロボット制御の中で、プログラミング教育への興味関心を喚起した。
 - ④ 以上、今回の研究開発を行うことで、情報教育から数学理科教育への手助けあるいは橋渡しを行うことができたと思われる。

7. 研究課題

以下のご意見をいろいろなところからいただいた。それをまとめた問題点・課題を以下に列挙する。

- (ア) ロボットアームも二足歩行ロボットも高価なため、普及させるには、サーボモータ・金具・制御基板などを大量生産して原価を下げる必要がある。そのため、一般の教材とするにはなかなか難しい。
- (イ) ロボットアームも二足歩行ロボットも組み立て・調整を教師が行う必要がある。そのため、サーボモータ・金具などの組み換えが、容易である必要がある。また、システムの拡張や改造に業者さんが関わるので、その部分にお金がかかってしまう。
- (ウ) 現在のロボットアームにも二足歩行ロボットにも、センサ機能がないのでフィードバック制御ができない。この部分に関するシステムの改良・改善は早急にすすめるべきである。
- (エ) WEB 制御あるいはプログラミングについて、問題解決型学習と言いながら、プログラムのコードを修正するだけで終わってしまう可能性がある。つまり、いじって試してなんとなくうまくいったから終わりになりかねないので、その部分の改善が必要である。
- (オ) 「応用性、拡張性、期待される効果などには極めて優れており、教材としても非常にシンプルに完成しているが、使う側のアイデア・指導力次第で結果が左右される極めてシビアな教材だ」というご意見もいただきました。どのように改善するべきか今後の

重要な検討課題と思われる。

8. 参考資料

- (ア) サーボモータの調整・組み立て・プログラム・操作のカリキュラム
- ① 中学校用カリキュラム（報告書の最後に掲載）
 - ② 高校生用カリキュラム（報告書の最後に掲載）
- (イ) 授業及び授業協力した学校の一覧（2009年度）
- ① 小学校・・・愛知県半田市立雁宿小学校
 - ② 中学校・・・北海道帯広市立八千代中学校、愛知県私立滝中学校
 - ③ 高等学校・・・北海道札幌北稜高等学校、愛知県私立滝高等学校
 - ④ 大学・・・北海道大学、高麗大学
- (ウ) 授業及びデモンストレーションした学校の一覧（2010年度）
- ① 小学校・・・愛知県半田市立雁宿小学校、愛知県半田市立板山小学校
 - ② 中学校・・・北海道帯広市立緑園中学校、愛知県私立滝中学校
 - ③ 高等学校・・・北海道札幌北稜高等学校、北海道札幌北高等学校、愛知県私立滝高等学校
 - ④ 大学・・・北海道大学

9. 謝辞

今回の2年間の研究には、日本全国あるいは韓国の高麗大学の李先生はじめ、いろいろな方々からの献身的なご支援とご協力をいただいた。この場をお借りしてお礼申し上げます。たいへん、お世話になりました。ありがとうございました。

① 中学生用カリキュラム

School-Robo



サーボモーターを制御してみよう
～ センター出し編 ～

制作・著者: School-ROBO

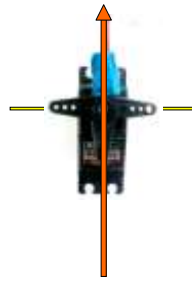
●サーボモーターのセンター出しをする理由(1)



- ・RCサーボは、ラジコン用のモーターとして利用されています。普通のモーターと違い、グルグル回転することはできず、回転は左右180度の範囲です。
- ・RCサーボに電流を流すとどうなるのでしょうか？
- ・答えは動きません。指でつまんで自由に動かすこともできます。
- ・つまり、どの位置にあるのか不定な状態です。これでは、電源を入れた直後のロボットの状態が分からず、プログラムでどこから制御して良いか分かりません。

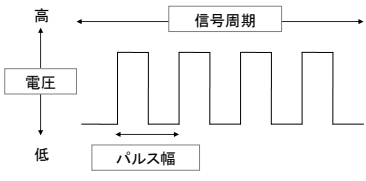
真ん中を90度とし、左右180度の範囲で可動

●サーボモーターのセンター出しをする理由(2)



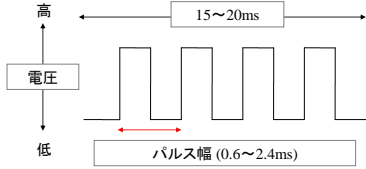
- ・そのため、電源を入れた直後の状態を決めるために初期位置(原点)として本書では、RCサーボをセンターの位置にすることにします。
※必ずセンターである必要はありません。ロボットに合わせて初期位置を設計します。
- ・では、どのようにしてセンターの位置にすれば良いのでしょうか？
- ・RCサーボを見るとケーブルが3本出ています。電源(+)2本の外、残り1本は位置を決めるための信号線として使います。この信号線にパルスを送ることで位置が決まります。

●サーボモーターを動作させるには(1)



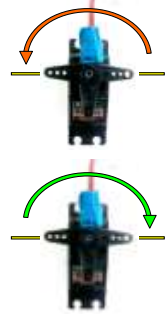
- ・パルスは、一定時間(信号周期)の中で、電圧の高低を何回繰り返すか、というものです。
- ・今回使うRCサーボは、信号周期が15～20ms(ミリ秒)、ニュートラル(センター)パルスは1.5ms、パルス範囲は1.5ms±0.9msとなっています。

●サーボモーターを動作させるには(2)



- ・つまり、最低パルス幅は
 $1.5\text{ms} - 0.9\text{ms} = 0.6\text{ms}$
- 最大パルス幅は
 $1.5\text{ms} + 0.9\text{ms} = 2.4\text{ms}$
- となり、0度の位置にするには「0.6ms」、センターの90度は「1.5ms」、180度の位置は「2.4ms」すれば良いことが分かります。

●サーボモーターを動作させるには(3)



- ・パルス幅が0.6～1.5ms(0～90度)の時は、RCサーボが左側へ動きます。
- ・パルス幅が1.5～2.4ms(90～180度)の時は、RCサーボが右側へ動きます。

●目的にあったパルス幅を作る(1)

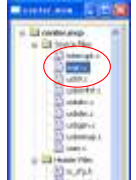
- ・プログラムでRCサーボをセンター(90度)にするためのパルスを作成します。
- 事前に、PICの開発元であるMicrochip社より、以下のソフトウェアがインストールされ、動作可能な状態である必要があります。
 - ⇒ MPLAB IDE v8.xx
 - ・バージョンは8系を使います。
 - ・ソースコードのコンパイルに必要です。
 - ⇒ Microchip C18 (MCC18)
 - ・Cソースのコンパイルに必要です。
 - ⇒ PICDEM FS USB Demo Tool (MCHPFSUSB v1.2)
 - ・コンパイルして生成されたHEXファイルをPICに書き込むのに必要です。

(c) School Robo 7

●目的にあったパルス幅を作る(2)

作業するパソコンに「C:\pic\center」が存在することを確認して下さい。存在しない場合は、担当者に申し出て下さい。

MPLAB IDEを起動し、[Project → Open]を選択後、「C:\pic\center\center.mcp」を選択してプロジェクトを読み込みます。

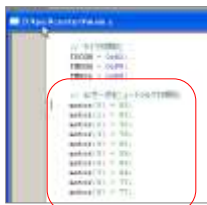


プロジェクトウインドウから、「main.c」をダブルクリックしてソースコードを開きます。

(c) School Robo 8

● 目的にあったパルス幅を作る(3)

「static void InitializeSystem(void)」関数(220行目～)まで画面をスクロールさせ、234行目から参照します。



配列変数motor[0]～motor[9]に指定している数値を変更します。
数値は度数になります。
motor[0] = 90;
で1番目のRCモーターが90度になります。[0]～[9]により、同時に10個のRCサーボを制御できます。

※RCモーター毎の誤差を考え、数値は限界値を覗いた1～179の範囲で指定して下さい。

● 目的にあったパルス幅を作る(4)



左図の赤丸のアイコンをクリックし、コンパイルをします。これにより、「center.hex」ファイルが作成されます。

PCとPIC基板をUSBケーブルで接続し、ブートローダーモード(BOOTスイッチを押しながら、RESETボタン押下)にします。次に、PICDEM FS USB Demo Toolを起動して、HEXファイルを書き込みします。

● 目的にあったパルス幅を作る(5)



「PICDEM FS USB0(Boot)」を選択し、[Load HEX File]ボタンを押下します。

「C:\pic¥center¥center.hex」を選択し、[開く]ボタンを押下し、続いて[Program Device]を押下すると、PICにプログラムを書き込みます。

● 目的にあったパルス幅を作る(6)

モーターを取り付けるピンの配列は以下のようになっていますので、プログラムしたモーター番号に合わせて接続します。



※モーター11は未対応。

● 角度を指定して使い易いようにする

PIC基板からUSBケーブルを外し、PIC基板にACアダプターを接続して電源を供給します。

いかがでしょうか？RCサーボが音をたてて瞬時にセンターの位置へ移動したでしょうか？

もし動かなかった、分かりにくかった場合は、一度ACアダプターを抜き、RCサーボの角度を手でゆっくりとセンターからズラしてから、ACアダプターを接続してみてください。

時間がある方は、motor[0]～ motor[9] の値を変えて色々試してみてください。

RCサーボを複数組み合わせ、ロボット等を製作すると、初期位置が90度では使いにくい場合があります。この場合は何度も様子を見て調整をしながら使いやすい初期角度を見つけて下さい。

School-Robo²



サーボモーターを制御してみよう
～ ロボットアーム組み立て編 ～

制作・著者：School-ROBO

● ロボットアームの組み立て概要



- ・写真は組み立て完成の状態です。
- ・5個のRCサーボを利用しています。
- ・腕の付け根から第1サーボ、第2サーボ・・・と数えることにします。
- ・電源を入れた時の初期状態として、腕がLの字型になるように組み立てます。
- ・各サーボの動作は以下の通りです。

- ① 腕を左右に振ります
- ② 腕を上下に振ります
- ③ 腕を左右に振ります
- ④ 腕を上下に振ります
- ⑤ 物を細んだり放したりします

(c) School Robo

15

● RCサーボの接続



- ・RCサーボの接続は、軸受けや金具、ネジを使って行います。
- ・RCサーボは、前回行った「センター出し」を行った状態で金具に取り付けます。必要であれば、センター出しの信号を流しながら行います。
- ・**ネジを締め過ぎないようにして下さい。ネジ山が壊れて軸受けや金具が使えなくなります。金具はモーターと同じくらい高価なものです。**
- ・新品の軸受けや金具はネジがスムーズに入らない場合があります。ネジを真っ直ぐにして慎重に取り付けて下さい。

(c) School Robo

16

● 台に第1サーボを取り付ける(1)



・台に第1サーボを取り付けます。写真のように取り付けて下さい。上下の位置にも注意します。



・第1サーボに取り付ける金具と軸受け、ネジを用意します。

● 台に第1サーボを取り付ける(2)



・軸受けにネジをはめ込みます。



・軸受けを金具に取り付けて下さい。※ネジはきつく締めないで下さい。

⇒ネジの締め方の例をP.8に記載しています。

● 台に第1サーボを取り付ける(3)



・第1サーボを基板に接続し、基板の電源を入れ、センター出しをします。
※ここで、5個のRCサーボのセンター出しを済ませても結構です。



・写真のように金具をサーボの軸に取り付け、指で押さえながら真ん中のネジを締めます。この時、**サーボに負荷が掛からないように、ネジを締め過ぎないように注意します。**

● 第2サーボを組み立てる(1)



・写真のようにパーツが揃っているか確認をします。
※以後のサーボも同じ手順で組み立てますので参考にして下さい。

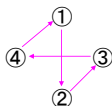


・金具に軸ネジを取り付けます。写真では白いワッシャをはめ込んでいますが、外れやすいので他のサーボを取り付ける直前ではめ込んで結構です。

● 第2サーボを組み立てる(2)



・金具に軸受けを取り付けます。ネジ4本は初めは軽く締め、位置に問題が無ければネジが回らなくなるまで締めます。但し、きつく締めないように注意して下さい。



左図のように、向かい合ったネジを少しずつ締めて行きます。

● 第3サーボを組み立てる(1)



・写真のように2個の金具を組み合わせさせてネジで取り付けます。



・金具に第3サーボを取り付けます。

● 第3サーボを組み立てる(2)



・第3サーボのセンター出しをするためにサーボの線を基板に接続します。※右から3番目のピン。



・第2サーボ用金具を、写真のように第3サーボの軸に取り付けます。※サーボに負荷を掛けないように指で押さえながら押し込んで取り付けて下さい。

● 第3サーボを組み立てる(3)



・第3サーボの軸をネジで締めます。



・写真のように第2サーボを取り付けます。

● 第2サーボを取り付ける(1)



・第1サーボの金具に第2サーボを写真のように取り付けます。軸ネジ側(写真向かって左)から、はめ込みします。

※第1サーボに電気を流し、センター出ししながらする場合は、サーボに負荷がかからないように注意して下さい。

・第2サーボの軸受けを取り付けします。写真のように少し前になるように取り付けします。(重さで下がる分を考慮し、少し前に出しておく。)

● 第2サーボを取り付ける(2)



・取り付けが完了した状態です。各サーボの位置に問題が無いか確認して下さい。少々ズレは後のプログラム修正で対応します。

● 第4～5サーボを組み立てる(1)



・今までの要領で写真のように腕の残りの部分を組み立てます。

● 第4～5サーボを取り付ける(1)



・組み立てた第4～5サーボのユニットを第3サーボの金具に取り付けします。写真のように少し上に向くように取り付けします。(重さで下がる分を考慮し、少し上に向けておく。)

・写真のよう軸受けを第4サーボの軸にはめ込んでネジ止めします。

※第4サーボに電気を流し、センター出ししながらする場合は、サーボに負荷がかからないように注意して下さい。

● 第4～5サーボを取り付ける(2)



・第5サーボの部分は写真のように、サーボをセンターにした時に手の部分が少し開いた状態になるように調整します。

・基板の電源が入った状態でL字型になっていること、電源を切って手で上下左右に動かして特に異常が無ければ組み立て完了です。

② 高等学校用カリキュラム

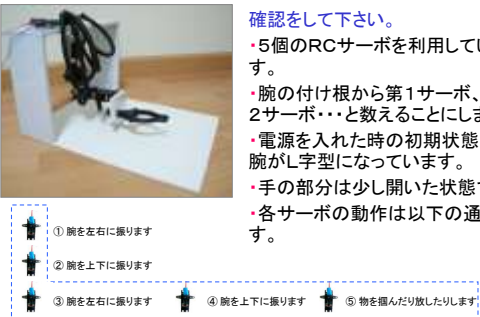


School-Robo

サーボモーターを制御してみよう
～ ロボットアーム操作編 ～

制作・著者: School-ROBO

● ロボットアームの各サーボの動作について

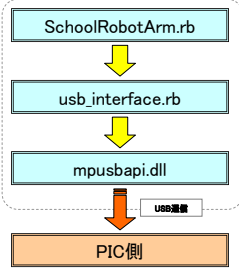


確認して下さい。

- 5個のRCサーボを利用しています。
- 腕の付け根から第1サーボ、第2サーボ・・・と数えることにします。
- 電源を入れた時の初期状態は、腕がし字型になっています。
- 手の部分は少し開いた状態です。
- 各サーボの動作は以下の通りです。


① 腕を左右に振ります
② 腕を上下に振ります
③ 腕を左右に振ります
④ 腕を上下に振ります
⑤ 物を掴んだり放したりします

● プログラムについて




- 「SchoolRobotArm.rb」と「usb_interface.rb」の2つを使います。言語はRubyで記述しています。
- 「SchoolRobotArm.rb」はメインプログラムです。今回はこちらを編集して、ロボットアームの操作をします。
- 「usb_interface.rb」はUSB通信をするためのプログラムです。利用のみで編集はしません。
- 「mpusbapi.dll」はMicrochip社が用意しているWindows用USB通信ドライバーです。

● プログラムの実行準備と動作確認



- エクスプローラでCドライブの直下にディレクトリ「robo」を作成して下さい。
- ディレクトリ「robo」に「SchoolRobotArm.rb」と「usb_interface.rb」を配置します。

● プログラムの実行準備と動作確認(2)



- コマンドプロンプトを起動して、ディレクトリ「robo」へ移動します。
- `>c:`
- `>cd %robo`
- 「SchoolRobotArm.rb」を引数「demo」と「100」をつけて実行します。
- `>ruby SchoolRobotArm.rb demo 100`
- ロボットアームが左右に動き出したら動作は正常です。
- Ctrlキーを押しながらCキーを押下してプログラムを停止します。

● プログラムの実行準備と動作確認(3)

「SchoolRobotArm.rb」は、3つの起動モードがあります。

① 5つの各RCサーボ(モーター)を任意の位置に移動

＜実行方法＞

`ruby SchoolRobotArm.rb MT1 MT2 MT3 MT4 MT5`

MT1～MT5は全て指定して下さい。省略はできません。

MT1 は腕の付け根から数えて1番目のモーターになり、MT5 は手の開閉用モーターになります。

MTnの値は角度となり、1～180の範囲で、電源ON直後は全てのモーターが 90になっています。

まずは、MT1～MT5を90から少しずつ変化させて動かせる範囲などを確認して下さい。

● プログラムの実行準備と動作確認(4)

② デモモード

＜実行方法＞

`ruby SchoolRobotArm.rb demo`

または

`ruby SchoolRobotArm.rb demo STEP`

デモで動作します。

向かって左に置いたブロックを掴んで右に移動、それをまた掴んで左に戻す動作を繰り返します。

プログラムの変更で、Hの字動作をさせることもできます。

※本プログラムの「デモモード用」move_testメソッドをご確認下さい。

STEPは、移動ステップ数になります。指定なし時は50になります。

小さくすると動きが荒くなり、移動も高速になります。

大きくすると、動きが細かくなりますが、移動も低速になります。

● プログラムの実行準備と動作確認(5)

③ 鉄琴を叩く動作

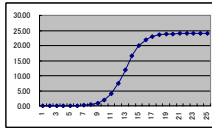
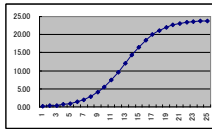
＜実行方法＞

`ruby SchoolRobotArm.rb`

鉄琴を叩く動作をします。

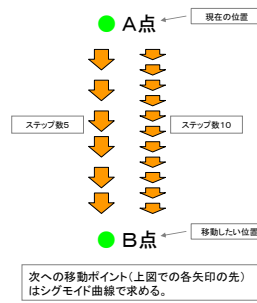
「もりびとこぞりて」の演奏を繰り返します。

●RCサーボの制御について(1)



- RCサーボは初めはゆっくり動かし始め、途中を速く、目的の場所に近くなったらゆっくり移動するようにしています。
- これを実現するにはS字カーブを描く必要があるため、シグモイド曲線を使っています。この処理をしている関数名は motor_move (255行目～)です。

●RCサーボの制御について(2)



- シグモイド曲線を利用することで、A点からB点への移動量が大きくても滑らかに動くようにしています。
- 更に滑らかさを決めるために「ステップ数」を指定する必要があります。値が小さければ素早くこちなく、大きければ動作はゆっくりですが、滑らかに動作します。これを使い分け、移動させたいおまかな点を指定してプログラムすることになります。

●プログラム編集のポイント(1)

```
#
# 自由編集用
#
def arm_personal
  # 2D座標値の値から、2D座標の距離へ移動
  x = 0
  y = 0
  size = nil
  radius = [90, 90, 90, 90, 90]
```

# 腕各八の字に動かす				
idou1 =	110	40	20	90
idou2 =	170	130	161	90
idou3 =	90	90	90	90
idou4 =	140	140	110	90
idou5 =	70	105	70	90

- 赤枠の単位で指定します。数値は角度です。
- 最初の各RCサーボ5個の位置は①、次は②、その次は③、となります。

- 「SchoolRobotArm.rb」をテキストエディタで開いて下さい。
- 今回編集する場所は387行目からの「自由編集用」です。
- 397行目からの配列idou1～idou5を編集します。
idou1 = RCサーボ1
idou2 = RCサーボ2
idou3 = RCサーボ3
idou4 = RCサーボ4
idou5 = RCサーボ5

●プログラム編集のポイント(2)

```
@Mrobo>ruby SchoolRobotArm.rb personal 100
```

```
@Mrobo>ruby SchoolRobotArm.rb 90 90 90 90 90
@Mrobo>ruby SchoolRobotArm.rb 95 90 90 90 90
@Mrobo>ruby SchoolRobotArm.rb 95 95 90 90 90
@Mrobo>ruby SchoolRobotArm.rb 95 95 95 90 90
```

【数値と動作の関係】
※観み方により動作が異なる場合があります。
MT1: 小さいと向かって右、大きいと向かって左に動く。
MT2: 小さいと下、大きいと上に動く。
MT3: 小さいと向かって右、大きいと向かって左に動く。
MT4: 小さいと上、大きいと下に動く。
MT5: 小さいと開く、大きいと閉く。

- 実行するには、「SchoolRobotArm.rb」に引数「personal」と「100」をつけて実行します。
➤ ruby SchoolRobotArm.rb personal 100
- まずは、どのように動かすかイメージして、RCサーボを任意の位置に移動するモードを使って実際に位置を確認しながら値を決めていって下さい。
このモードはシグモイド曲線を使っていないため、数値が前回と比べ大きく変化しないように(±5程度で)指定して下さい。