

MATLABを用いた水中ロボットシミュレーションシステムの開発

森元映治[†], 平 雄一郎, 中村 誠

Development of a Simulation System for Underwater Robots by MATLAB

Eiji Morimoto[†], Yuichiro Taira and Makoto Nakamura

An autonomous underwater robot equipped with a manipulator is expected to play an important role in future ocean development such as repairing marine cables, constructing offshore structures and removing seabed sediment. In order to make it fit for practical use, an effective control method for the manipulator in the presence of hydrodynamic effects needs to be developed. In general, control methods for mechanical systems are validated by simulation and/or experiment. A simulation study is cost-effective in comparison with an experimental one, especially in such an expensive system as underwater robot that requires a huge tank, waterproof robots, and so on. In this report, we develop a simulation system for underwater robot manipulators by means of the computer software MATLAB/Simulink. Since the software is widely used in the field of control and robotics and has many tools for controllers, the underwater robot simulation system developed in this report facilitates confirming the effectiveness of a control method for underwater robot manipulators.

Fishery engineering, Simulation, Underwater robot, Manipulators, MATLAB

緒 言

魚介類行動生態調査, 漁場における海底障害物・汚染源の除去作業, 海底ケーブルの設置・補修などの海中作業を安全かつ効率的に行うために, 海中ロボットの利用が期待されている^{1,2)}。このロボットには作業のために人間の腕に相当するマニピュレータが搭載されている必要がある。また, 人件費などのコスト面や作業者の負担を考慮すれば, ロボットは自律的に作業を遂行する機能を備えている方が望ましい。本報では, 作業用マニピュレータを有する自律型水中ロボット(海中ロボットを含む)を対象とする。

自律型水中ロボットの実用化と普及のためには, マニピュレータに対する高性能な動作制御法の開発が必要である。一般に, 制御法の開発においては, 実機を用いた実験によりその性能を解析・評価する。しかしながら, 水中ロボットマニピュレータのように, 充実した実験環境を得ることが容易ではない対象ではこのことが支障となる。な

お, 水中ロボットマニピュレータの制御実験としては, たとえば文献³⁾⁻⁶⁾が報告されているが, 安全性を確保できるくらい大きな水槽や防水したロボットなど, 費用のかかる実験装置を用いている。このような場合, 計算機による数値シミュレーション解析が有効である。そこで, 著者らはニューラルネットワークを用いたシミュレーション法を報告している⁷⁾。この手法は実験データの学習に基づくシミュレーション法であり, 水中ロボットマニピュレータのような正確な数学モデルを得ることが困難である対象に有用である。しかしこの手法には, 1) ロボットのあらゆる動作を再現するために多くのパターンの学習用実験データが必要である, 2) このシミュレーションシステムにコントローラ(制御入力)を組み込むことが容易ではない, という問題がある。

一方, 制御工学分野では, MATLAB/Simulink^{8,9)}という制御用ソフトウェアが計算機シミュレーションの際によく用いられている。このソフトには, コントローラの開発

に必要なものがすでに用意されており、それらを組み合わせれば複雑で大規模なコントローラも比較的容易に作成することができる。本報では、MATLABを用いた水中ロボットマニピュレータのシミュレーションシステムを開発する。

方 法

モデルの設計

ここでは、制御シミュレーションシステムにおいて制御対象に相当するロボットのモデル部分を構築する。このモデルは2リンクマニピュレータを有する水中ロボットとし、水面に垂直な2次元平面内で運動するものとする。このような垂直型2リンク水中ロボットマニピュレータの運動方程式は次式で与えられる¹⁰⁾。

$$M(z)\ddot{z} + c(z, \dot{z}) + f(z, \dot{z}) + g(z) = \tau \quad (1)$$

ただし、 $M(z) \in R^{3 \times 3}$ は付加質量を含む慣性行列、 $c(z, \dot{z}) \in R^3$ は付加質量を含む遠心力・コリオリ力ベクトル、 $f(z, \dot{z}) \in R^3$ は流体抗力ベクトル、 $g(z) \in R^3$ は重力・浮力ベクトル、 $\tau \in R^3$ は本体の推進力およびマニピュレータの関節トルクベクトル、 $z \in R^3$ は本体の位置姿勢およびマニピュレータの関節角度ベクトルである。 $M(z)$ が正定対称行列であること¹⁰⁾を考慮し、式(1)の両辺に $M(z)^{-1}$ を掛け整理すれば次式を得る。

$$\ddot{z} = M(z)^{-1} \{ \tau - c(z, \dot{z}) - f(z, \dot{z}) - g(z) \} \quad (2)$$

式(2)の非線形微分方程式の厳密解を求めることは困難である。このような場合、通常のシミュレーションでは、数値積分により位置 z と速度 \dot{z} を求める。一方、MATLAB/Simulinkでは、積分要素がすでに用意されているため、Fig. 1のように構成すればよい。なお、Simulinkは図のようにブロック線図(構成要素をブロックで表現し、信号の

流れを表す線でこれを結んだ線図)で構築したシステムに対してシミュレーションを実行するものである。モデル部分のシミュレーションの流れは以下のとおりである。

- 1) ある制御目的を達成する制御入力 τ がコントローラからモデル部分に入力される。
- 2) 制御入力 τ と、現時点の位置 z と速度 \dot{z} を式(2)の右辺に代入することにより加速度 \ddot{z} が求まる。
- 3) 加速度 \ddot{z} を積分することにより次時点の速度 \dot{z} が求まる。
- 4) 次時点の速度 \dot{z} を積分することにより次時点の位置 z が求まる。
- 5) 計算された位置 z と速度 \dot{z} がモデル部分から出力され、コントローラで利用される。

関節サーボドライバの設計

一般に、マニピュレータの各関節に取り付けられたモータは、速度入力型あるいはトルク入力型サーボドライバにより駆動される。本報で開発する水中ロボットシミュレーションシステムは、文献6)で使用された実機を参考に開発されており、この実機には速度入力型サーボドライバが実装されている。ここでは、制御シミュレーションシステムにおけるサーボドライバ部分を構築する。一般のサーボドライバと同様に、マニピュレータの関節角速度が目標値(定数)に調整されるような簡単なPI制御式サーボドライバを採用する。MATLAB/Simulinkでは、ゲイン要素、積分要素、加え合わせ要素(複数の信号の和を出力する要素)が用意されているため、前項で設計したモデル部分と組み合わせるとサーボドライバ部分をFig. 2のように構成すればよい。なお、図において、 K_p と K_i はそれぞれ位置と速度ゲイン、 z_d は目標値である。サーボドライバ部分のシミュレーションの流れは以下のとおりである。

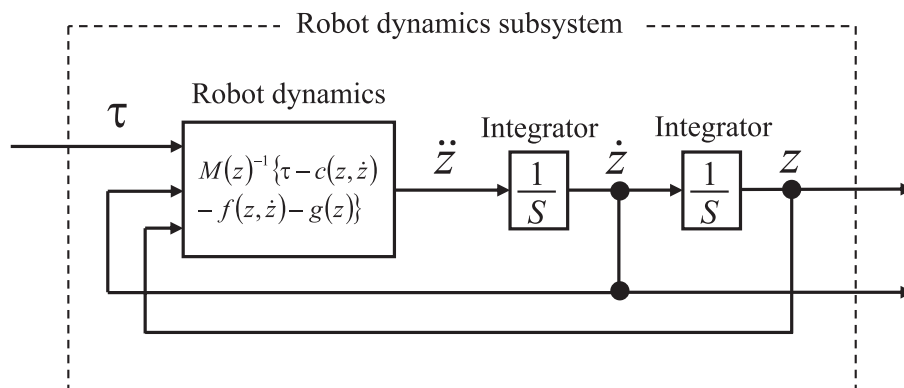


Fig. 1. Block diagram for subsystem of robot dynamics

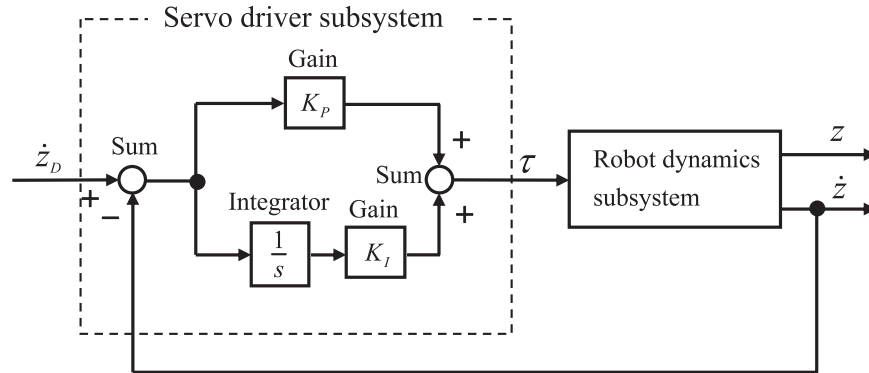


Fig. 2. Block diagram for velocity control servo system

- 1) 現時点の速度 \dot{z} がモデル部分からサーボドライバ部分に入力される。
- 2) 制御量(速度 \dot{z})と目標値の偏差を用いてPI制御則の演算が行われ、現時点の制御入力 τ が決定される。
- 3) 制御入力 τ がサーボドライバ部分からモデル部分に入力される。

結果と考察

特性解析シミュレーション

ここでは、前節で設計したモデル部分の動作確認と特性解析のために行ったシミュレーションの結果を示す。なお、水中ロボットの自然な動作をわかりやすくするために、サーボドライバ部分を取り除いている。シミュレーションのモデルは、前節で述べたように、2リンクマニ

ピュレータを有する水中ロボット(垂直型2リンク水中ロボットマニピュレータ)とし、文献6)で使用されたロボットの実機を参考にしてパラメータ(質量や長さなど)を設定した(Table 1 参照)。

マニピュレータの各関節に一定トルク($\tau_1 = 2.0 \times 10^{-1}$ [N·m], $\tau_2 = 5.0 \times 10^{-3}$ [N·m])を加えた場合のシミュレーション結果をFig. 3に示す。ただし、 τ_1 と τ_2 はそれぞれマニピュレータの第1, 2関節のトルクである。図の(a)は付加質量, 流体抗力, 重力・浮力を無視した場合のロボットの挙動, (b)は流体抗力, 重力・浮力を無視した場合のロボットの挙動, (c)は付加質量, 重力・浮力を無視した場合のロボットの挙動, (d)は重力・浮力を無視した場合のロボットの挙動である。すなわち、図の(a)は流体力が作用しない真空中のロボットの挙動, (b)は流体力として付加質量による力のみが作用するロボットの挙動,

Table 1. Parameters of 2-link underwater robot manipulator

	Base	Link 1	Link 2
Mass [kg]	28.32	4.25	1.23
Volume [$\times 10^{-3}$ m ³]	30.38	2.51	0.91
Moment of Inertia [kg m ²]	1.33	0.19	0.012
Length [m]	0.2 \times 0.81	0.25	0.25
Width [m]	0.42	0.12	0.12
Added mass (x) [kg]	72.7	1.31	0.1
Added mass (y) [kg]	6.28	3.57	2.83
Added Moment of Inertia [kg m ²]	1.05	0.11	0.06
Drag coefficient	1.2	1.2	1.2

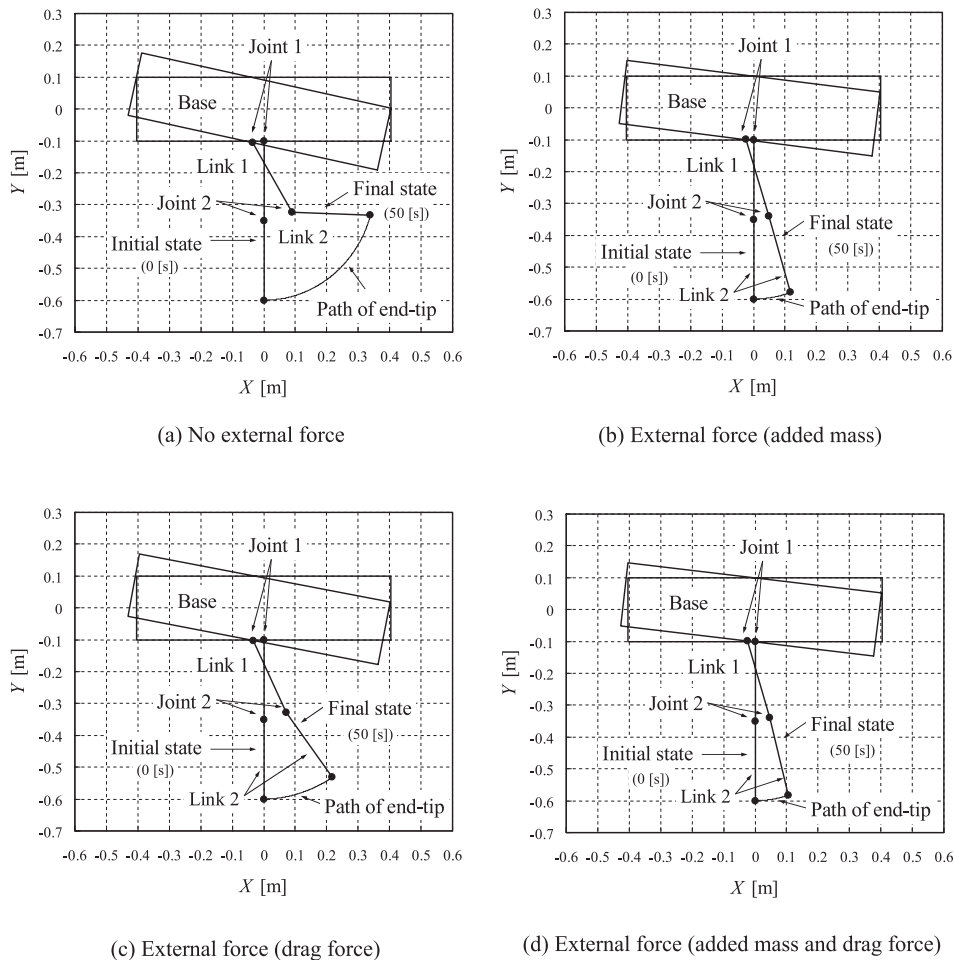


Fig. 3. Motion of underwater robot manipulator

(c)は流体力として流体抗力のみが作用するロボットの挙動、(d)は(b)と(c)の流体力が同時に作用するロボットの挙動である。なお、流体の影響を理解しやすくするために、すべての場合において、重力・浮力の影響を無視した。図の(a)と(b)より、付加質量の力がロボットの動作を抑制していることがわかる。また、図の(a)と(c)より、流体抗力も付加質量と同様な作用があることがわかる。一方、流体抗力よりも付加質量の力の方が動作に影響を与えていることが図の(b)と(c)から読み取れる。ただし、流体抗力は速度の2乗に比例する力であり、ロボットの動作が速いほど大きくなる。したがって、この場合よりも大きな関節トルクを加え、マニピュレータを素早く動かせば、流体抗力の抑制がより大きくなると考えられる。最後に、図の(d)は付加質量と流体抗力の両方を作用させている結果であるが、これと図の(b)を比較すれば、両者の挙動に明確な差は現れていない。すなわち、付加質量のみの場合と、付加質量と流体抗力の両方の場合が同様な結果となっ

ている。この原因として、付加質量の影響によりマニピュレータの動作が遅くなったために速度の2乗に比例する流体抗力の作用が抑えられていることが考えられる。

このシミュレーションにより、水中ロボットの挙動において流体力の影響を無視することは適切でなく、流体力を考慮した水中ロボット特有の制御手法の開発が必要であることがわかる。

サーボドライバの制御シミュレーション

速度入力型サーボドライバの制御性能を調べるために行ったシミュレーションの結果をFig. 4に示す。なお、過渡応答性能ではなく定常応答性能を調べるために、図の横軸を長時間に設定している。図において制御開始時から目標値に一致しているように読み取れるのは、このためであることに注意いただきたい。このシミュレーションでは、マニピュレータ関節のサーボドライバへの速度入力(速度の目標値)は両関節ともに 1.75×10^{-2} [rad/s] (約1 [deg/s])

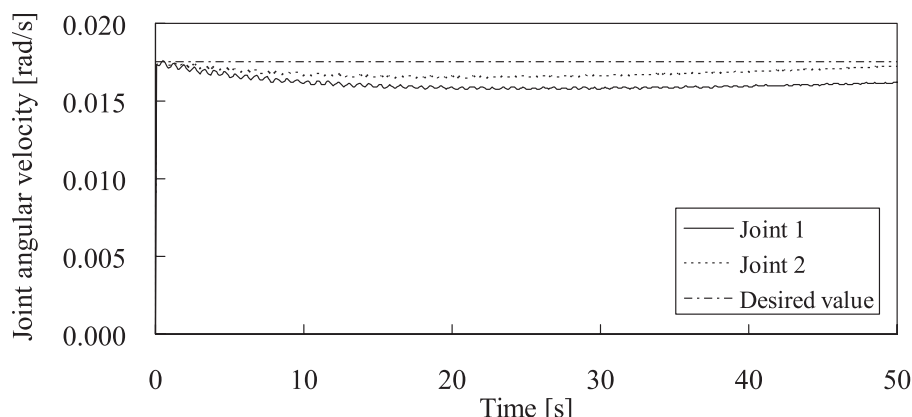


Fig. 4. Control performance of servo drivers

とした。したがって、両関節角速度が目標値 1.75×10^{-2} [rad/s]に一致しているほどサーボドライバの制御性能がよいといえる。そこで、図を観察すれば、それなりの制御性能は得られているが、良好な制御結果とまではいえないことがわかる。通常、水中ではなく陸上のマニピュレータでは、このPI制御式サーボドライバで十分な制御性能が得られるので、この形式のサーボドライバが普及していると考えられる。逆に、水中ロボットの場合、流体の影響のためにこのような簡単な形式の（市販されている）サーボドライバでは適切でないことを示唆しているともいえる。すなわち、サーボドライバを含めて制御を行う部分は、流体の影響を考慮したものでなければならないことがわかる。

結 言

本報では、MATLABを用いた水中ロボットマニピュレータのシミュレーションシステムを構築した。さらに、構築したシステムを用いて特性解析シミュレーションならびにサーボドライバの制御シミュレーションを行い、水中ロボット制御法の開発において流体力の影響を無視できないことがわかった。今後、本報で構築したシミュレーションシステムを用いて、流体の影響を考慮した高性能なマニピュレータ制御法の開発を行っていく予定である。なお、シミュレーションシステム開発の際に用いた水中ロボットの運動方程式は、水中ロボットのモデル化・制御分野においてよく用いられているものであるという点からすれば、このシステムはそれなりの信頼性があると考えられる。ただし、実機を用いた実験結果と比較することは有益であるので、今後検討していきたい。

文 献

- 1) 浦環, 高川真一: 海中ロボット. 成山堂 (1997)
- 2) 浦環: 海中に求められるロボット. 日本ロボット学会誌, 22, 692-696 (2004)
- 3) McLain TW, Rock SM, Lee MJ: Experiments in the Coordinated Control of an Underwater Arm/Vehicle System. *Autonomous Robots*, 3, 213-232 (1996)
- 4) 坂上憲光, 川村貞夫: 時間軸変換を利用した水中ロボットマニピュレータのフィードフォワード入力生成. 日本ロボット学会誌, 21, 562-568 (2003)
- 5) Sagara S, Tanikawa T, Tamura M, Katoh R: Experiments on a Floating Underwater Robot with a Two-Link Manipulator. *Artificial Life and Robotics*, 5, 215-219 (2001)
- 6) Sagara S, Shibuya K, Tamura M: Experiment of Digital RAC for an Underwater Robot with Vertical Planar 2-Link Manipulator. *Proc. of the Ninth Int. Symp. on Artificial Life and Robotics*, 337-340 (2004)
- 7) 平雄一郎, 森元映治, 横田源弘, 中村誠: ニューラルネットワークを用いた水中ロボットマニピュレータのシミュレーション. 水大校研報, 53, 71-78 (2005)
- 8) Using MATLAB Version 6 (CD-ROM). MathWorks (2002)
- 9) Using Simulink Version 5 (CD-ROM). MathWorks (2002)
- 10) Antonelli G: *Underwater Robots Motion and Force Control of Vehicle-Manipulator Systems*. Springer-Verlag (2003)