

無重力空間におけるロボットの姿勢制御・移動制御

大城, 貴仁 / OSHIRO, Takahito

(発行年 / Year)

2009-03-24

(学位授与年月日 / Date of Granted)

2009-03-24

(学位名 / Degree Name)

修士(工学)

(学位授与機関 / Degree Grantor)

法政大学 (Hosei University)

修士論文

無重力空間におけるロボットの
姿勢制御・移動制御

2008 年度

大学院工学研究科
システム工学専攻修士課程

07R6109

おおしろ たかひと
大城 貴仁

指導教授 渡辺嘉二郎 教授

Master's Thesis

ATTITUDE CONTROL AND MOVEMENT CONTROL OF THE ROBOT IN ZERO GRAVITY SPACE

Takahito OSHIRO

Abstract

In Japan, the fight competition by using biped locomotion robot is often held under the auspices of ROBO-ONE committee which aims to make progress in robot technology. The committee is also designing that they operate the robot in space from the earth. And they have desired solutions to attitude control, strength, and the attachable method of the robot to the rocket. First of all, this paper describes attitude control of biped locomotion robot in space. The author proposes the robot is controlled by I-PD control gets the stability of attitude.

Key Words: biped locomotion robot, ROBO-ONE, attitude control, I-PD control

目次

第1章	はじめに	5
1.1	研究背景	5
1.2	宇宙システム	6
1.3	ROBO-ONE	6
第2章	仮定と問題の記述	7
2.1	仮定の記述	7
2.2	問題の記述	7
第3章	制御方式	8
3.1	宇宙システムにおける誘導, 制御系の構成	8
3.2	スラスタ制御	9
3.3	制御方式	9
3.3.1	PID制御	9
3.3.2	I-PD制御	11
第4章	部分的モデルマッチング法	13
第5章	シミュレーション環境	16
第6章	スラスタの搭載	17
6.1	スラスタの位置	17
第7章	角度応答シミュレーション	18
7.1	I-PD制御を用いた角度応答の安定化, 高速応答化	18
7.1.1	シミュレーション方法	18
7.1.2	シミュレーション結果	20
7.1.3	考察	22
第8章	追従制御	23

目次

8. 1	目標対象に対する追従制御の安定化	23
8. 2	シミュレーション方法	23
8. 3	シミュレーション結果	24
8. 4	考察.....	24
第9章	3軸姿勢制御.....	25
9. 1	I-PD 制御による3軸姿勢制御.....	25
9. 2	シミュレーション方法	25
9. 3	シミュレーション結果	26
9. 4	考察.....	26
第10章	移動制御.....	27
10. 1	I-PD 制御による移動制御.....	27
10. 1. 1	スラスタの位置の再決定.....	27
10. 1. 2	姿勢制御シミュレーション	28
10. 1. 3	移動シミュレーション	31
第11章	むすび.....	34
	参考文献.....	35
	ROBO-ONE 戦歴	36
	謝辞	37
	付録	38

第1章 はじめに

1. 1 研究背景

ロボットは産業用,アミューズメント,医療など多くの分野で活躍している.2000年の世界ロボット連盟(IFR)の統計によると,世界では約78万台の産業用ロボットが稼働している.その内,約45%は日本である.また,世界の日本製産業用ロボットの比率は60%を超え,様々な所で活躍している.Fig.1に2025年のロボットの市場予測を示す.

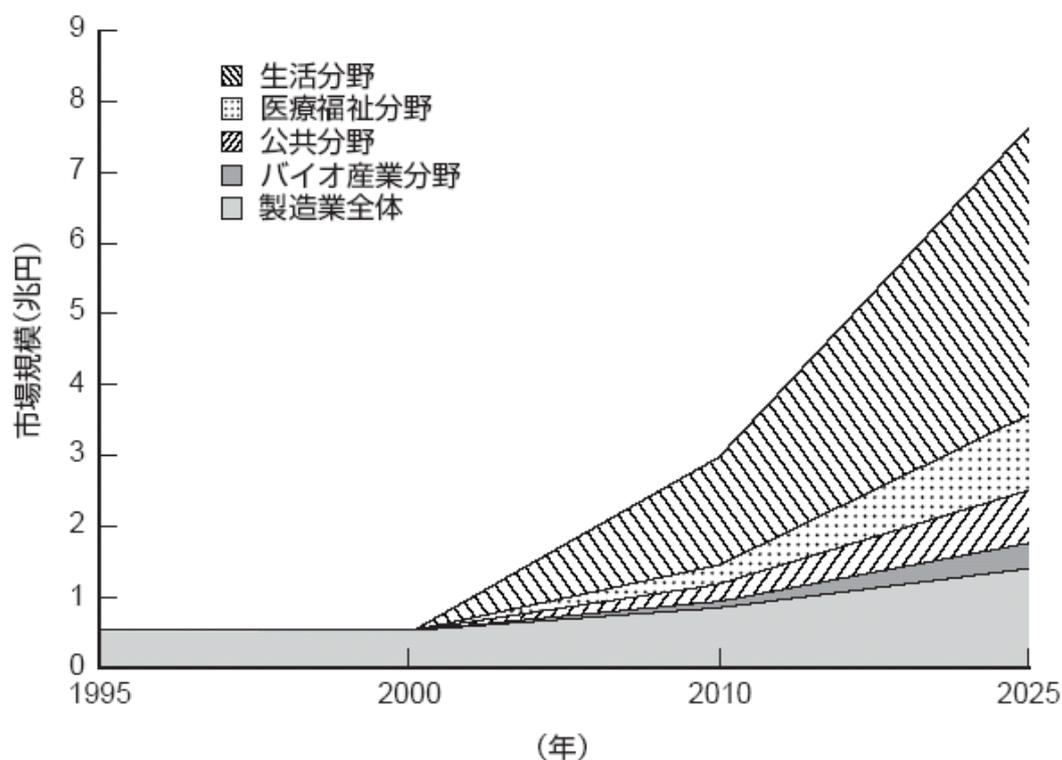


Fig.1 Market forecast of the robot.

産業用ロボットの多くは自動車,電化製品,半導体などの製造業で活躍してきた.そして,1980年代から日本の産業競争を強化するために,日本のロボットの技術力を製造業中心から日常生活支援や危険作業代替などへ移行する動きがあり,ロボット産業をさらに拡大する事が期待される.

現在,二足歩行ロボットは本田技研株式会社のASIMOに代表される様に,人々の身近な存在となってきている.

しかし、ロボットが人と共に共存する事は難しいと考えられる。

ロボットには危険作業代替や人間が使用する道具を、形状を変える事無くそのまま使用する事が出来るといった利点がある。そのような利点を生かす場は、宇宙であると考えた。

宇宙探査や衛星修理など人間が行うには危険が伴う作業をロボットが、現在人間が使用している道具をそのまま使用し作業を行う事が挙げられる。

ロボットの耐久性やシステムの信頼性向上がロボット普及における最大の課題である。現在の問題点として宇宙空間での二足歩行ロボットの姿勢制御が挙げられる。

そこで、本研究として宇宙空間でロボットが目標通りに姿勢を制御するために適切なコントローラを設計していく。

1. 2 宇宙システム

宇宙を眺める対象としてではなく、現実の行動の対象として考えるようになったとき、人類が考えたのは宇宙旅行であった。その第一歩として人工衛星が打ち上げられるようになったとき、人類は未知なる宇宙空間の現実に触れ探究心を駆りたてられると同時に、新しい宇宙空間の利用法に気づいた。真理の探求の対象として、見るもの(科学観測目的)と社会生活に役立たせるもの(実利用目的)との二つがあり、これらを軸に人類は宇宙開発を行ってきた。科学観測目的では天文観測や惑星探査等の科学衛星の系列を生み、実利用目的では通信衛星や気象観測衛星等の実用衛星の系列を生んだ。

スペースシャトルや宇宙ステーションの開発によりこれらを用いた新しい利用分野が生まれている。宇宙空間での長期滞在が人間に及ぼす医学上の影響のデータ収集や、それに必要な技術を開発するための実験、純度・均質度の高い材料やバイオ製品を作り出す実験などがある。さらに宇宙で発電所や観光施設の建設を行うといった構想まである。

1. 3 ROBO-ONE

日本ではROBO-ONEという二足歩行ロボットの競技大会が2002年より開催されている。この大会の概要はロボット同士が戦い、相手の機体を転倒させるものである。また2004年から、ROBO-ONE on PC(以下 on PC)という大会も始まった。on PCは2010年に開催予定の宇宙でのROBO-ONEを実現させるために設置されたパソコン上によるシミュレーション大会である。この宇宙大会の目的は前述の背景のもと、ロボット技術の大幅な進化を狙うところにある。ロボットの耐久性やシステムの信頼性向上がロボット普及における最大の課題と認識し、宇宙空間でも十分活用できる二足歩行ロボットやそれを管理するシステム技術の進化を目指している。

第2章 仮定と問題の記述

シミュレーションを行うにあたり，以下の仮定と問題を設ける．

2. 1 仮定の記述

- (A1) モータの角度出力を固定しロボットの形状を変化させない
- (A2) ロボットの質量変化は無いものとする

2. 2 問題の記述

- (P1) シミュレーションソフト内におけるノイズの検討
- (P2) I-PD 制御による角度応答の安定化，高速応答化
- (P3) 目標対象に対する追従制御の安定化
- (P4) I-PD 制御による3軸姿勢制御
- (P5) I-PD 制御による移動制御

本論では制御の手法としてPID制御を規範としたI-PD制御を用いる事とする．

第3章 制御方式

3.1 宇宙システムにおける誘導，制御系の構成

衛星やロケットのような宇宙機の操作には，軌道制御と姿勢制御がある．軌道制御とは宇宙機の位置，時間変化に関する操作であり，姿勢制御とは角度に関する操作であると定義できる．具体的には，軌道制御では位置・速度を決定する航法と目標の状態へもっていくための誘導，姿勢制御では目標の方向へ姿勢をもっていく指向とその状態で姿勢を保持しておく安定操作が，それぞれの主要な要素であると Fig.2 に示すように定義できる．

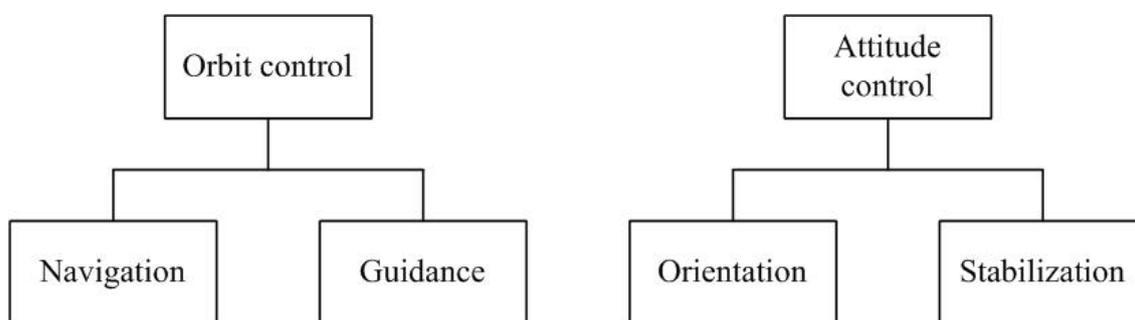


Fig.2 The details of orbit control and attitude control

また，誘導・制御系の基本的な構成を Fig.3 に示す．

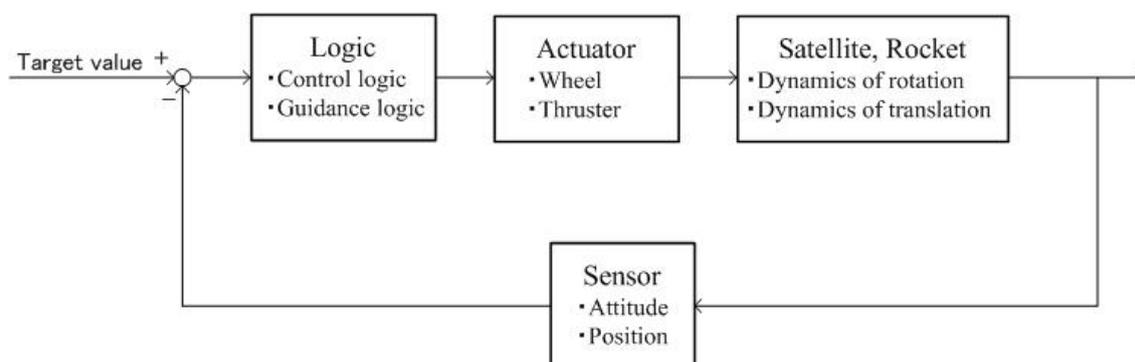


Fig.3 Fundamental structure of guidance and control system

制御対象はロボットであり，アクチュエータの力によって運動させる．しかし，目的とする運動を行うには，適当な指令をこのアクチュエータに与えなければならない．この指令は実際の姿勢・位置が目標からどれだけずれているかを求め，適切な論理計算を行うことによって得られる．実際の状態を知るには姿勢・位置のセンサによる検出が必要であり，センサを介した通常のフィードバック制御が必須である．

センサについては位置を知る加速度センサ，角度を知るジャイロ，姿勢基準を求める地球センサなどが挙げられる。

ロボットの運動は回転・並進の力学で表現される。通常は剛体の回転，質点系の運動として考えてよいが，必要に応じて弾性体や多質点系などの高度なモデルが使用される。姿勢制御では安定したミッションを遂行するために，定常誤差を少なくして精度を上げ過度の振動を抑えるような標準的な論理が用いられる。

3. 2 スラスタ制御

トルクを発生させるアクチュエータの代表的なものの一つにスラスタがある。スラスタはノズルから質量を高速で外部に放出することによって得られる反力を制御トルクに利用するものである。スラスタ制御系は微妙な制御が難しいかわりに，大きなトルクを発生させることが可能である¹⁾。on PC では ROBO-ONE 宇宙大会を想定しているので，ロボットの移動手段も考慮しなければならない。そこで本研究ではスラスタを用いて，姿勢制御を行うことにする。しかし，スラスタの燃料となる推薬に関してはここでは議論しない。

3. 3 制御方式

実際の現場でも用いられる様々なフィードバック制御について述べる。

3. 3. 1 PID 制御

まず，以下のように定義する。

$r(s)$: 目標値

$e(s)$: 偏差

$u(s)$: 操作量

$y(s)$: 制御量

$C(s)$: コントローラ

$P(s)$: 制御対象

k_p : 比例ゲイン

k_i : 積分ゲイン

k_d : 微分ゲイン

Fig.4 に PID 制御のブロック線図を示す。

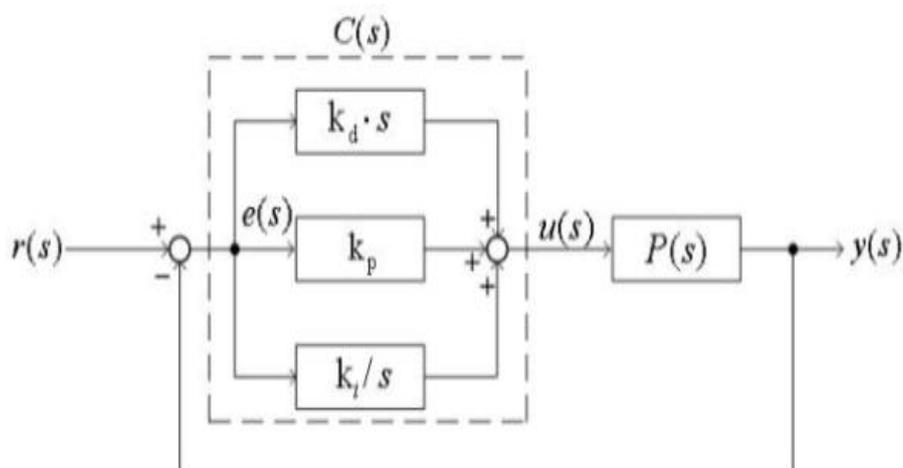


Fig.4 The block diagram of PID control

PID コントローラ $C(s)$ には以下の3つの要素が含まれている。

(1) 比例要素(P要素)

現在の情報を反映させる要素で、制御量 $y(s)$ と目標値 $r(s)$ との偏差 $e(s)$ が大きいとき操作量 $u(s)$ を大きくし、 $e(s)$ が小さいとき $u(s)$ を小さくする。

(2) 積分要素(I要素)

偏差 $e(s)$ の過去の情報を反映させる要素で、積分を行うことで定常偏差(オフセット)を無くす働きがある。

(3) 微分要素(D要素)

偏差 $e(s)$ の将来の動きを予測するような要素で、 $e(s)$ の変化量を反映させ安定性を向上させる。

PID コントローラ $C(s)$ を、ラプラス演算子 s を用いて表すと、

$$C(s) = k_p + \frac{k_i}{s} + k_d \cdot s \quad (1)$$

となる。

3. 3. 2 I-PD 制御

I-PD 制御のブロック線図を Fig.5 に示す.

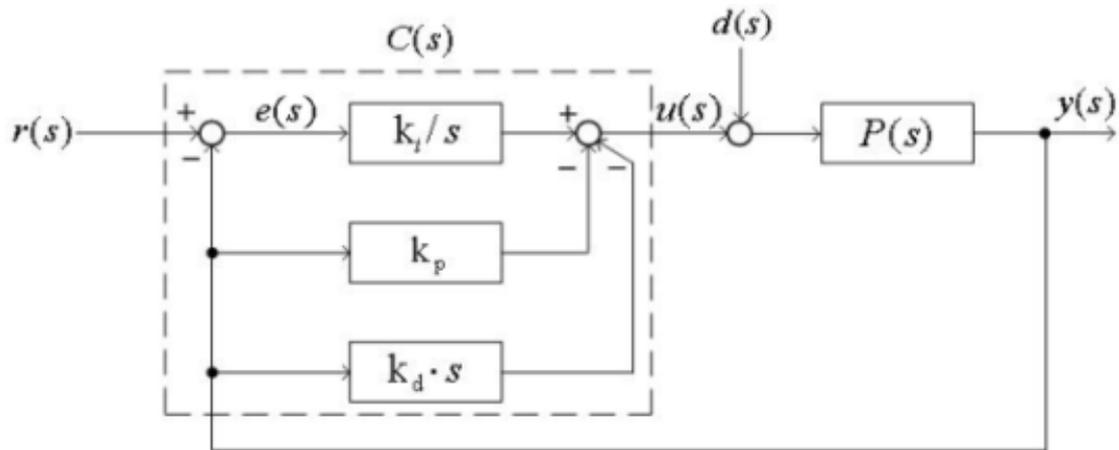


Fig.5 The block diagram of I-PD control

I-PD 制御は比例微分先行型 PID 制御であり，比例要素と微分要素を制御量 $y(s)$ に働かせたものである．ここで $u(s)$ は，

$$u(s) = \frac{k_i}{s} r(s) - \left(\frac{k_i}{s} + k_p + k_d \cdot s \right) y(s) \quad (2)$$

で表され，

$$C_1(s) = \frac{k_i}{s} + k_p + k_d \cdot s \quad (3a)$$

$$C_2(s) = \frac{k_i}{k_d s^2 + k_p s + k_i} \quad (3b)$$

とおくと式(2)は，

$$u(s) = C_1(s)(C_2(s)r(s) - y(s)) \quad (4)$$

と書き換えることができる．これは Fig.6 に示す 2 自由度制御系と同等である．
 入力 $r(s)$ 及び外乱 $d(s)$ から出力 $y(s)$ への伝達関数 $G_{yr}(s)$ ， $G_{yd}(s)$ は

$$G_{yr}(s) = \frac{P(s)C_1(s)C_2(s)}{1+P(s)C_1(s)} \quad (5a)$$

$$G_{yd}(s) = \frac{P(s)}{1+P(s)C_1(s)} \quad (5b)$$

で表される。

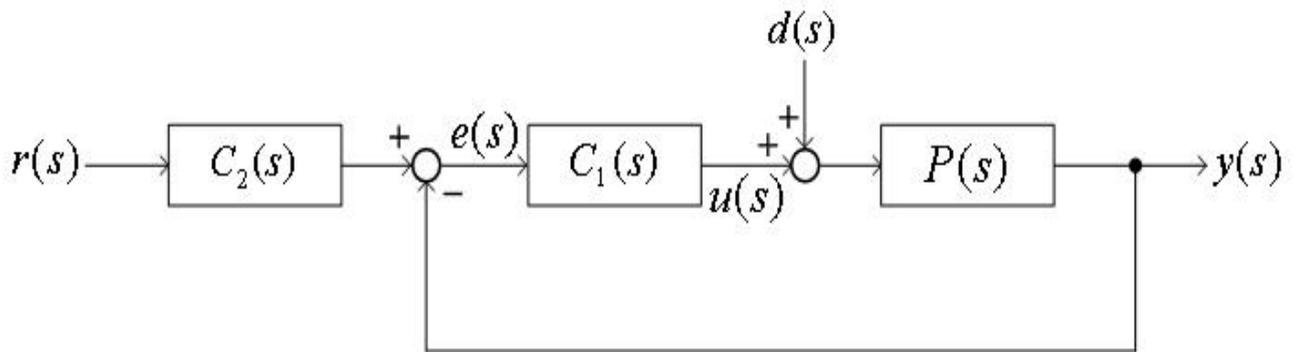


Fig.6 The block diagram of 2DOF control system

式(5a)及び式(5b)を用いて Fig.6 のシステムは

$$y(s) = G_{yr}(s)r(s) + G_{yd}(s)d(s) \quad (6)$$

となる。I-PD 制御では目標値 $r(s)$ を2次遅れ要素の目標値フィルタ $C_2(s)$ に通すことにより滑らかにし、PID 制御を行っている²⁾。

第4章 部分的モデルマッチング法

システムを安定にするためのコントローラ設計は実際容易ではない。そこで、ここでは I-PD 制御における部分的モデルマッチング法について述べる。部分的モデルマッチング法とは、システムを規範モデルと近似的に一致させる方法である。Fig.6 の $P(s)$ を、

$$P(s) = 1/p(s) \quad (7)$$

と、 $y(s)$ と $r(s)$ の関係は $d(s) = 0$ とすると式(8)で表す事が出来る。

$$y(s) = \frac{1}{1 + \frac{s}{k_i}(p(s) + k_p + k_d s)} \quad (8)$$

ここで、

$$y(s) = W(s)r(s) \quad (9)$$

$$W(s) = \frac{1}{1 + \alpha_1 s + \alpha_2 (\sigma s)^2 + \alpha_3 (\sigma s)^3 + \alpha_4 (\sigma s)^4} \quad (10)$$

$$p(s) = p_0 + p_1 s + p_2 s^2 + p_3 s^3 \quad (11)$$

式(11)を、式(8)に代入し、式(10) と s の係数を比較すると、

$$k_i \sigma = p_0 + k_p \quad (12a)$$

$$k_i \alpha_2 \sigma^2 = p_1 + k_d \quad (12b)$$

$$k_i \alpha_3 \sigma^3 = p_2 \quad (12c)$$

$$k_i \alpha_4 \sigma^4 = p_3 \quad (12d)$$

が成り立つ。

ここで、有効な参照モデルとして α のとるべき値を Table1 に示す。

Table1 Effective reference models on α_n

	n	α_2	α_3	α_4
Binomial	3	0.3333	0.03704	
	4	0.375	0.0625	0.003906
Bessel	3	0.4	0.0667	
	4	0.4268	0.09524	0.009524
ITAE	3	0.3786	0.1006	
	4	0.4664	0.1067	0.01882
Butterworth	3	0.5	0.125	
	4	0.503	0.1479	0.02188

また Table1 を用いたステップ応答波形を Fig.7 に示す. 制御対象P(s)は,

$$P(s) = \frac{1}{2s^2+3s+1} \quad (13)$$

である.

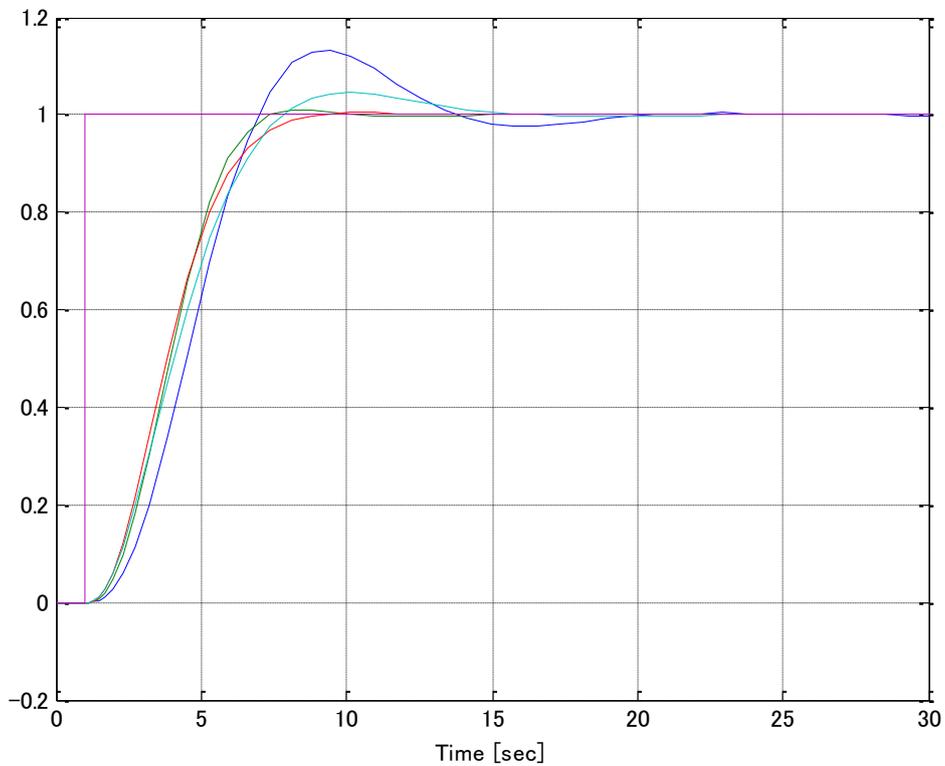


Fig.7 The each waveform by Table1

また, 高速応答狙いなら σ は小さな値を選ぶ. Fig.8に Table1 の Binomial を用いた σ のとり方による応答波形の違いを示す.

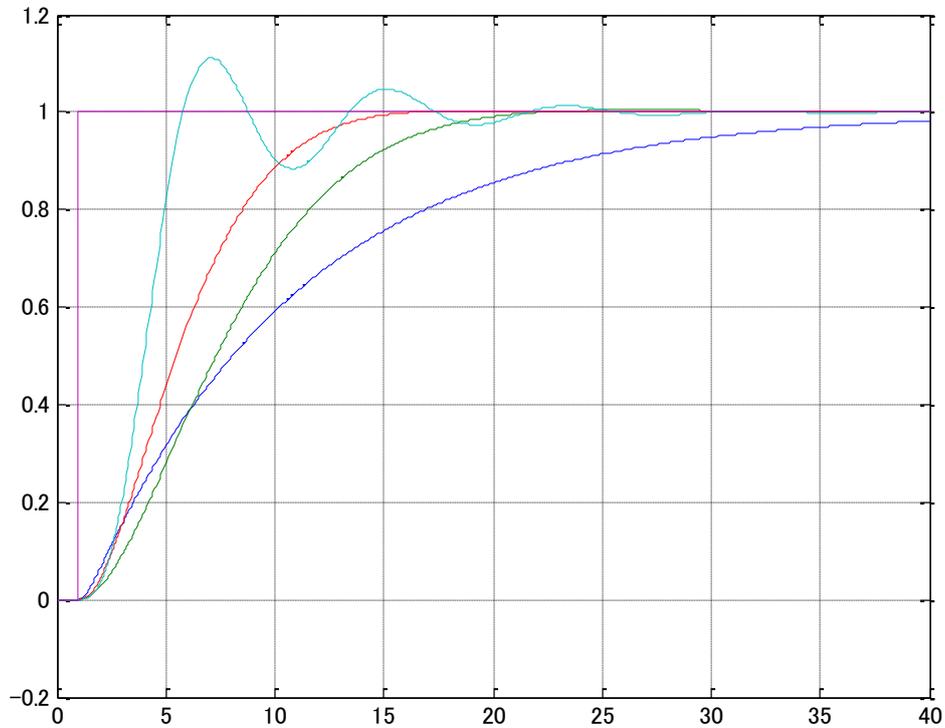


Fig.8 The each waveform by changing σ

Fig.8 より σ の値を大きくしていくと応答が遅くなっていくのが分かる.

しかし, σ の値を小さくしすぎると, オーバーシュートが発生し, 逆に収束が遅くなっていく. そのため, 適切な σ を選定していく事が重要である.

第5章 シミュレーション環境

本研究で用いるシミュレーションソフトを以下に示す.

- Autodesk Inventor
- MATLAB, Simulink
- VisualNastran4D(VN4D)

これらのシミュレーションソフトの相関図を Fig.9 に示す.

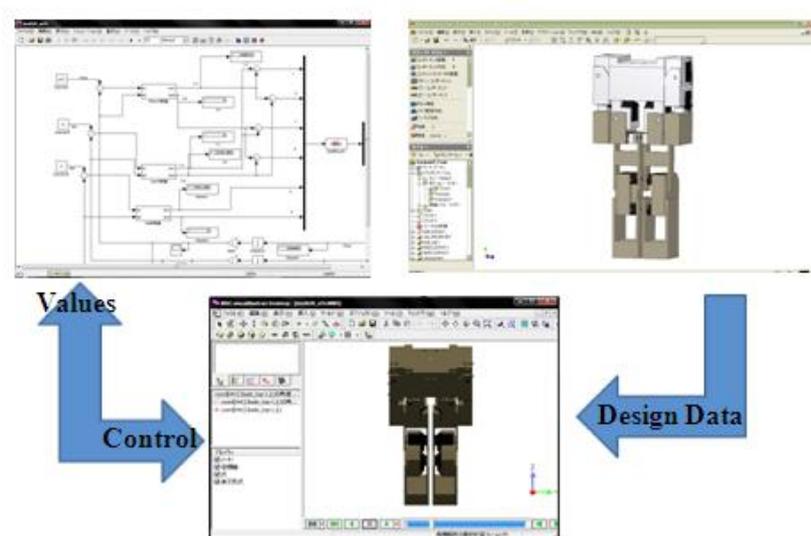


Fig.9 Correlation of software

Inventor でロボットの設計を行いVN4D にデータを送る. Simulink で制御系を組み, シミュレーションを VN4D と連動させて行う.

第6章 スラスタの搭載

6.1 スラスタの位置

Fig.10 の位置にスラスタを置き、座標軸を定義する。矢印の先がスラスタの位置であり、矢印の向きがロボットに力を与える方向である。

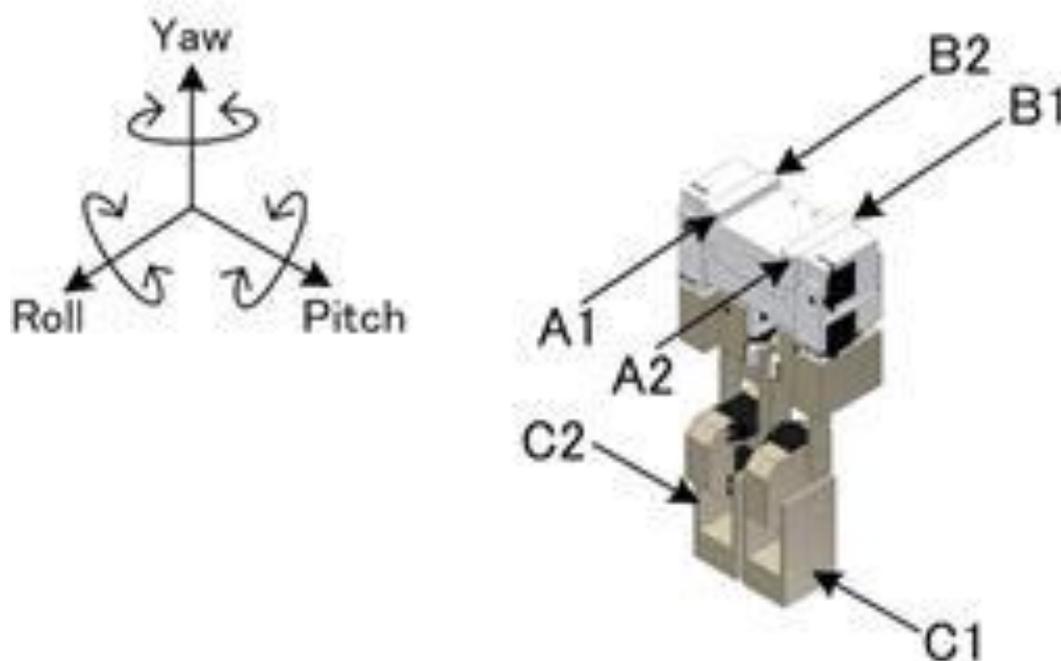


Fig.10 Positions of the each thruster and definition of axes

各軸の回転を得るためのスラスタは Table2 で示される組み合わせで行う。これらの組み合わせは、極力他の軸の回転に影響を与えないようにスラスタの位置を決定する。

Table2 Combinations of the each thruster

Roll	[C1], [C2]
Pitch	[A1,A2], [B1,B2]
Yaw	[A1,B1], [A2,B2]

第7章 角度応答シミュレーション

7. 1 I-PD 制御を用いた角度応答の安定化, 高速応答化

(P1)の問題について述べる. ROBO-ONE 宇宙大会を想定すると目標(相手のロボット)に姿勢を向けることが必要である. そこで, ここでは目標値を角度にとり, 姿勢を制御する事を考える. さらに, 目標に到達するために, 時間がかかると相手のロボットに対する行動が遅れてしまう. そこで高速化も図る.

7. 1. 1 シミュレーション方法

目標値を Roll 軸が $-\frac{\pi}{2}$ rad, Yaw 軸が $\frac{\pi}{4}$ rad, Pitch 軸が $\frac{\pi}{2}$ radとなるように与える. それぞれ1軸に対して角度応答シミュレーションを行う. 初期条件としてロボットはどの軸に対しても0radの方向を向いており, あらゆる速度を持たない. コントローラ内のパラメータ決定には部分的モデルマッチング法を用いて行う.

まず, 部分的モデルマッチング法を行うにあたり制御対象の伝達関数 $P(s)$ を求める. VN4D への入力力は F , 出力は角度 θ であるため $P(s) = \theta/F$ である. また, トルク T と力 F の関係は重心の位置から力のベクトルの作用線上までの仮想上の距離 R を用いてベクトルで考えない時,

$$T = R \cdot F \quad (14)$$

と表される. 仮想上の距離というのは, 各組み合わせで使うスラスタに与える力が同じである事を考慮して, トルクの計算において別のモデルと等価に置き換えた時の距離の事を意味する.

$$P(s) = \frac{\theta(s)}{F(s)} = \frac{R}{Is^2} \quad (15)$$

となる. ここで求めるべき物理量は慣性モーメント I である. 慣性モーメント I は,

$$I = \frac{T}{2\theta_{\Delta t}} \Delta t^2 \quad (16)$$

$\theta_{\Delta t}$: 回転角度

Δt :力を与えた時間

で表される.

Table3 に求めた慣性モーメント I_{mn} を示す.

I_{mn} : m と n のスラストを稼動させたときの慣性モーメント(8.3.1 と同様)

Table3 The moment of inertia of the robot [N·m·sec²/rad]

	mn		
	A1A2	A1B1	C1
I_{mn}	1.987×10^{-3}	4.334×10^{-4}	2.159×10^{-3}

また,

$$\begin{cases} I_{A1A2} = I_{B1B2} \\ I_{A1B1} = I_{A2B2} \\ I_{C1} = I_{C2} \end{cases} \quad (17)$$

である. 部分的モデルマッチング法を用いて求めたパラメータを Table4 に示す. 参照モデルはオーバーシュートが無い Table1 における Binomial で $\sigma = 0.4$ とする.

σ をあらかじめ決めたのは, $p(s)$ が(11)で表されるため, (12d)の関係式が成り立たず各パラメータが求まらないからである.

Table4 Parameter in the each controller

	mn		
	A1A2	A1B1	C1
k_{pmn}	2.11	1.35	2.85
k_{imn}	5.27	3.39	7.12
k_{dmn}	0.28	0.18	0.38

7. 1. 2 シミュレーション結果

結果を Fig.11 から Fig.13 に示す.

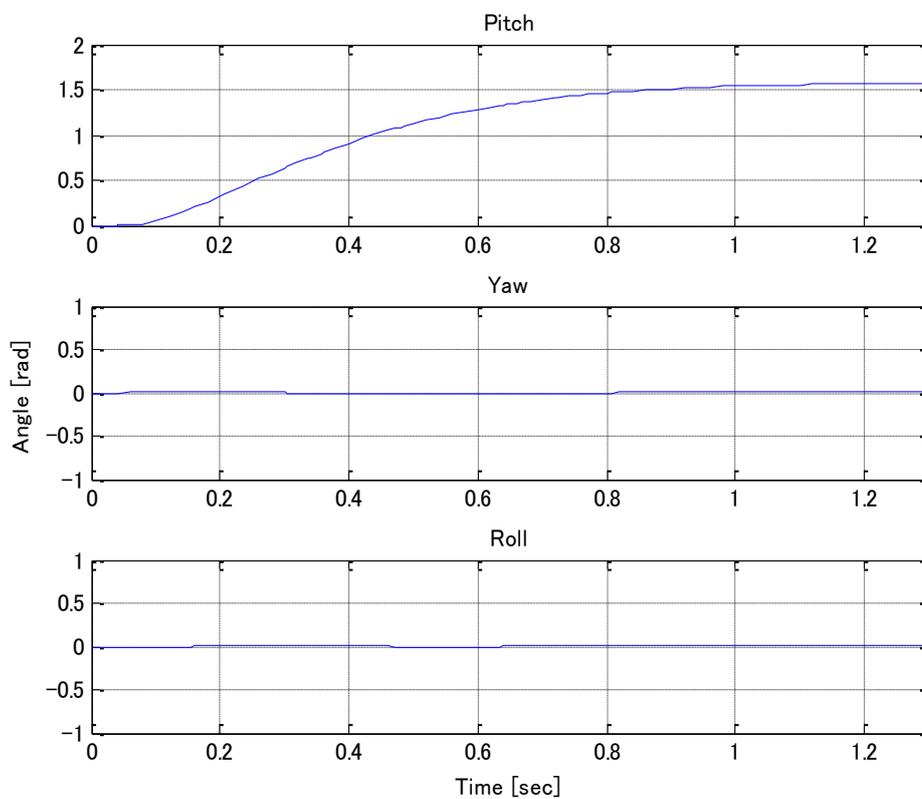


Fig.11 The result of angle of pitch axis

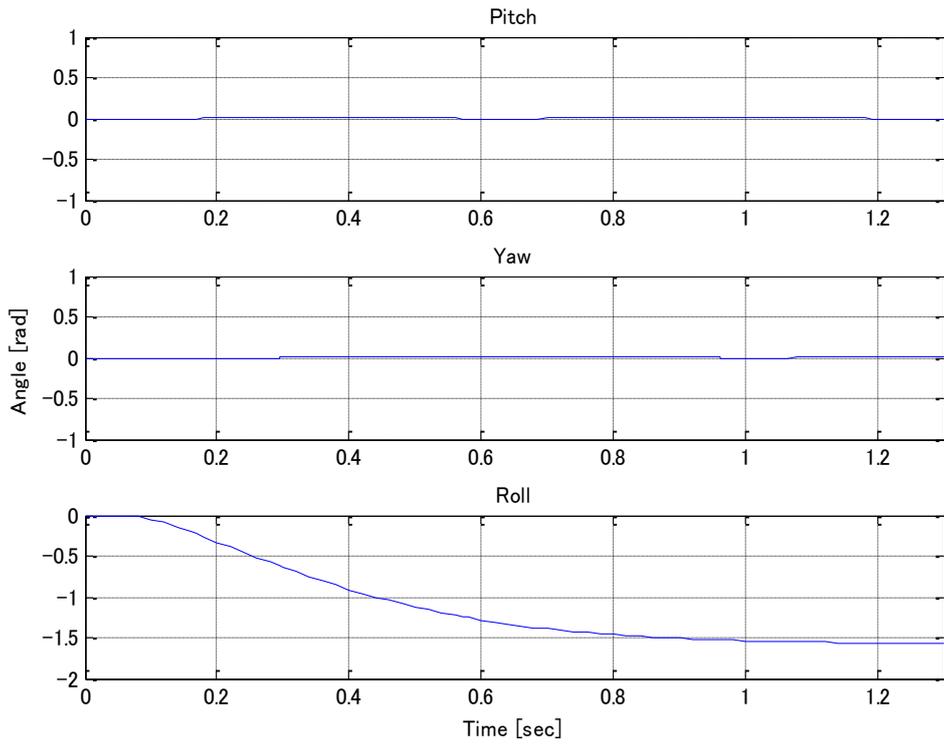


Fig.12 The result of angle of roll axis

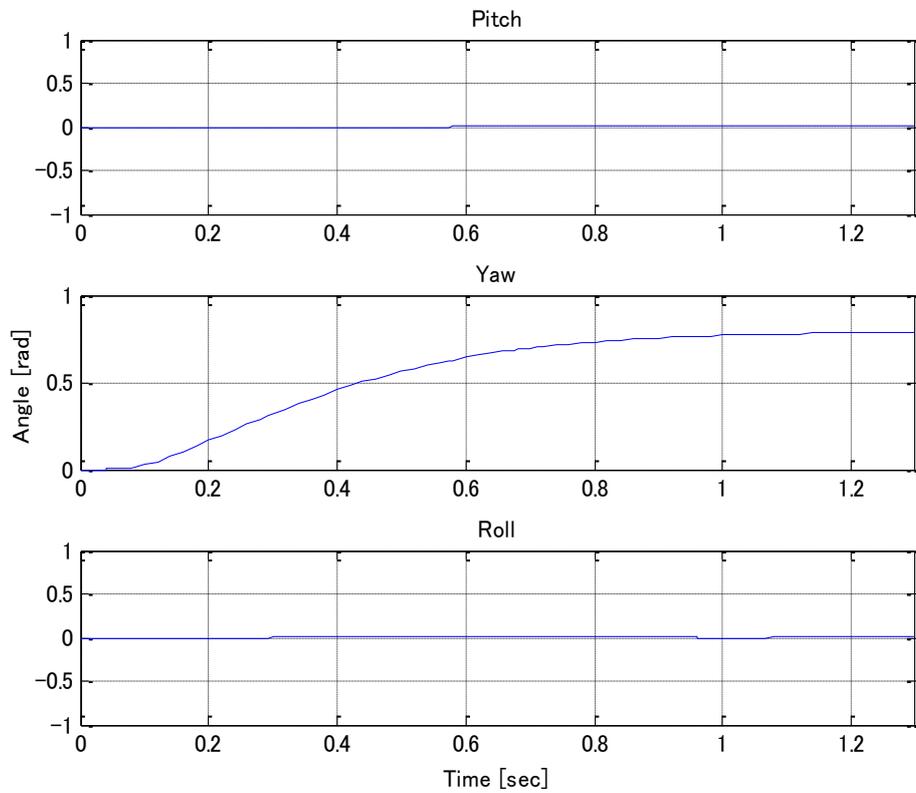


Fig.13 The result of angle of yaw axis

どの軸もオーバーシュート無し、定常偏差無しの良好な応答となった。約1sec付近で全ての軸の角度が目標値に至った。

σ を調節することで高速応答化を図る事が出来た。

7. 1. 3 考察

まず、先ほどのグラフの目標値を与えている角度をまとめて、Fig.14 に示す。

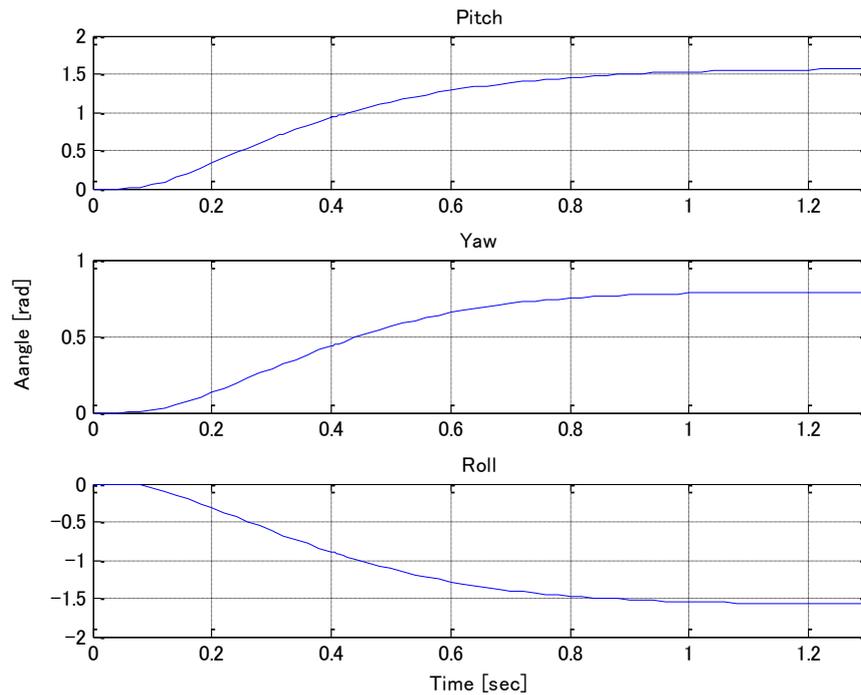


Fig.14 The result of angle

角度応答シミュレーションでは、 σ の値を小さくする事で応答速度を速くする事が出来ると考えた。しかし、 σ の値が小さ過ぎると逆に発散していく。そこで、シミュレーションを繰り返し行う事で最適な σ の値を選定していく事が必要である。

Fig.14 に示すように今回は $\sigma = 0.4$ とする事で、安定した応答を得る事が出来た。

第8章 追従制御

8.1 目標対象に対する追従制御の安定化

(P2)の問題について述べる。ROBO-ONE 宇宙大会では、目標物は相手のロボットである。ロボットは、自ロボットと同じく移動をするため、相手の移動に合わせて自分の姿勢を制御すればより優位に立てる。そこで、ここでは相手の移動に合わせて姿勢を制御し、さらに相手に対して常に正面を向く事を考える。

8.2 シミュレーション方法

7章で選定した σ を用いてシミュレーションを行う。ターゲットは、ロボットのYaw軸周りを $\frac{\pi}{10}\text{rad/s}$ で回転している事とし、ロボットはそれを追従する事とする。初期条件は7章と同じくロボットはどの軸も0radの方向を向いており、あらゆる速度を持たない事とする。

Yaw軸に対して追従制御をしていく。

8.3 シミュレーション結果

結果を Fig.15 に示す.

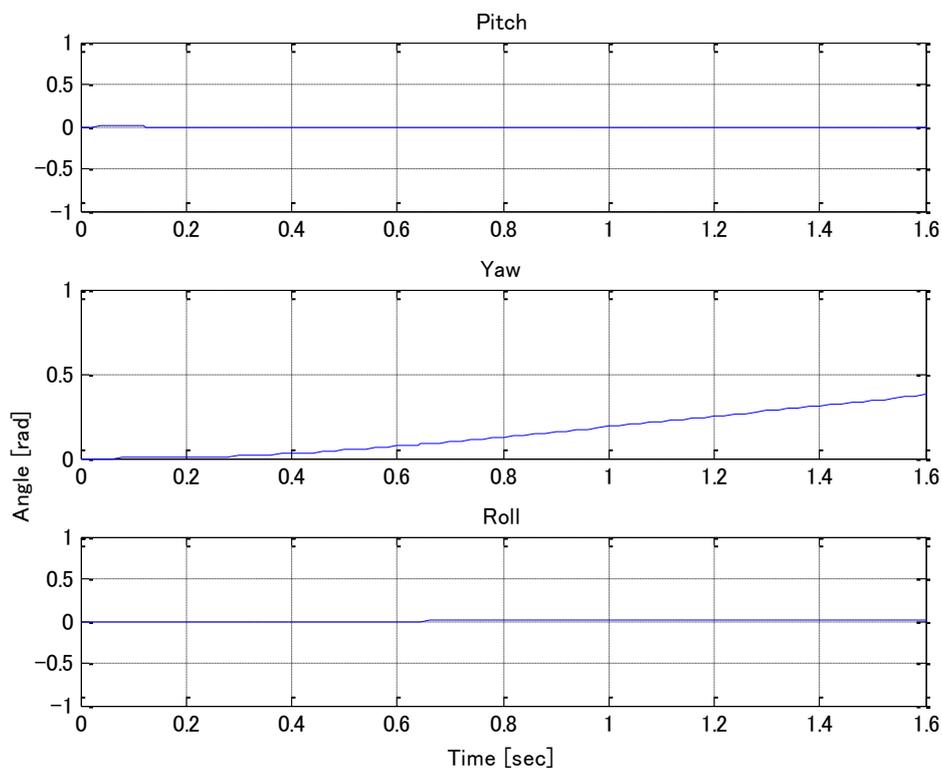


Fig.15 The result of angle

Yaw 軸に対して追従制御を行ったが、約 1sec 付近から安定して追従制御していく事が出来た.

8.4 考察

追従制御では、角度応答シミュレーションを応用する事で制御する事が出来た.

Fig.15 に示すように、約 1sec 後から、Yaw 軸のグラフの傾きは比例関係にある事から、安定した追従を行っている事が分かる.

第9章 3軸姿勢制御

9.1 I-PD 制御による3軸姿勢制御

(P3)の問題について述べる。7章では、姿勢制御を1軸ずつ行なった。しかし、実際ロボットの姿勢を制御するには、3軸を同時に制御した方が、高速化を図る事が出来ると考えた。

また、衛星を制御する際にも、同時に制御している。そこで、本章では、3軸同時に姿勢を制御し、安定化を図る。

9.2 シミュレーション方法

目標値を Roll 軸, Yaw 軸, Pitch 軸それぞれ $\frac{\pi}{4}$ rad となるように与える。初期条件としてロボットはどの軸に対して 0 rad の方向を向いており、あらゆる速度を持たない事とする。コントローラ内のパラメータ決定には部分的モデルマッチング法を用いて行う。

参照モデルは、Table1 における Binomial で Pitch 軸(A1A2) $\sigma = 6$, Roll 軸(A1B1) $\sigma = 17$, Yaw 軸(C1) $\sigma = 10$ を選択した。Table5 にコントローラのパラメータを示す。

Table5 Parameter in the each controller

	mn		
	A1A2	A1B1	C1
k_{pmn}	0.0094	7.4977×10^{-4}	0.0046
k_{imn}	0.016	4.4104×10^{-5}	4.5548×10^{-4}
k_{dmn}	0.0187	0.0042	0.0152

9.3 シミュレーション結果

結果を Fig.16 に示す.

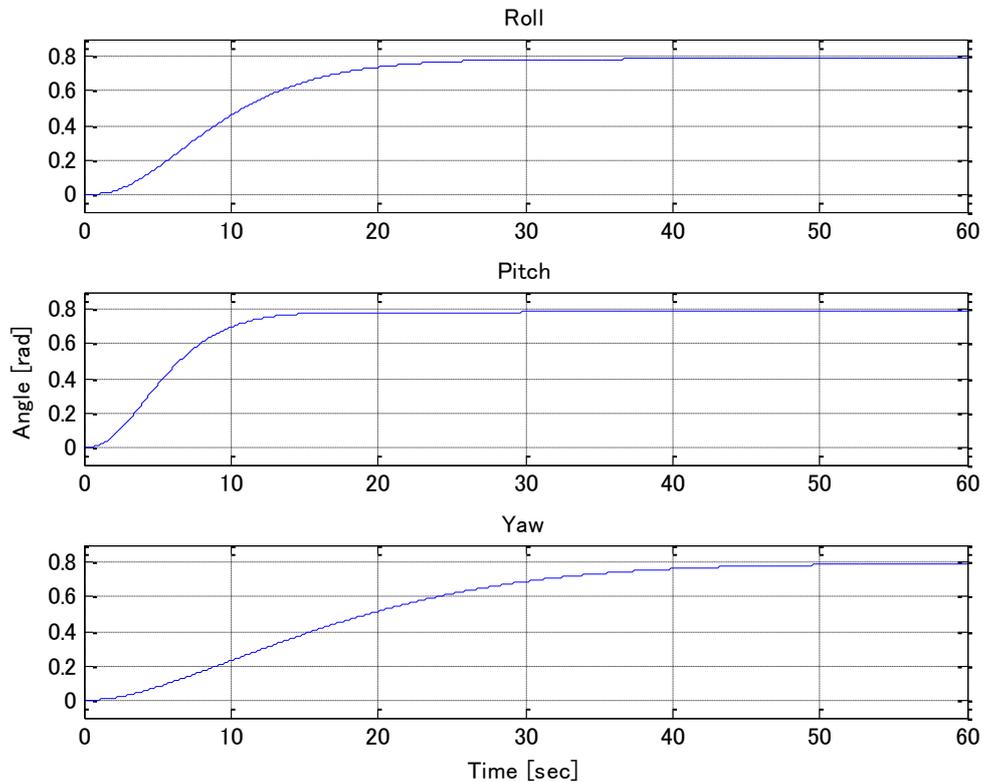


Fig.16 The result of angle

どの軸も、オーバーシュート無し、定常偏差無しの良好な応答となった.

9.4 考察

3軸制御では、3軸の σ 値のずれを調節する事で安定した応答を得る事が出来た. Pitch軸とYaw軸制御で同じスラストを用いているため、足し合わせによる姿勢制御の誤差を、コントローラを用いて制御する事で安定させる事が出来た.

全ての軸が安定するのに約50secかかっている. 7章の方法でそれぞれ1軸ずつ制御した場合、1軸が安定するのに約1secしかかかっていないので、全ての軸を制御するのに、約3secとなる. そのため、高速化を図る事が必要と考えられる.

第10章 移動制御

10.1 I-PD 制御による移動制御

(P4)の問題について述べる。ロボットが、ROBO-ONE 宇宙大会で動作するために必要なもう一つの要素として、移動が挙げられる。移動出来ないと相手のロボットに対して何もアプローチが出来ない。そこで、本章ではスラストによる移動制御の安定化を図る。

10.1.1 スラストの位置の再決定

移動制御するにあたり、ロボットが移動する手段として、姿勢制御でも使用しているスラストを用いて行なう事とする。

しかし、Fig.10 で設置したスラストの位置は、ロボットに対して回転のみを与える目的で設定している。

そのため、新たにスラストの位置を再設定しなくてはならない。そこで、Fig.17 に示すようにスラストの位置を再設定した。

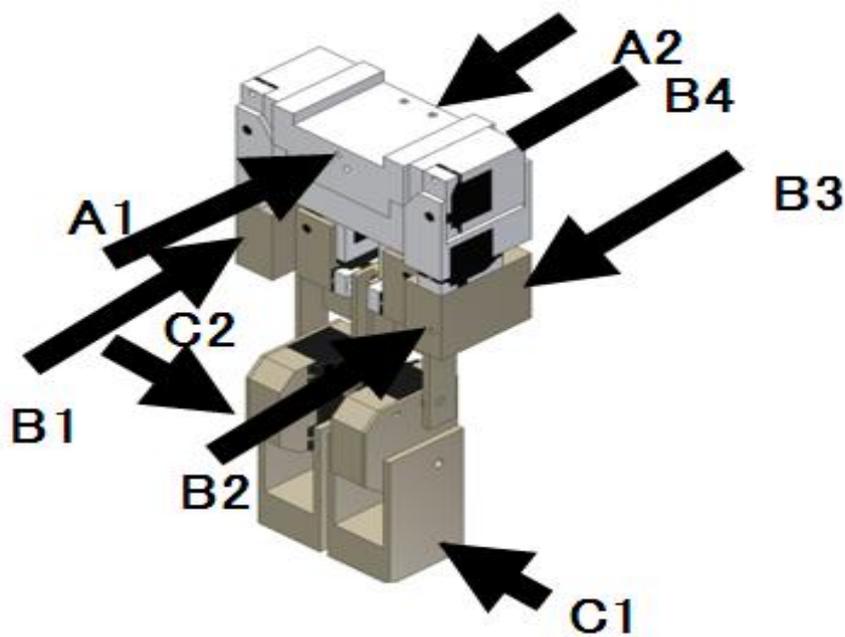


Fig.17 Positions of the each thruster

移動する方向については、ロボットの前後方向のみと仮定する。各軸の回転、移動を得るためのスラスタは Table6 で示される組み合わせで行う。回転は極力、他の軸の回転に影響を与えないようにスラスタの位置を決定する。

Table6 Combinations of the each thruster

Roll	[B1,B3] , [B2,B4]
Pitch	[A1] , [A2]
Yaw	[C1] , [C2]
Forward	[B2,B4]
Back	[B1,B3]

10. 1. 2 姿勢制御シミュレーション

10. 1. 2. 1 シミュレーション方法

スラスタの位置を変更したため、再度姿勢シミュレーションを行なう。今回は、簡易モデルを設計し、そのモデルを用いてパラメータ設計を行なう。簡易モデルで得られたパラメータを用いて実機モデルによるシミュレーションを行なう事によってシミュレーション時間の短縮を図る。

Roll 軸, Yaw 軸, Pitch 軸それぞれ $\frac{\pi}{4}$ radとなるように与える。初期条件としてロボットはどの軸に対して0radの方向を向いており、あらゆる速度を持たない事とする。コントローラ内のパラメータ決定には部分的モデルマッチング法を用いて行う。Table7 に求めた慣性モーメント I_{mn} を示す。

Table7 The moment of inertia of the robot [N·m·sec²/rad]

	mn		
	A1	B1B3	C1
I_{mn}	1.7964×10^{-3}	2.0855×10^{-3}	4.2869×10^{-3}

また,

$$\begin{cases} I_{A1} = I_{A2} \\ I_{B1B3} = I_{B2B4} \\ I_{C1} = I_{C2} \end{cases} \quad (18)$$

である。参照モデルは、Table1における Binomial で $\sigma = 5$ を選択した。Table8にコントローラのパラメータを示す。

Table8 Parameter in the each controller

	mn		
	A1	B1B3	C1
k_{pmn}	0.0277	0.0450	0.0356
k_{imn}	0.055	0.0090	0.0071
k_{dmn}	0.0462	0.0751	0.0593

10. 1. 2. 2 シミュレーション結果

結果を fig.18, Fig.19 に示す。

まずは、簡易モデルによるシミュレーション結果である。

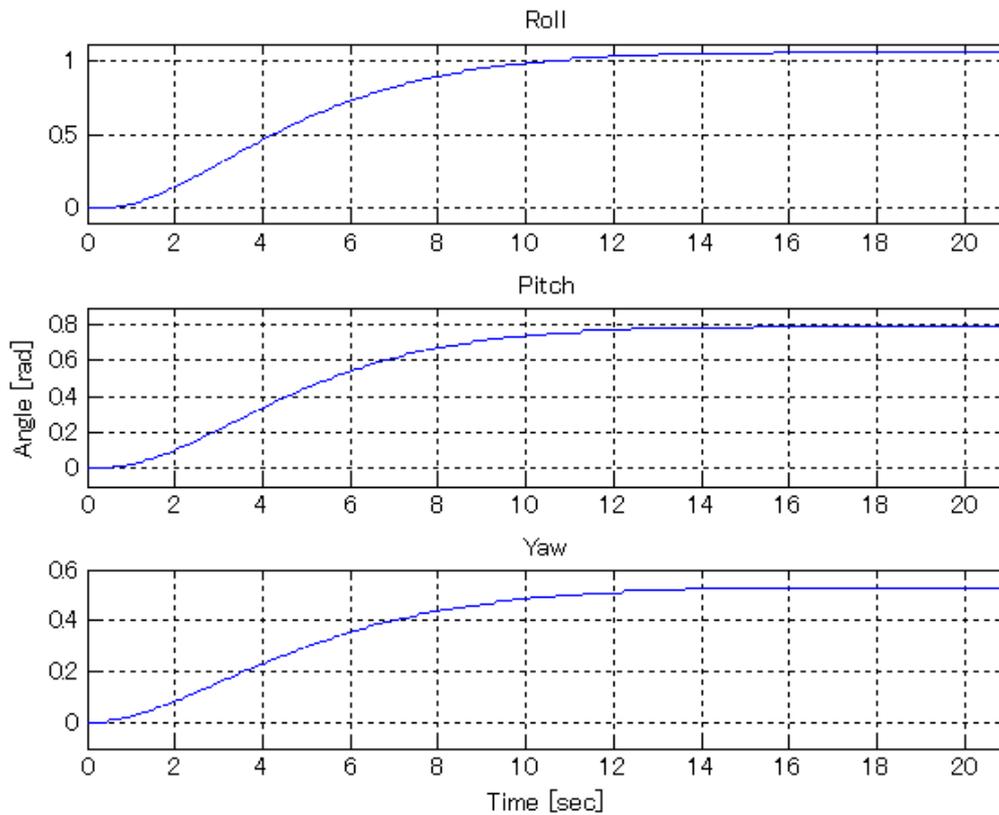


Fig.18 The result of angle

次に、実機モデルによるシミュレーション結果である。

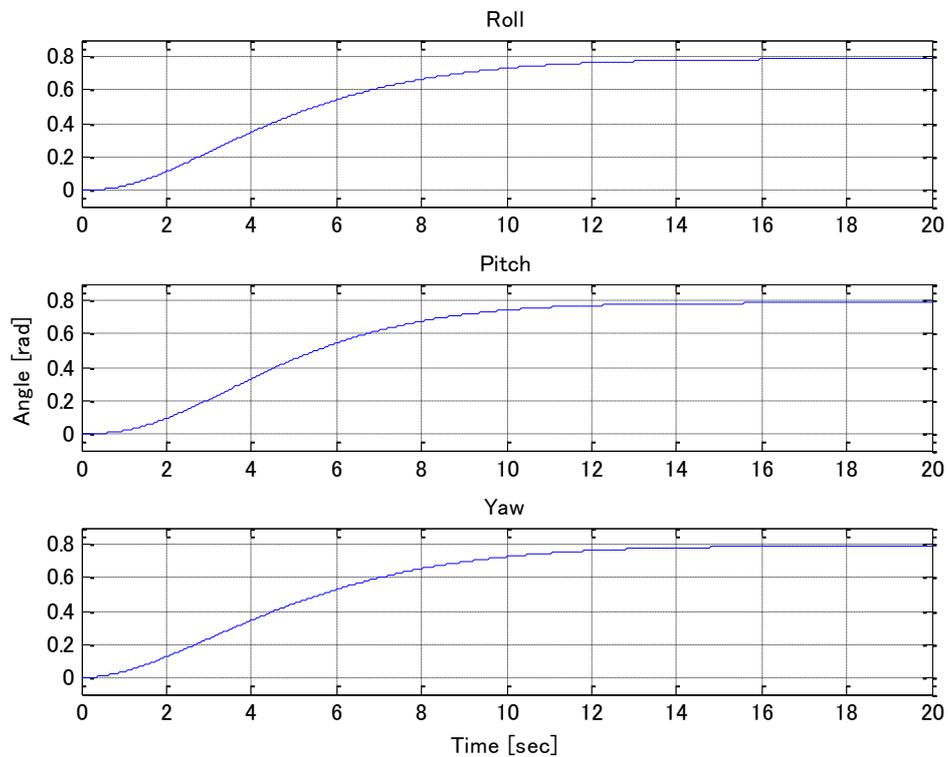


Fig.19 The result of angle

Fig.18, Fig.19 とともに、オーバーシュート無し、定常偏差無しの安定した波形となった。シミュレーションとしては、簡易モデルでシミュレーションする事により、一回にかかるシミュレーション時間が大幅に短縮された。

10. 1. 2. 3 考察

スラスタの位置を変更しても、姿勢を制御可能である事が分かった。スラスタの位置は、他の軸を考慮する事によって設定した。

また、簡易モデルを用いる事で一つのシミュレーションにかかる時間が大幅に短縮された。そのため、何度もシミュレーションを行なってもトータルとして短縮化を図る事が出来た。

10. 1. 3 移動シミュレーション

10. 3. 1. 1 シミュレーション方法

目標値をロボットが 0.3[m]前進する事となるように与える. また, ロボットの姿勢は初期姿勢を保つように 3 軸それぞれに目標値を与える.

以上の条件を踏まえ, 部分的モデルマッチングを行いコントローラのパラメータを設計していく.

まず, 部分的モデルマッチング法を行うにあたり制御対象の伝達関数 $P(s)$ を求める. VN4D への入力力は F , 出力は距離 r であるため,

$$P(s) = \frac{r(s)}{F(s)} = \frac{1}{Ms^2} \quad (19)$$

となる. Table8 にコントローラのパラメータを示す.

参照モデルは, Table9 における Binomial で $\sigma = 6$ を選択した.

Table9 Parameter in the each controller

	m
k_{pm}	0.2492
k_{im}	0.0356
k_{dm}	0.5815

10.3.1.2 シミュレーション結果

結果を Fig.20 に示す.

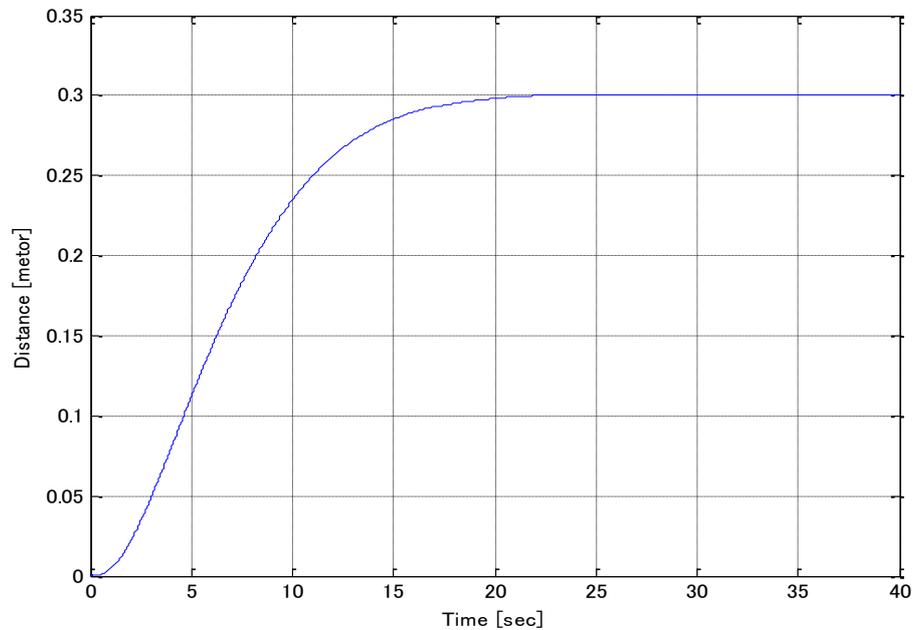


Fig.20 The result of distance

約 22sec で目標値である 0.3m に収束した. オーバーシュート無しの良好な応答となった.

10.3.1.3 考察

現在の姿勢を制御したまま, 移動制御を行なったため, 時間はかかったが移動制御可能であった. 移動するためのスラストと Yaw 軸を制御するためのスラストが同じであるため, 足し合わせの誤差を考慮したコントローラ的设计をする事が出来た.

また, Fig.21 に移動している時の, ロボットの姿勢角度を示す.

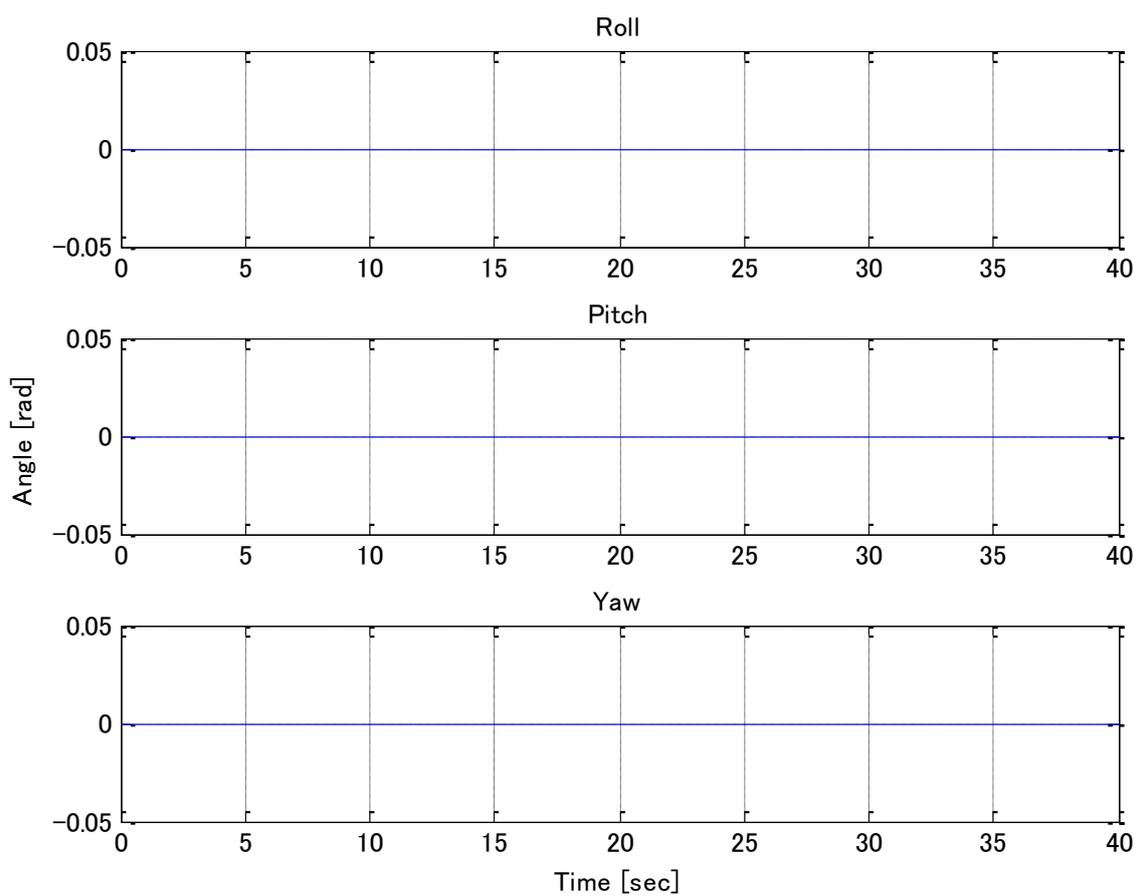


Fig.21 The result of angle

Fig.21 に示すように、移動の際、ロボットの姿勢角度は常に 0rad である。よって、姿勢を初期状態から、変化させずに移動出来ている。

第 11 章 むすび

ROBO-ONE 宇宙大会においてロボットの姿勢制御，移動制御は必要不可欠である．本研究では I-PD 制御を用いて姿勢の安定化，高速化，目標物に対する追跡行動の安定化，移動制御の安定化について行った．特に I-PD 制御のコントローラ設計では部分的モデルマッチング法を用い，不安定系な制御対象をシステム全体として安定にする事が出来た．

無重力空間での姿勢制御，移動制御を行なう事が出来たが，宇宙では軌道制御などが挙げられる．そこで，衛星軌道を考慮した姿勢制御，移動制御を行なう必要がある．

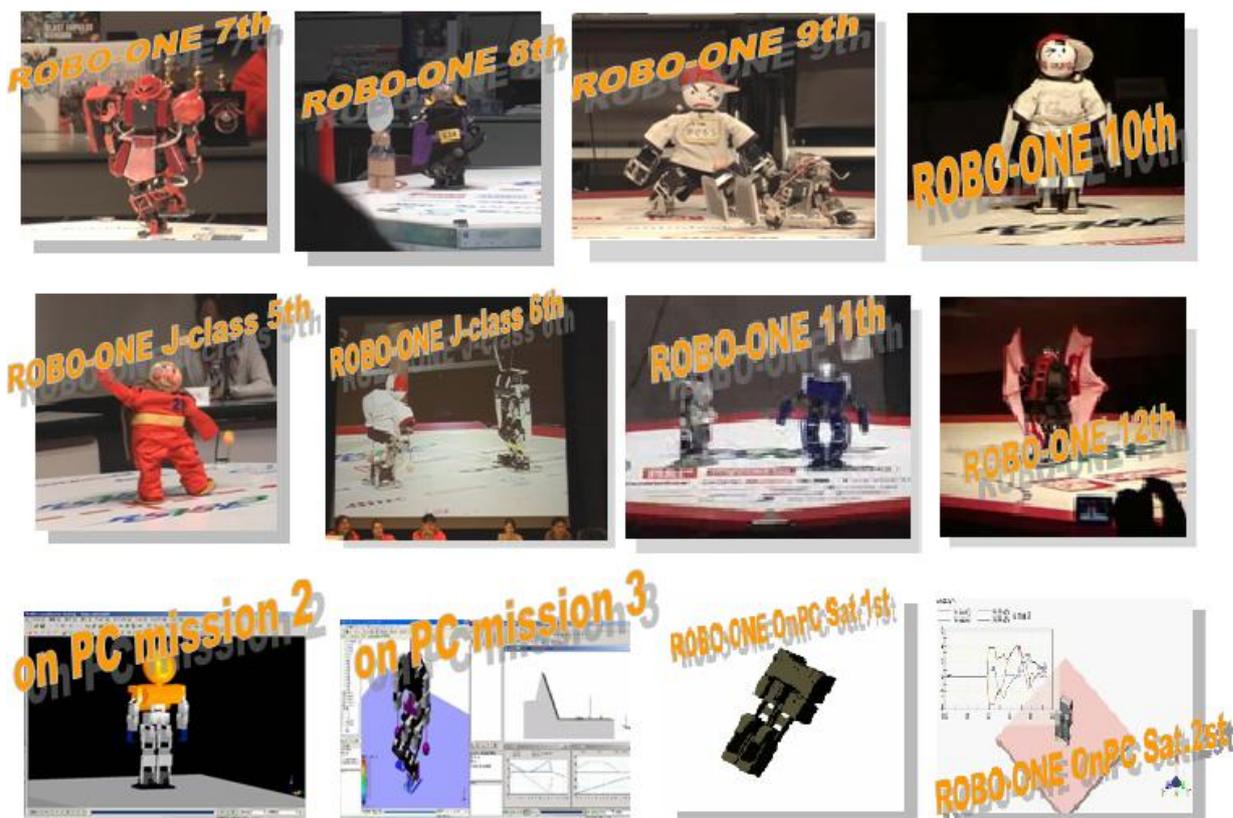
宇宙空間でも同様の制御を行なう事が出来れば ROBO-ONE 宇宙大会でより効果的な動きを発揮する事が出来る．

参考文献

- 1) 茂原正道：宇宙工学入門，培風館，pp56，1994
- 2) 井上和夫，川田昌克，西岡勝博：MATLAB/Simulink によるわかりやすい制御工学，森北出版，pp92-94，2001

ROBO-ONE 戦歴

- 第7回ROBO-ONE 予選7 3位
- 第5回ROBO-ONE J-class 予選2 0位 決勝トーナメント Best 1 6位
- 第8回ROBO-ONE 予選2 4位 決勝トーナメント Best 3 2位
- ROBO-ONE on PC mission 2 6位入賞
- ROBO-ONE on PC mission 3 5位入賞
- 第6回ROBO-ONE J-class 予選2 6位 決勝トーナメント Best 3 2位
- 第9回ROBO-ONE 予選4 4位
- 第10回ROBO-ONE 予選4 3位
- ROBO-ONE on PC Sat.1st
- 第11回ROBO-ONE 予選2 3位
- 第12回ROBO-ONE 予選2 0位
- 予選2 3位
- ROBO-ONE on PC Sat.2st 準優勝



謝辞

本研究を進めるにあたり適切なご指導ご教示を頂いた渡辺嘉二郎教授，小林一行准教授に深く感謝いたします。

また，ロボット班として長年切磋琢磨し，支えてくれた人たちに感謝します。宮川君。彼がいなければ今の自分はないといっても過言ではありません。スケジュール管理，先を見る力，そして，リーダーとして班を盛り上げていく力など見習うべき点がたくさんありました。青木君。自分を追い込んだ時の力の出し方はすごいという表現しか出来ません。

三穂君。様々な分野を研究してきたその探求心。いつまでも持ち続けて下さい。

先輩方もありがとうございました。特に一色先輩には CAD の使用方法からロボットの設計まで一からご指導していただきました。ありがとうございます。

後輩の合田君。色々無茶な事でも積極的に行動する力いつまでも持ち続けて下さい。田倉君。シミュレーションは大変だったのにも関わらず，最後までやり通す意志の強さ。いつまでも大事にして下さい。

岡本君，中山君，小宮山さん。ロボット班の今後は三人の力にかかっています。頑張ってください。

また，杉本君，徳田君，増山君，立川君，梅村君。研究中にも関わらず色々話をしてくれてありがとう。杉本君は，特にプライベートでも色々お世話になりました。

最後に何不自由なく大学院まで行かせてくれた両親に深く感謝いたします。

付録

本研究で行ったシミュレーションの動画を付属の CD-RW に添付する。
各動画のファイル名は以下の通りである。

ALL_A_B.wma

ALL_A_F.wma

all_angle_before_f.wma

all_f.wma

easy_dis.wma

easy_robo_3_F.wma

easy_robo_F.wma

PITCH_0001.wma

Robo_dis.wma

ROLL_0002.wma

target.wma

YAW.wma