Motion Control for Adaptation to Human Environment

2005

Toshiaki Tsuji

Contents

Contents for Figures

Contents for Tables

Chapter 1

Introduction

1.1 Background of research on robotics — the North Pole is way far away —

Development of civilization has always been with that of technology. Many inventions brought the development. The first important inventions of mankind are supposed to be stone tools, fire lighting and words. It is not too much to say that these inventions humanized ancient people. From ancient days onward, mankind has invented important things such as paper, gunpowder, printing, and compass. These inventions had strong impacts on human society and triggered quantum leaps for civilization. A steam engine developed by James Watt in 1765 had one of the greatest impacts on human society. After the steam engine had been put to practical use, majority of human labor was replaced by machines. This event is called industrial revolution. The way of human society changed a lot with it. Many inventions have been made after the steam engine: telephone, electric lamp, gasoline engine, airplane, nylon, rocket, computer. It seems that evolution of technology is accelerating more and more. Furthermore, still a lot of researchers and engineers are trying to invent new things. This indicates human longing for sophisticated tools. Great inventors have always been dreaming about something sophisticated.

The longing also shows up in fiction. Science fiction (Sci-Fi) has become popular since the end of 19th century. Jules Verne and H. G. Wells explored the future possibilities opened by modern technology and made the large stream of Sci-Fi. The first idea of a rocket appeared in "Le Voyage Dans La Lune(From the Earth to the Moon)" [1] of Verne. Many robots also appeared in Sci-Fi novels. "L'Eve Future(Tomorrow's Eve)" [2], written by Villiers de l'lsle-Adam in 1886, described an android, a humanlike robot. Karel Capek, a famous Czech writer, created the word "robot" in his drama "R. U. R." [3]. A huge number of Sci-Fis about robots have been published

after l'lsle-Adam and Capek. The first antetype of a robot is, however, viewed as the "creature" in "Frankenstein; or, the Modern Prometheus" [4] written by Mary Shelley. The creature feels animosity toward Baron Frankenstein, who created him as an unwanted ugly being. At the end of the story, the creature is last seen on his way to the North Pole to destroy himself [4]. According to many pedestrian reviewers, the epilogue implies there are some sacred areas mankind should not enter into. Although the story was first published in 1818, way before the emergence of Sci-Fi, it raises a fundamental issue for scientists. The issue is a mental conflict between curiosity to scientific discovery and fear to technology transcending nature. It governs much of today's debate about the onrushing new age of technology, especially of biotechnology [5].

However, the evolution of technology accelerated and many inventions influenced by Sci-Fi were performed despite of the conflict. It is a well-known fact that R. H. Goddard, the father of modern rocket propulsion, was a dedicated reader of Verne and Wells' novel. Mankind finally arrived at the moon in 1969 thanks to Goddard and other scientists. Many sophisticated tools such as TV, mobile phone, and personal computer were invented after they appeared in Sci-Fi. The evolution indicates that real world is approaching the otherworld Sci-Fi writers dreamed of.

By the way, how about robots? Many robots have been developed for practical use. An American company Unimation put a playback robot into practice in 1960. Today, 770 thousand robots are working in the world for assembling, welding, and so on in factories [6]. And now, are the robots approaching to something Sci-Fi writers dreamed of? At first, most of robots in practice are not humanoids. Furthermore, although many humanoid robots have been developed after Honda "P2" realized natural walking motion [7], they are still completely different from the robots in Sci-Fi. They do not have a spirit and they do not speak. But the problem here is not about these kinds of difference because it is not a prerequisite for robots to be humanlike. The essential problem of current robots is a serious shortage of adaptability to human environment.

In this study, the word "environment" stands for things which are surrounding a robot. A wall, a floor, an obstacle, and all objects that may come into contact with the robot are parts of environment. The word "human environment" means the environment also includes human.

Robots do not have flexibility to work in unknown complicated environment contrary to strong demands of robots to work coordinate with human. Unfortunately, the technology is behind prediction in Sci-Fi when it comes to robotics. The North Pole is still way far away. It

doesn't mean that previous researches on robotics were unsatisfactory but it preferably shows profundity of the robotics field.

Since the study of robots in human environment deals with a profound problem, the study may have a spillover effect on other research areas such as artificial intelligence, human interface, and so on. The author believes that many principles, unknown but valuable for mankind, are buried in the problem.

Furthermore, the problem is a key for progress of the robotics industry. Although the robotics industry is expected to grow enormously in the near future [6], the growth is based on an assumption that robots can work in human environment. The problem must be solved to develop robots for welfare, medical service and disaster relief. In sum, this thesis aims at advancing adaptability of robots to human environment. The final goal of this study is to establish a robot control system that supports human in real environment.

1.2 Important researches for Adaptation to Human Environment

A wide array of approaches exists for autonomous robots adapting to human environment. According to [8], there are three minimum-required abilities for autonomous robots: environment recognition, motion planning, and motor control. The following description shows important researches for the three.

Environment recognition is indispensable for robots cooperating with human since human environment keeps changing its state. There have been many researches working on geometric modeling of the environment from sensory data [9]. Some of them used a laser range finder [10] or an ultrasonic sensor [11]. Stereo vision is a passive ranging method and is important in many situations where active ranging methods are not feasible [12]. This study assumes that geometric information in known environment is given by remote sensors of some kind although the robots do not mount any of them in fact. We do not discuss about sensing methods for the remote sensors since many prior arts exist. On the other hand, there are limited numbers of researches for environment recognition based on force sensing [13–16]. Environment recognition based on force information has a strong advantage to acquire not only a geometric model but also stiffness of the environment [14]. It also has some difficulties: an environment model is unknown until

the moment of contact; only active ranging is available; and force response depends on control of the robot. One of the focuses in this study is how to acquire environmental information during contact motion. This issue should be discussed together with motion planning.

Motion planning is a governing issue for environmental adaptation. Here, motion planning is an issue to decide the trajectory of a robot in response to variation of situations. Its studies have mainly been performed on path planning of a wheeled mobile robot in complicated environment [17–19]. Majority of these methods applied artificial intelligence. A biped robot is also a good candidate to assess validity of a motion planning method since improper motion planning causes instability at once. A number of studies have already dealt with path planning of a biped robot in complicated environment [20–22]. Motion planning on multi-robot systems was a popular topic and an issue on dealing with interaction between multiple robots was attracting interests [23–25]. Some of them expect that the interaction may generate an emergent intelligence [26]. Although motion planning is an old issue in robotics, a breakthrough is still expected for establishing a self-organized robot system.

Motor control is the most fundamental and mature topic of the three. Substantial issues are already solved by previous studies. Among them, force control in real-world environment still remains some problems. Although an adaptability of a robot in human environment can be increased by force control [27], it is a problem that force control may deteriorate tracking performance of position control. Hybrid control is an effective method to solve the problem while it requires acceleration control [28] for acquiring its ideal response. This study therefore focuses on how to implement an ideal acceleration control system to robots in human environment.

1.3 Way of approach in this study

It is a tough road for robots to acquire adaptability to human environment. The main problem is how to solve interaction between a robot and human environment. Interaction among robot, human and environment is schematized as shown in Fig. 1-1.

Here, human manipulation means that a robot complies with manipulation force from human. Most of robots in practice are only accomplished at this ability. Autonomous adaptation means the robot autonomously quarries environmental information and adapts to varying environment. The present robots still need improvement to acquire this ability. Sensation sharing means that

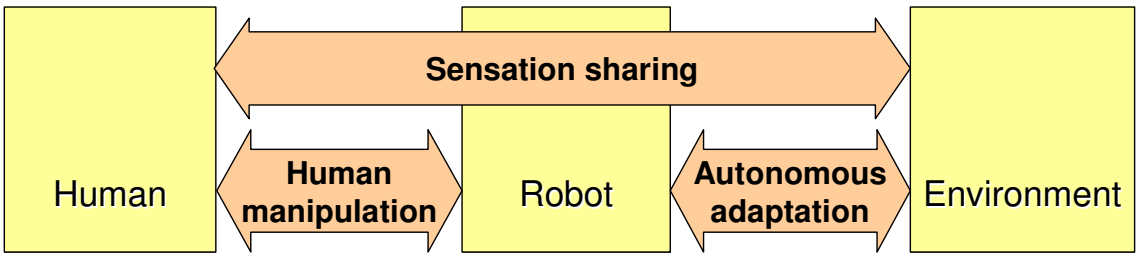


Fig.1-1: Schematic diagram of robot for human assist

human acquires environmental information through the robot. A robot reproducing haptic information of remote environment for human is equivalent to a robot sharing sensation with human. Bilateral control is a key technique for sensation sharing.

A robot in human environment has to deal with these interactions. There are three issues that have to be solved. Firstly, a robot should improve the ability of autonomous adaptation to environment. Secondly, a flexible design framework to decouple and deal with the complicated interactions is required. Thirdly, control performance should be further improved to approximate the control system to the theoretical model. Coordinated with these issues, the author categorizes the themes into three parts: sophisticated motion planning, flexible controller design and robust motion control, as shown in Fig. 1-2. This study aims at advancing adaptability of robots in human environment by dealing with each factor one by one.

This paper discusses motion planning of a robot in Chapter 2. Motion planning has attracted great interests since it is one of the most dominant elements for autonomous adaptation to environment. Most of studies on motion planning dealt with kinematic trajectory planning while hard real-time control [29] considering dynamic interaction with environment is requisite for robots in human environment. Hence, motion planning in this study is discussed together with acquisition of environmental information based on force sensing. A biped robot is an effective tool to verify such a method since it requires both elements: proper motion planning and force sensing with hard real-time control. The biped robot will lose its walking stability if either element is missed. In sum, Chapter 2 discusses trajectory planning of a biped robot in view of motion planning for environmental adaptation.

In the next place, flexibility of controller design is discussed in Chapter 3 and Chapter 4. The

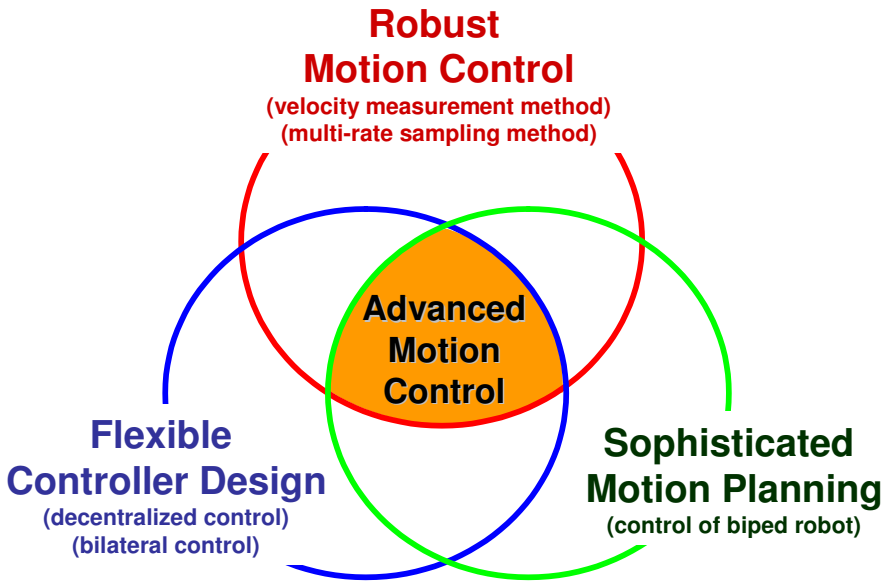


Fig.1-2: Categories for environmental adaptation

study on a biped robot shows that hybrid control based on modal decomposition is effective for adaptation to unknown environment. The next problem is complexity of controller design by several reasons as follows: firstly, human environment is so complicated that the robot needs a hyper-DOF mechanism for adaptation; secondly, a state of environment keeps changing in practice; finally, a task for a sophisticated control system may also keep changing. Robots in human environment are required to keep changing and modifying their tasks due to variation of the environment. The robots are also required to deal with many kinds of exceptions such as performance limits of the robot and faults. Hence, an idea of functionality is introduced to solve the complexity of controller design. A framework of controller design based on functionality is simple and explicit. Furthermore, it provides flexibility of design for a large scale system to deal with task alteration and environment alteration in a unified manner. Chapter 3 proposes the concept of functionality and verifies the proposal with a bilateral control system. Chapter 4 extends the framework to decentralized control systems.

The design framework in this study simplifies development of a hybrid control system with multi input and multi output. Although decoupled subsystems may have interference to each other in practice, independence of subsystems is preserved by applying acceleration control

based on disturbance observer. This fact implies that acceleration control is indispensable for development of a robust hybrid control system which is a mixture of position control and force control. Hence Chapter 5 discusses robustness of acceleration control, which is one of the most fundamental issues in motion control. Here, the word "robust" may sometimes give an impression of a stiff and hazardous system. Robots have such an image because many of them have treated external force as disturbance. By compensating external force as disturbance, i.e. being robust against external force, they become stiff and hazardous. It is a well-known fact, however, that a robot can adjust its compliance by applying force control [28]. In such a case, accuracy and a bandwidth of acceleration control determine a limit of adaptability to environment. Therefore improvement of robustness in acceleration control is a fundamental requirement for environment adaptation of a robot. A new velocity measurement method for acceleration control is proposed and improvement of robustness of acceleration control with the method is shown.

In the last place, this paper is summarized in Chapter 6.

Each chapter has a section for introduction to show other related studies. It also shows motivation of the study for adaptation to human environment.

Chapter 2

Motion planning for adaptation to environment

2.1 Background to study biped robots

This chapter discusses motion planning of a biped robot in view of environmental adaptation.

Biped robots are studied in many institutes and some of their demonstrations are well publicized. "ASIMO" [30] from Honda Motor Co. Ltd., "QRIO" [31] from Sony Corp., "HRP-2" [32] from AIST in Japan, and TOYOTA's "i-foot" [33] are well-known examples.

As shown above, Japanese companies lead the study of biped robots while the study is also active in other countries. Leg Laboratory in MIT developed a lot of walking or hopping robots [34–36]. Among them, the study of Raibert is well known as a pioneer of hopping robots [34]. McGeer [37] has shown a passive dynamic walking, a walk without active control or energy input. This study attracts interests since it provides an opportunity to think about a physical principle of walking motion. "Johnnie" from Technical University of Munich showed very good performance of biped locomotion [38].

All of these robots have their distinctive methodology and machinery. The first method of dynamic walking with high performance is "ZMP control" [7] performed by Honda "P2", the prototype of ASIMO. Zero moment point (ZMP), proposed by Vukobratovic [39], is an indicator to show stability of dynamic walking. ZMP control is a method that accelerates the upper body of the biped robot so that ZMP converges to its stability region. P2 was in the world limelight due to its experimental demonstration of dynamic walking based on the ZMP control.

Inverted pendulum mode is also a widely-applied method for trajectory planning that acquires walking stability with small calculation amount [40, 41]. This study is based on linier inverted pendulum mode (LIPM) [42], proposed by Kajita et. al. Mitobe et. al. proposed a method combines ZMP control and inverted pendulum mode [43]. Although inverted pendulum mode

is useful for a trajectory planning in real time, the walking motion is limited to the trajectory based on pendulum dynamics.

Minakata et. al. proposed a method to accelerate the robot by shortening length of the virtual pendulum model [44]. Sugihara et. al. proposed ZMP manipulation, a method to modify a walking trajectory by manipulating the ZMP position [45]. The ZMP is projected to a virtual horizontal plane in order to extend this method to unleveled ground. Pendulum length modified in [44] determines the time constant of biped locomotion while ZMP in [45] determines acceleration on the robot. In sum, these methods improve flexibility of trajectory planning based on inverted pendulum mode.

On the other hand, the author has proposed a method to extend the idea of virtual pendulum model [46]. The method is to manipulate the supporting point of the pendulum. The point is named virtual supporting point (VSP) and it is applied as an index for biped walking. By applying VSP as an index, both the time constant of biped locomotion and the acceleration on biped robot are modified in a unified manner. Furthermore, this method can plan a stable trajectory prospectively since a stability region of VSP is derived in a theoretical sense. These characteristics of the method would be a fundamental technique for biped robots in human environment. In Section 2.3, the method is explained in detail.

One of the important issues for the biped robot is that disturbance may easily lead to instability. Particularly, large disturbance may occur due to collision. Compare to other fixed manipulators, the biped robot has a limitation for disturbance rejection since large input torque on ankle joint may cause tipping.

Modification of stepping point is very effective to deal with this kind of disturbance. Therefore a method to modify the stepping point against collision is proposed in Section 2.4. According to the method, the robot will not counter the disturbance but plan a new trajectory based on dynamics of the pendulum if large disturbance occurs to the robot. A relationship between the stepping point and the trajectory is figured out and the disturbance is compensated by controlling the stepping point. Experimental results show that the method is indispensable for robots in human environment since walking stability under possibility of collision is approved by the method.

The author selected a biped robot as a tool to verify motion planning in human environment

since walking robot has mechanical advantage to adapt to human environment. The major advantage is that the robot can select its surface of ground contact discretely. The robot can step over obstacles or stairs with this advantage. Hence obstacle avoidance will be a key technique for biped robots.

One technique is to investigate some stable walking patterns and decide one pattern from the conditional branch in each step [47]. Beyond that, there is a global planning method which does not suffer from deadlock or local loops [20]. Dunn and Howe proposed a method to adjust either position of a stepping point or walking speed [48]. Kajita and Tani realized obstacle avoidance in dynamic walking [42].

Section 2.5 describes a method to select a series of footholds for obstacle avoidance. The model of obstacle is derived to simplify the algorithm of collision prediction. An idea of stamp area is introduced to avoid large stride alteration. Manipulation of VSP realizes sudden alteration of stride.

Section 2.6 discusses biped locomotion in unknown environment. Although trajectory planning is a key for the adaptation, there are some situations that it is impossible to acquire the adaptation only by trajectory planning. For example, strong collision with the ground may occur when a gradient of the ground is unknown. Nishikawa et al. have shown effectiveness of compliance control in such a case [49]. Although compliance control improves adaptability to unknown environment, it may also degrade tracking performance of the robot. An idea of an environmental mode is therefore introduced to acquire both environmental adaptability and tracking performance. Walking on unleveled ground is achieved by a hybrid control system based on environmental modes.

Finally, this chapter is summarized in Section 2.7.

2.2 Controller and mechanism of biped robot

2.2.1 Modeling of biped robot

Fig. 2-1 shows the biped robot used in this study. It is a 3-dimensional biped robot with 6 DOF in the sagittal plane and 4 DOF in the frontal plane. A model of the robot is shown in Fig. 2-2. Here, x-coordinate denotes the lateral direction, y-coordinate denotes the traveling direction and z-coordinate denotes the vertical direction. The robot is assumed as a mass system

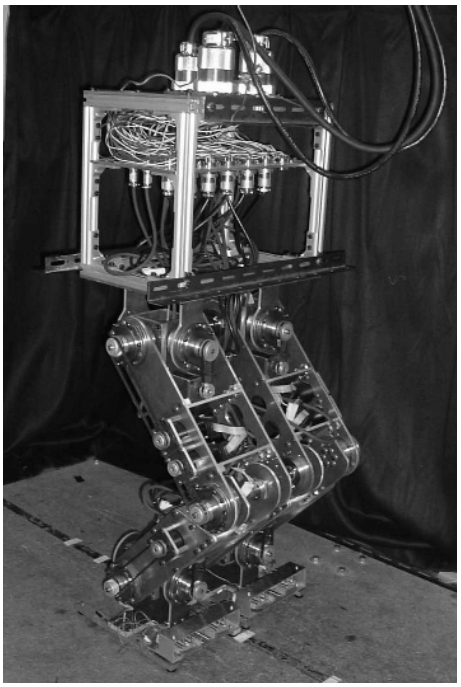


Fig.2-1: Overview of biped robot

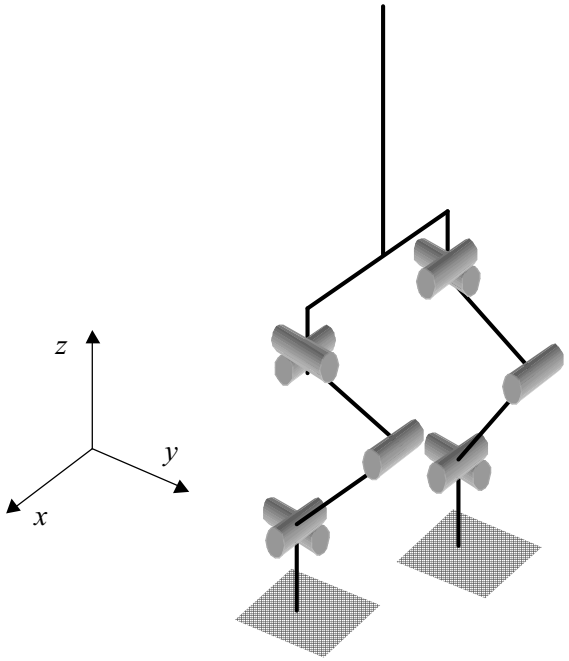


Fig.2-2: Model of biped robot

which has its mass on the COG of each link. The measured parameters of the robot are shown in Table 2.1.

Its dynamics is calculated on (2.1) [50]. The trunk link is treated as the base link of the robot here.

$$\begin{bmatrix} \boldsymbol{H}_b & \boldsymbol{H}_{bm} \\ \boldsymbol{H}_{bm}^T & \boldsymbol{H}_m \end{bmatrix} \begin{bmatrix} \ddot{\boldsymbol{x}}_b \\ \ddot{\boldsymbol{\phi}} \end{bmatrix} + \begin{bmatrix} \boldsymbol{c}_b \\ \boldsymbol{c}_m \end{bmatrix} = \begin{bmatrix} \boldsymbol{F}_b \\ \boldsymbol{\tau} \end{bmatrix} + \begin{bmatrix} \boldsymbol{J}_b^T \\ \boldsymbol{J}_m^T \end{bmatrix} \boldsymbol{F}_h \tag{2.1}$$

where $\boldsymbol{H}_b$, $\boldsymbol{H}_{bm}$, $\boldsymbol{H}_m$ are inertia matrices, $\boldsymbol{x}_b$ is an absolute position of the base link, $\boldsymbol{c}_b$, $\boldsymbol{c}_m$ are non-linear terms, $\boldsymbol{\tau}$ denotes torque on joints, $\boldsymbol{\phi}$ denotes angle of joints, $\boldsymbol{J}_b$, $\boldsymbol{J}_m$ are Jacobean matrices for the base link and joints respectively, $\boldsymbol{F}_b$, $\boldsymbol{F}_h$ are force to the base link and an end effector.

3 DOF of swing leg tip position, 2 DOF of swing leg tip attitude, 3 DOF of COG position, 2 DOF of trunk attitude are controlled respectively. The attitude of the base link and that of swing leg tip have only 2 DOF despite the attitude in a 3-dimensional space should have 3 DOF. The robot in this study does not have a degree of freedom for rotation around z-axis.

Table 2.1: Mechanical parameters of biped robot

	Mass [kg]	Length [m]	COG from upper joint [m]
Trunk	15.0	0.40	0.200
Waist joint link	2.6	0.00	-0.020
Upper link of leg	3.5	0.30	0.126
Lower link of leg	3.4	0.30	0.095
Ankle joint link	2.6	0.00	-0.020
Foot link	2.0	0.12	0.023

2.2.2 Linear Inverted Pendulum Mode

Real-time trajectory planning is indispensable for a biped robot in human environment while it is difficult to exactly figure out a trajectory for stable walking. Linear inverted pendulum mode (LIPM) [42] is applied to figure out the trajectory in real time.

The robot is modeled as an inverted pendulum as shown in Fig. 2-3. The pendulum has a mass point on COG of the robot. The supporting point of the pendulum is set on the ankle joint of the support leg in conventional methods. The trajectory is controlled so that the height of the pendulum becomes constant. At the same time, the trajectory is controlled so that no torque occurs on the supporting point.

COG should accelerate only on the horizontal direction since the height of the pendulum is constant. The acceleration is figured out from dynamics of the pendulum by (2.2).

$$\ddot{\boldsymbol{r}} = \frac{g}{h}\boldsymbol{r} \tag{2.2}$$

where $\boldsymbol{r}$ is a 2-dimensional vector on a horizontal plane that denotes position of COG based on the supporting point, g is the gravity acceleration, and h is the height of the pendulum.

A trajectory of the pendulum is figured out uniquely by (2.3) and (2.4), developed forms of (2.2).

$$\boldsymbol{r} = \frac{\boldsymbol{r}_0 - \sqrt{\frac{h}{g}}\dot{\boldsymbol{r}}_0}{2} e^{-\sqrt{\frac{g}{h}}t} + \frac{\boldsymbol{r}_0 + \sqrt{\frac{h}{g}}\dot{\boldsymbol{r}}_0}{2} e^{\sqrt{\frac{g}{h}}t} \tag{2.3}$$

Table 2.2: Details on robot, motors and gears

Robot	Entire length	122[cm]
	Entire weight	38.4[kg]
	DOF	10
Motor	Type	DC servo motor
	Maker	maxon motor
	Model number	REϕ40
	Rated power	150[W]
	Nominal voltage	48[V]
	Stall torque	2500[mNm]
	Max. continuous current	3330[mA]
	Max. continuous torque	201[mNm]
	Torque constant	60.3[mNm/A]
	Moment of inertia	134[gcm^2]
	Resolution of encoder	500[pulse/rev]
Gear	Maker	Harmonic drive systems Inc.
	Model number	CSF-17-100
	Reduction ratio	100:1
	Moment of inertia	0.079[$\times 10^{-4}$kgm^2]
	Max. acceptable torque	110[Nm]
	Max. acceptable input revolution	3500[rev/min]

$$\dot{\boldsymbol{r}} = \frac{\dot{\boldsymbol{r}}_0 - \sqrt{\frac{g}{h}}\boldsymbol{r}_0}{2} e^{-\sqrt{\frac{g}{h}}t} + \frac{\dot{\boldsymbol{r}}_0 + \sqrt{\frac{g}{h}}\boldsymbol{r}_0}{2} e^{\sqrt{\frac{g}{h}}t} \tag{2.4}$$

here, t is time and $\boldsymbol{r}_0$ is the initial value of $\boldsymbol{r}$ i.e. $\boldsymbol{r}$ at $t = 0$.

If the robot follows the trajectory, the support foot surface keeps stable contact with the ground because no torque should occur on the supporting point.

Essentially, inverted pendulum is a statically unstable system. Biped locomotion can be sustained by repeating steps. The trajectory switches after the step due to a changeover of the

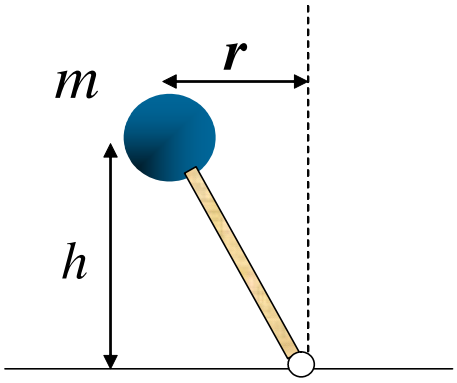


Fig.2-3: Inverted pendulum

supporting point. The definition of the support leg and the swing leg also changes at the moment. It is named "supporting-point changeover" and it is executed during the double support phase. Position and velocity of the pendulum at the supporting-point changeover are derived by (2.5) and (2.6), respectively.

$$\boldsymbol{r}_{ch-} = \frac{\boldsymbol{r}_0 - \sqrt{\frac{h}{g}}\dot{\boldsymbol{r}}_0}{2} e^{-\sqrt{\frac{g}{h}}t_{ch}} + \frac{\boldsymbol{r}_0 + \sqrt{\frac{h}{g}}\dot{\boldsymbol{r}}_0}{2} e^{\sqrt{\frac{g}{h}}t_{ch}} \tag{2.5}$$

$$\dot{\boldsymbol{r}}_{ch-} = \frac{\dot{\boldsymbol{r}}_0 - \sqrt{\frac{g}{h}}\boldsymbol{r}_0}{2} e^{-\sqrt{\frac{g}{h}}t_{ch}} + \frac{\dot{\boldsymbol{r}}_0 + \sqrt{\frac{g}{h}}\boldsymbol{r}_0}{2} e^{\sqrt{\frac{g}{h}}t_{ch}} \tag{2.6}$$

here, t_{ch} denotes a time of the supporting-point changeover. $\boldsymbol{r}_{ch-}$ denotes an estimated position of the COG at t_{ch}.

Since $\boldsymbol{r}$ is a relative position of the COG based on the supporting point, its value alters when the supporting-point changeover occurs. $\boldsymbol{r}_{ch+}$, estimate position of the COG just after the changeover is figured out as follows:

$$\boldsymbol{x}_{ch} = \boldsymbol{r}_{ch-} + \boldsymbol{l}_0 \tag{2.7}$$

$$\boldsymbol{r}_{ch+} = \boldsymbol{x}_{ch} - \boldsymbol{l}_{ch} \tag{2.8}$$

$$\dot{\boldsymbol{r}}_{ch+} = \dot{\boldsymbol{r}}_{ch-}. \tag{2.9}$$

Here, $\boldsymbol{x}_{ch}$ is an absolute position of the COG at $t = t_{ch}$, $\boldsymbol{l}_0$ is an absolute position of the present stepping point, $\boldsymbol{l}_{ch}$ is an absolute position of the next stepping point.

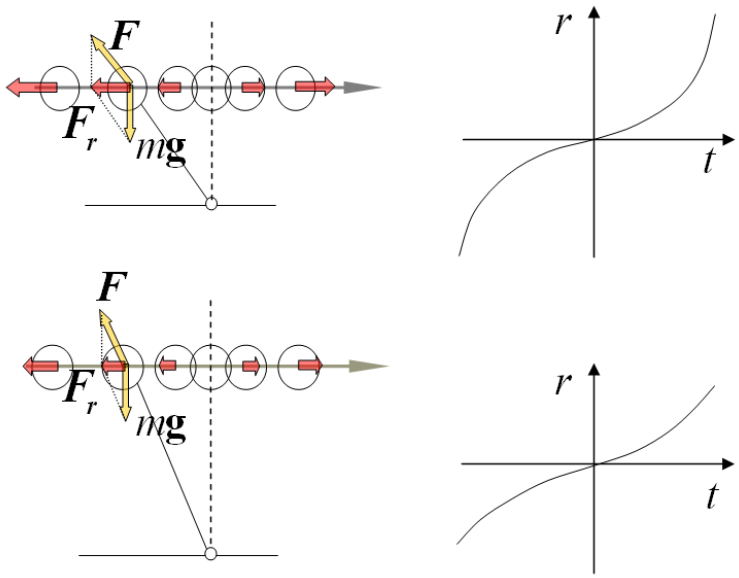


Fig.2-4: Length and acceleration

By extending the equations, a future trajectory after the supporting-point changeover is figured out by (2.10) and (2.11).

$$\boldsymbol{r}_n = \frac{\boldsymbol{r}_{ch+} - \sqrt{\frac{h}{g}}\dot{\boldsymbol{r}}_{ch+}}{2} e^{-\sqrt{\frac{g}{h}}(t-t_{ch})} + \frac{\boldsymbol{r}_{ch+} + \sqrt{\frac{h}{g}}\dot{\boldsymbol{r}}_{ch+}}{2} e^{\sqrt{\frac{g}{h}}(t-t_{ch})} \tag{2.10}$$

$$\dot{\boldsymbol{r}}_n = \frac{\dot{\boldsymbol{r}}_{ch+} - \sqrt{\frac{g}{h}}\boldsymbol{r}_{ch+}}{2} e^{-\sqrt{\frac{g}{h}}(t-t_{ch})} + \frac{\dot{\boldsymbol{r}}_{ch+} + \sqrt{\frac{g}{h}}\boldsymbol{r}_{ch+}}{2} e^{\sqrt{\frac{g}{h}}(t-t_{ch})} \tag{2.11}$$

here, $\boldsymbol{r}_n$ is a position of the COG at time t.

By substituting $\boldsymbol{x}_n = \boldsymbol{r}_n + \boldsymbol{l}_{ch}$ and $\boldsymbol{r}_{ch+} = \boldsymbol{x}_{ch} - \boldsymbol{l}_{ch}$ into (2.10) and (2.11), a relationship between $\boldsymbol{x}_n$ and $\boldsymbol{l}_{ch}$ is derived as follows:

$$\boldsymbol{x}_n = \frac{\boldsymbol{x}_{ch} - \boldsymbol{l}_{ch} - \sqrt{\frac{h}{g}}\dot{\boldsymbol{x}}_{ch}}{2} e^{-\sqrt{\frac{g}{h}}(t-t_{ch})} + \frac{\boldsymbol{x}_{ch} - \boldsymbol{l}_{ch} + \sqrt{\frac{h}{g}}\dot{\boldsymbol{x}}_{ch}}{2} e^{\sqrt{\frac{g}{h}}(t-t_{ch})} + \boldsymbol{l}_{ch} \tag{2.12}$$

$$\dot{\boldsymbol{x}}_n = \frac{\dot{\boldsymbol{x}}_{ch} - \sqrt{\frac{g}{h}}(\boldsymbol{x}_{ch} - \boldsymbol{l}_{ch})}{2} e^{-\sqrt{\frac{g}{h}}(t-t_{ch})} + \frac{\dot{\boldsymbol{x}}_{ch} + \sqrt{\frac{g}{h}}(\boldsymbol{x}_{ch} - \boldsymbol{l}_{ch})}{2} e^{\sqrt{\frac{g}{h}}(t-t_{ch})} \tag{2.13}$$

here, it is assumed that $\dot{\boldsymbol{x}}_{ch} = \dot{\boldsymbol{r}}_{ch-}$ since $\dot{\boldsymbol{l}}_{ch} = \boldsymbol{0}$. $\boldsymbol{x}_n$ is an absolute position of the COG at time t, All the vectors in these equations are 2-dimensional vectors on a horizontal plane. Stepping point is defined as a point the swing leg will land. The representative point of the swing leg is the center point of the foot surface in this study.

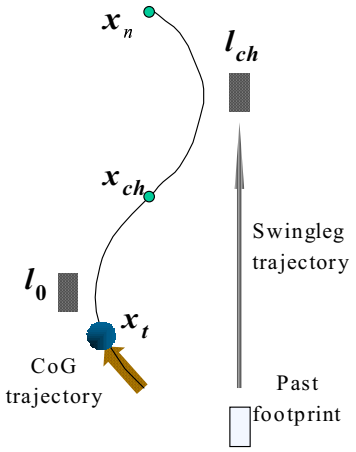


Fig.2-5: Trajectory of COG

Equations (2.12) and (2.13) show a future trajectory and it depends on a stepping point $\boldsymbol{l}_{ch}$ which is controlled from the trajectory of the swing leg. The proper $\boldsymbol{l}_{ch}$ which satisfies the desired $\boldsymbol{x}_n$ or $\dot{\boldsymbol{x}}_n$ is determined uniquely from (2.14) and (2.15).

$$\boldsymbol{l}_{ch} = \frac{-\sqrt{\frac{h}{g}}(e^{-\sqrt{\frac{g}{h}}(t-t_{ch})} - e^{\sqrt{\frac{g}{h}}(t-t_{ch})})\dot{\boldsymbol{x}}_{ch} - 2\boldsymbol{x}_n}{e^{-\sqrt{\frac{g}{h}}(t-t_{ch})} + e^{\sqrt{\frac{g}{h}}(t-t_{ch})}} + \boldsymbol{x}_{ch} \tag{2.14}$$

$$\boldsymbol{l}_{ch} = \frac{-\sqrt{\frac{h}{g}}(e^{-\sqrt{\frac{g}{h}}(t-t_{ch})} + e^{\sqrt{\frac{g}{h}}(t-t_{ch})})\dot{\boldsymbol{x}}_{ch} - 2\sqrt{\frac{h}{g}}\dot{\boldsymbol{x}}_n}{e^{-\sqrt{\frac{g}{h}}(t-t_{ch})} - e^{\sqrt{\frac{g}{h}}(t-t_{ch})}} + \boldsymbol{x}_{ch} \tag{2.15}$$

The robot can achieve the command position or the command velocity at the moment of the supporting-point changeover by controlling $\boldsymbol{l}_{ch}$. If it is presumed that x-coordinate of $\boldsymbol{x}_{ch}$ is constant in steady locomotion, the walking cycle T is figured out as follows:

$$T = -2\sqrt{\frac{h}{g}} ln \left(\frac{r_{0x} - \sqrt{\frac{h}{g}}\dot{r}_{0x}}{r_{0x} + \sqrt{\frac{h}{g}}\dot{r}_{0x}} \right) \tag{2.16}$$

where r_{0x} denotes the x-coordinate value of $\boldsymbol{r}$ at the moment the robot steps.

2.3 Biped locomotion with VSP

2.3.1 Concept of VSP

This section describes the idea of virtual supporting point(VSP), a new indicator of inverted pendulum mode locomotion, and explains its application.

As shown in (2.2), the acceleration to the COG inversely relates to the height of COG. In the inverted pendulum mode locomotion, it is desirable to have a higher pendulum model to acquire a smooth trajectory with less acceleration. The height of COG, however, is limited by mechanical structure of the robot.

Therefore a method that modifies the model of the inverted pendulum is proposed. The left figure of Fig. 2-6 denotes conventional methods. As shown in the figure, the supporting point is set on the ankle joint of the support leg which becomes free joint. In the proposed method, the supporting point is set somewhere else virtually instead of letting the ankle joint free. For example, the virtual supporting point could be set on the supporting surface or even under the ground, as shown in the center and right figure of Fig. 2-6. Here, supporting surface denotes a convex polygon consists of vertices at ground contact points. It is assumed that only corners of foot surface contact to ground.

Modification of VSP only affects the trajectory planning. The trajectory is planned so that no torque occurs on the supporting point of the pendulum.

By shifting the position of VSP, the height of the pendulum could be adjusted without changing the posture of the robot. This advantage could be taken to speed adjustive locomotion and footstep allocation. Furthermore, parameters of the inverted pendulum could be adjusted beyond the mechanical limit. For example, the height of the pendulum could be set higher than the mechanical limit as shown in Fig. 2-7.

Speed on dynamic walk always fluctuates due to dynamics of the pendulum. The fluctuation of speed reduces by heightening the model of the pendulum. Mobility improves with it since velocity fluctuation in high speed locomotion destabilizes walking motion in practice.

2.3.2 Relationship between VSP and ZMP

A relationship between VSP and ZMP is shown in this section so that stability of biped locomotion applying VSP could be figured out.

In the conventional method applying inverted pendulum mode, ZMP is the point which extension of pendulum and supporting surface cross. Similarly, ZMP is the point which extension of pendulum and supporting surface cross because no torque should occurs on the extension. A

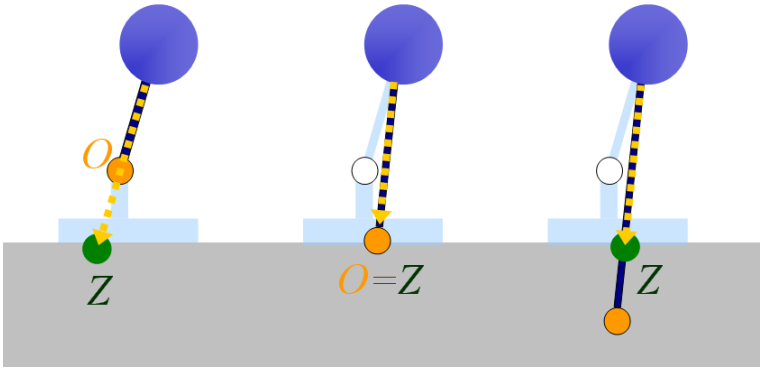


Fig.2-6: Relationship between VSP and ZMP

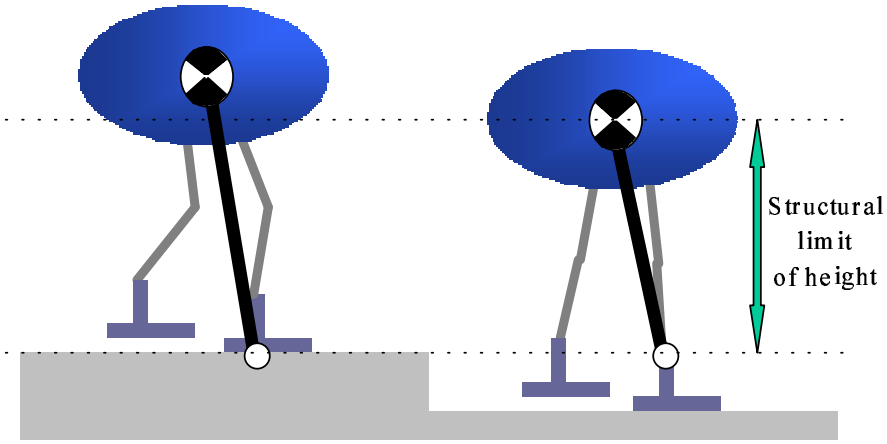


Fig.2-7: Mechanical limit

theoretical value of $\boldsymbol{Z}$, ZMP position on a horizontal plane is figured out as follows:

$$\boldsymbol{Z} = \frac{d}{h_g + d}\boldsymbol{r} \tag{2.17}$$

where h_g is height of COG and d is depth of VSP. $\boldsymbol{Z}$ is a relative ZMP position based on VSP.

The biped locomotion is most stable when VSP is set on the supporting surface because ZMP matches with VSP theoretically in such a case. This fact also supports the claim that mobility of the robot improves with VSP in a proper position. Fig. 2-8 shows an overhead view of foot surface. It shows that ZMP moves relative to COG motion.

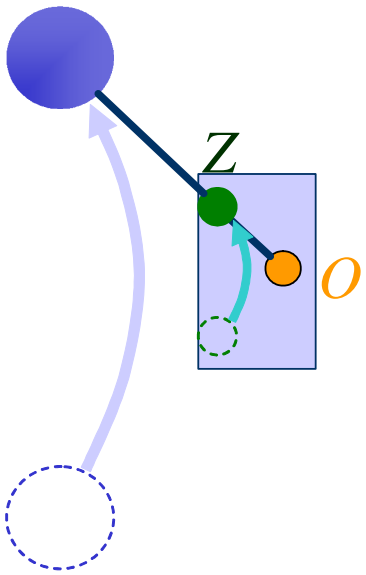


Fig.2-8: Movement of ZMP

ZMP is an indicator to approve grounding of the foot surface. Hence walking stability is assured temporarily if ZMP is in the supporting surface. To walk in safety, $\boldsymbol{r}$ should not go further than the line at which ZMP comes to the limit as shown in Fig. 2-9. In that sense, COG position is limited in a kind of a stable area depending on limitation of ZMP. The stable area of COG is figured out as follows:

$$if \quad d > 0$$

$$\frac{h_g+d}{d} Z_{xmin} \leq r_x \leq \frac{h_g+d}{d} Z_{xmax} \tag{2.18}$$

$$\frac{h_g+d}{d} Z_{ymin} \leq r_y \leq \frac{h_g+d}{d} Z_{ymax} \tag{2.19}$$

$$if \quad d < 0$$

$$-\frac{h_g+d}{d} Z_{xmax} \leq r_x \leq -\frac{h_g+d}{d} Z_{xmin} \tag{2.20}$$

$$-\frac{h_g+d}{d} Z_{ymax} \leq r_y \leq -\frac{h_g+d}{d} Z_{ymin} \tag{2.21}$$

where r_x is the x-axial element of $\boldsymbol{r}$, r_y is the y-axial element of $\boldsymbol{r}$, Z_{xmin} is the minimum limit of $\boldsymbol{Z}$ in x-axis, Z_{xmax} is the maximum limit of $\boldsymbol{Z}$ in x-axis, Z_{ymin} is the minimum limit of $\boldsymbol{Z}$ in y-axis, Z_{ymax} is the maximum limit of $\boldsymbol{Z}$ in y-axis.

On the other hand, if the step width of locomotion is decided as shown in Fig. 2-10, a stable

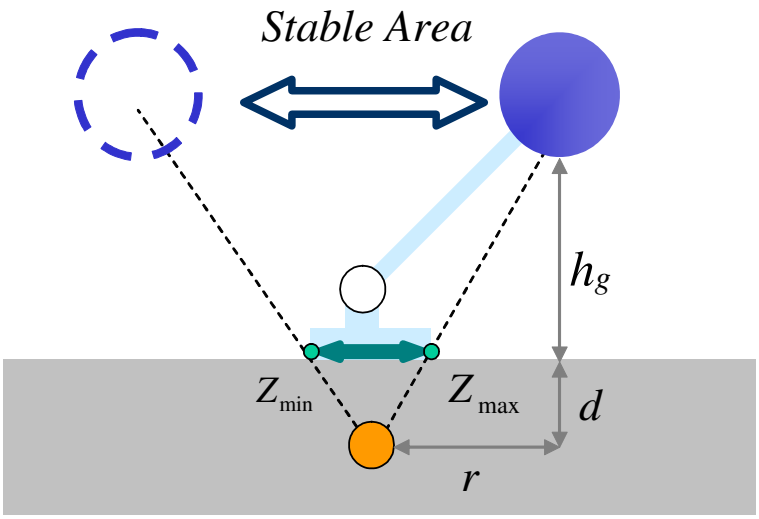


Fig.2-9: Stable area of COG

area of VSP for stable locomotion can be figured out. The stable area of VSP which considers only the motion in x-coordinate is figured out as follows:

$$d \geq \frac{h_g Z_{xmax}}{r_{xmin} - Z_{xmax}}, \quad if\ r_{xmin} < Z_{xmax} \tag{2.22}$$

$$d \leq \frac{h_g Z_{xmax}}{r_{xmin} - Z_{xmax}}, \quad if\ r_{xmin} > Z_{xmax} \tag{2.23}$$

$$d \geq \frac{h_g Z_{xmin}}{r_{xmin} - Z_{xmin}}, \quad if\ r_{xmin} > Z_{xmin} \tag{2.24}$$

$$d \leq \frac{h_g Z_{xmin}}{r_{xmin} - Z_{xmin}}, \quad if\ r_{xmin} < Z_{xmin} \tag{2.25}$$

$$d \geq \frac{h_g Z_{xmin}}{r_{xmax} - Z_{xmin}}, \quad if\ r_{xmax} > Z_{xmin} \tag{2.26}$$

$$d \leq \frac{h_g Z_{xmin}}{r_{xmax} - Z_{xmin}}, \quad if\ r_{xmax} < Z_{xmin} \tag{2.27}$$

$$d \geq \frac{h_g Z_{xmax}}{r_{xmax} - Z_{xmax}}, \quad if\ r_{xmax} < Z_{xmax} \tag{2.28}$$

$$d \leq \frac{h_g Z_{xmax}}{r_{xmax} - Z_{xmax}}, \quad if\ r_{xmax} > Z_{xmax}. \tag{2.29}$$

The stable area which considers the motion in y-coordinate can be figured out similarly.

As shown in Fig. 2-8, ZMP moves forward in each step. Essentially, it has equivalent effect to

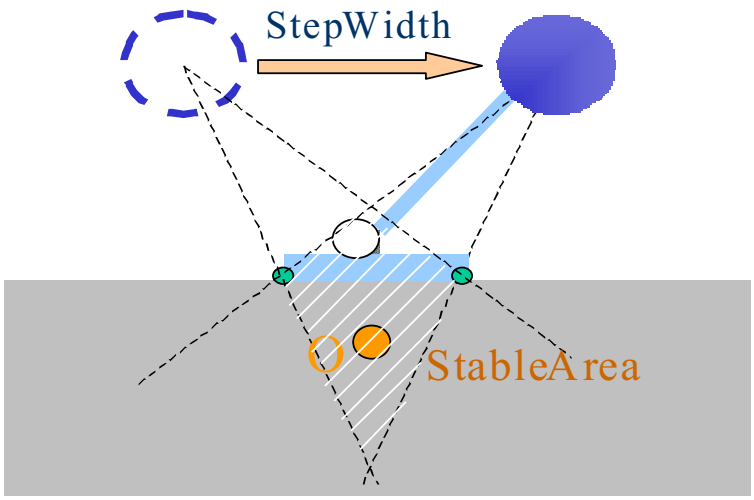


Fig.2-10: Stable area of VSP

the method of ZMP manipulation. There is an advantage, however, that the future trajectory can be figured out and the future stability can be determined easily compared to ZMP manipulation. Therefore VSP is useful for planning a trajectory beforehand.

2.4 Modification of stepping point for stabilization

2.4.1 Replanning of trajectory

This section describes a method to stabilize walking locomotion under large disturbance. Temporary but large disturbance due to collision may occur in human environment. Although a common way to recover from disturbance is to counteract and reject the disturbance, a biped robot cannot just reject large disturbance since large torque input to the ankle joint may cause tipping. Hence the robot must act passively against large disturbance.

In the method planned in this section, a new trajectory is replanned by (2.3) and (2.4) when large disturbance occurs. Tipping does not arise since the new trajectory is passive against the disturbance. ZMP is utilized for detection of disturbance in this method. ZMP is observed all the time and the trajectory is replanned when ZMP leaves its stable area.

The robot misses a desired trajectory after the trajectory replanning while the new trajectory

is compliant to the disturbance. $\boldsymbol{l}_{ch}$, the next stepping point is also replanned by (2.15) to return to the desired trajectory after the next step. Although it is possible to figure out an ideal stepping point in any situation, the point may be out of a motion range of the robot if the disturbance is too large. Therefore, the limit of $\boldsymbol{l}_{ch}$ is set depending on the motion range and $\dot{\boldsymbol{x}}_n(i)$, desired COG velocity at the moment of the ith supporting-point changeover, is modified so as to converge gradually to the desired trajectory.

$$\dot{\boldsymbol{x}}_n = k_{tr}(\boldsymbol{x}_{ncmd} - \boldsymbol{x}_n) + \dot{\boldsymbol{x}}_{ncmd} \tag{2.30}$$

where k_{tr} is a gain for trajectory convergence, $\boldsymbol{x}_{ncmd}$ is a desired position

2.4.2 Trajectory planning of swing leg

A trajectory of the swing leg is planned based on the next stepping point $\boldsymbol{l}_{ch}$ and estimated COG position $\boldsymbol{x}_{ch}$. The trajectory is figured out so as to step and lift off the swing leg with zero velocity.

$$\boldsymbol{l}(t) = \frac{\boldsymbol{\omega}}{2}(1 - \cos(\frac{\pi}{T}t)) + \boldsymbol{l}_p \tag{2.31}$$

$$l_h(t) = \frac{h_f}{2}(1 - \cos(\frac{2\pi}{T}t)) \tag{2.32}$$

here, $\boldsymbol{\omega} = \boldsymbol{l}_{ch} - \boldsymbol{l}_p$, $\boldsymbol{l}(t)$ denotes a horizontal position of the swing-leg tip, $\boldsymbol{l}_p$ denotes the horizontal position of the former stepping point, T is a walking cycle, $l_h(t)$ is the height of the swing-leg tip, h_f is the maximum height of the swing leg trajectory.

2.4.3 Stability of walking motion

A condition to continue stable walking motion is derived here. Trajectory planning based on inverted pendulum mode should be performed so that ZMP stays in its stability region At the same time, constant contact force in the vertical direction should occur on the supporting leg since the height of the pendulum is constant. Hence, the foot surface should ideally keep its ground contact.

On the other hand, large disturbance may occur due to collision. Instability of walking motion due to the disturbance is detected from the ZMP position. In such a situation, ground contact of the support foot surface is held by the proposed method of trajectory replanning. The robot

may not be able to follow the new trajectory, however, due to mechanical limits or torque limits. The robot reaches to these limitations when the walking speed is too high. Hence, walking stability under the limitations is discussed in view of walking speed.

Manipulability of walking speed

As figured out in (2.12), the proposed method determines the COG velocity $\dot{\boldsymbol{x}}_n$ depending on stepping point $\boldsymbol{l}_{ch}$. The relation between $\boldsymbol{l}_{ch}$ and $\boldsymbol{x}_n$ is shown schematically in Fig. 2-11. A long stride reduces the walking speed while a small stride increases it. When the robot is walking in high speed, it is difficult to manipulate the walking speed since the stride can hardly be stretched further. The limit velocity is derived in terms of manipulability of walking speed.

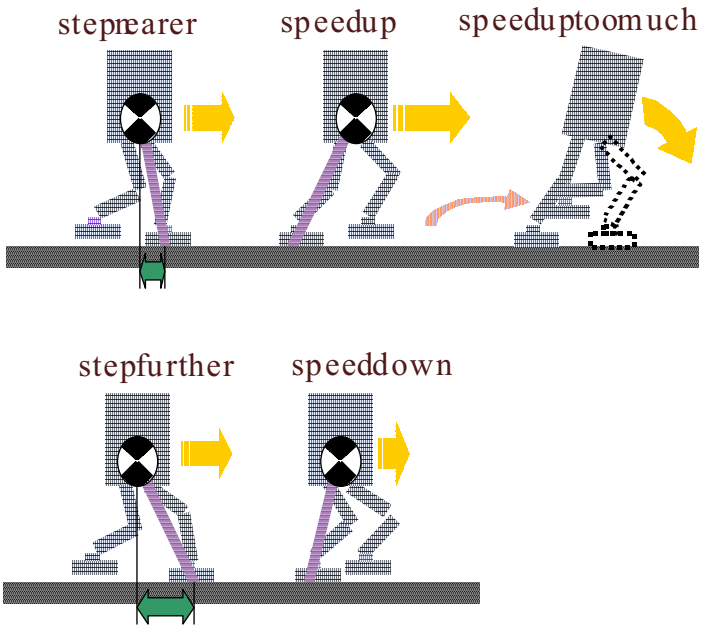


Fig.2-11: Speed control

$\boldsymbol{r}_{max}$ is termed as the maximum value of $\boldsymbol{r}$. In other words, it stands for $\boldsymbol{r}$ at the moment of the supporting-point changeover when the robot is walking with maximum speed. $\boldsymbol{r}_{max}$ and $\dot{\boldsymbol{r}}_{max}$ are figured out from $\boldsymbol{\tau}_{max}$ by dynamics of the inverted pendulum.

$$\ddot{\boldsymbol{r}}_{g_{max}} = \boldsymbol{J}_g \boldsymbol{H}^{-1} \boldsymbol{\tau}_{max} \tag{2.33}$$

$$\ddot{\boldsymbol{r}}_{t_{max}} = \boldsymbol{J}_t \boldsymbol{H}^{-1} \boldsymbol{\tau}_{max} \tag{2.34}$$

$$\dot{\boldsymbol{r}}_{g_{max}} = \frac{\sqrt{\frac{h}{g}}(e^{\sqrt{\frac{g}{h}}T} - e^{-\sqrt{\frac{g}{h}}T})}{2 - (e^{\sqrt{\frac{g}{h}}T} + e^{-\sqrt{\frac{g}{h}}T})} \ddot{\boldsymbol{r}}_{g_{max}} \tag{2.35}$$

$$\dot{\boldsymbol{r}}_{t_{max}} = \frac{T}{\pi} \ddot{\boldsymbol{r}}_{t_{max}} \tag{2.36}$$

$$\boldsymbol{r}_{g_{max}} = \frac{h}{g} \ddot{\boldsymbol{r}}_{g_{max}} \tag{2.37}$$

$$\boldsymbol{r}_{t_{max}} = \frac{T^2}{\pi^2} \ddot{\boldsymbol{r}}_{t_{max}} \tag{2.38}$$

$$\boldsymbol{r}_{max} = min(\boldsymbol{r}_{g_{max}}, \boldsymbol{r}_{t_{max}}) \tag{2.39}$$

$$\dot{\boldsymbol{r}}_{max} = min(\dot{\boldsymbol{r}}_{g_{max}}, \dot{\boldsymbol{r}}_{t_{max}}) \tag{2.40}$$

here, $\boldsymbol{J}_g$ is a Jacobian matrix of the COG, $\boldsymbol{J}_t$ is a Jacobian matrix of the swing-leg tip, $\ddot{\boldsymbol{r}}_{g_{max}}$ is maximum acceleration of the COG, $\ddot{\boldsymbol{r}}_{t_{max}}$ is maximum acceleration of the swing-leg tip.

T_m, a walking cycle during maximum speed locomotion is figured out as follows:

$$T_m = -\sqrt{\frac{h}{g}} l_n \frac{-r_{max} + \sqrt{\frac{h}{g}}\dot{r}_{max}}{r_{max} + \sqrt{\frac{h}{g}}\dot{r}_{max}}. \tag{2.41}$$

Note that T_m depends on r_{max} and $\dot{r}_{max}$ while T only depends on r_{0x} and $\dot{r}_{0x}$, movement in the lateral direction. $\dot{r}_{ch}$, velocity of COG at the moment of the supporting-point changeover, is derived by substituting (2.41) to (2.6).

$$\dot{r}_{ch} = \frac{\sqrt{\frac{g}{h}}r_0 + \dot{r}_0}{2} \cdot \frac{-\sqrt{\frac{g}{h}}r_{max} + \dot{r}_{max}}{\sqrt{\frac{g}{h}}r_{max} + \dot{r}_{max}} + \frac{-\sqrt{\frac{g}{h}}r_0 + \dot{r}_0}{2} \cdot \frac{\sqrt{\frac{g}{h}}r_{max} + \dot{r}_{max}}{-\sqrt{\frac{g}{h}}r_{max} + \dot{r}_{max}} \tag{2.42}$$

Here, $\dot{r}_{ch}$, r_{ch}, $\dot{r}_0$ are all scalar values for deriving limit velocity. r_0 is set in proportion to $\dot{r}_0$.

$$r_0 = k\frac{\dot{r}_0}{\dot{r}_{max}} r_{max} \tag{2.43}$$

where, k is a coefficient to determine average walking speed. Average walking speed does not alter when $k = 1$. On the contrary, average walking speed reduces when $k > 1$ due to a long stride.

$$\begin{aligned} \dot{r}_{ch} &= \frac{\sqrt{\frac{g}{h}}(k\frac{r_{max}}{\dot{r}_{max}}\dot{r}_0 + 1)\dot{r}_0}{2} \cdot \frac{-\sqrt{\frac{g}{h}}r_{max3} + \dot{r}_{max}}{\sqrt{\frac{g}{h}}r_{max} + \dot{r}_{max}} + \frac{-\sqrt{\frac{g}{h}}(k\frac{r_{max}}{\dot{r}_{max}}\dot{r}_0 + 1)\dot{r}_0}{2} \cdot \frac{\sqrt{\frac{g}{h}}r_{max} + \dot{r}_{max}}{-\sqrt{\frac{g}{h}}r_{max} + \dot{r}_{max}} \\ &= \frac{(1-2k)\frac{g}{h}r_{max}^2 + \dot{r}_{max}^2}{-\frac{g}{h}r_{max}^2 + \dot{r}_{max}^2}\dot{r}_0 \end{aligned} \tag{2.44}$$

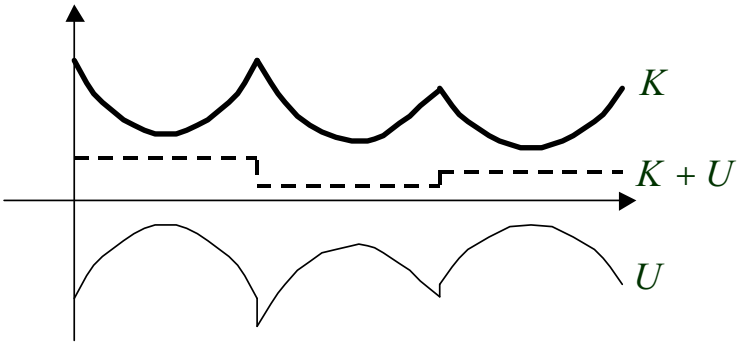


Fig.2-12: Virtual energy function

$\boldsymbol{r}_0$ of the next step equals to current $\boldsymbol{r}_{ch}$, when the supporting-point changeover occurs.

$$\begin{aligned} \dot{r}_{ch}(i+1) &= \frac{(1-2k)\frac{g}{h}r^2_{max} + \dot{r}^2_{max}}{-\frac{g}{h}r^2_{max} + \dot{r}^2_{max}}\dot{r}(i) \\ \dot{r}_{ch}(n) &= (\frac{(1-2k)\frac{g}{h}r^2_{max} + \dot{r}^2_{max}}{-\frac{g}{h}r^2_{max} + \dot{r}^2_{max}})^n \dot{r}_{init} \end{aligned} \tag{2.45}$$

here, $r_{ch}(n)$ denotes r_{ch} of nth step. A condition of velocity converging to zero is derived from (2.45).

$$1 < k < \frac{\dot{r}^2_{max}}{\frac{g}{h}r^2_{max}} \tag{2.46}$$

In case of $\dot{r}_{ch}(n)$ larger than $\dot{r}_{max}$, k must be smaller than 1 because of the torque limit. Then, velocity is out of control and the robot finally falls.

A virtual energy function is introduced as follows to apply the idea to stability assessment.

$$K = \frac{1}{2}M\dot{\boldsymbol{r}}^2 \tag{2.47}$$

$$U = -\frac{Mg}{2h}\boldsymbol{r}^2 \tag{2.48}$$

$$E_{max} = \frac{1}{2}M\dot{\boldsymbol{r}}^2_{max} - \frac{Mg}{2h}\boldsymbol{r}^2_{max} \tag{2.49}$$

Here, K is kinetic energy, U is potential energy, and M is total weight of the robot.

An energy conservation law is applicable to the energy function shown above on the assumption that momentum around VSP is zero. Hence the total energy is constant during an interval of the supporting-point changeover.

Kinetic energy K is continuous when a supporting-point changeover occurs while potential energy U alters discontinuously depending on the stepping point. Velocity is manipulable if

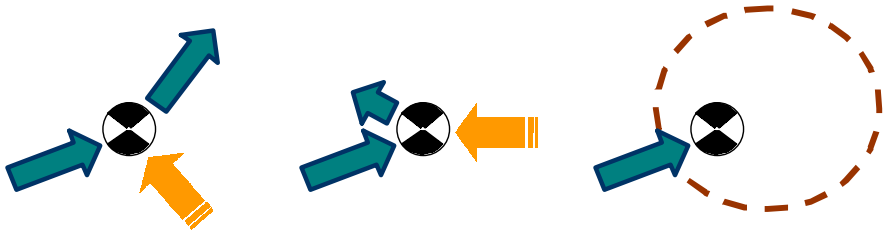


Fig.2-13: Disturbance and kinetic energy

total energy is smaller than E_{max}.

$$K + U < E_{max} \tag{2.50}$$

In sum, the biped locomotion sustains when r is always smaller than r_{max} in the trajectory planning under the condition shown in (2.50).

Possibility of falling can be predicted by applying the property of energy conservation while ZMP shows only the temporal walking stability. Therefore, walking motion could be stabilized beforehand by controlling the total energy when it comes close to E_{max}. The total energy is controlled by planning a proper stepping point.

2.4.4 Limit of collision impulse

This section describes how far the robot can bear with disturbance due to collision.

A biped robot has a mechanism longer in the vertical direction. Because of the mechanism, impulse on a horizontal direction affects the pendulum more than that on the vertical direction. At the same time, most of collision in practice may occur in a horizontal direction. This study therefore limits the collision in a horizontal direction.

The disturbance due to collision is very large but temporal. The robot replans its trajectory based on the dynamics of the pendulum when it detects collision. As shown in Fig. 2-13, kinetic energy K alters when collision occurs. Even if the collision occurs, walking motion carries on by the trajectory replanning as long as the total energy satisfies (2.50).

By substituting (2.47) and (2.48) to (2.50),

$$\frac{1}{2}M\dot{\boldsymbol{r}}^2 - \frac{Mg}{2h}\boldsymbol{r}^2 < E_{max}$$

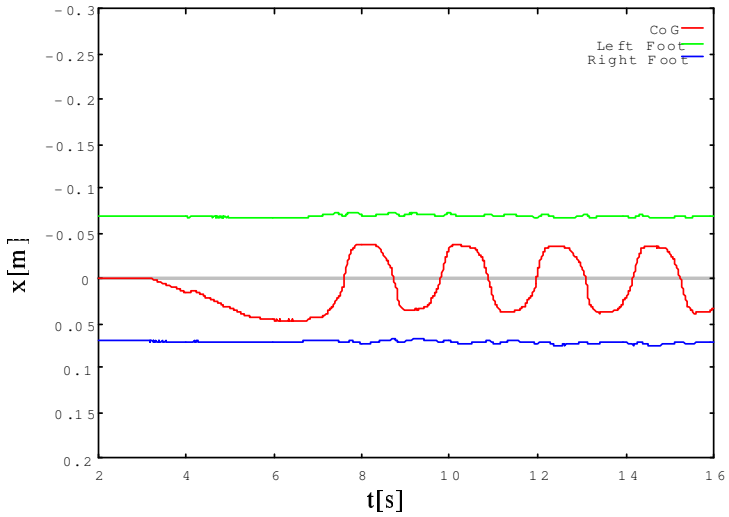


Fig.2-14: Simulation result without impulse

$$|\dot{\boldsymbol{r}}| < \sqrt{\frac{2}{M}E_{max} + \frac{g}{h}|r|^2}. \tag{2.51}$$

(2.52) shows the limit of collision impulse.

$$\left|\frac{\boldsymbol{F}_{dis}}{M} + \dot{\boldsymbol{r}}\right| < \sqrt{\frac{2}{M}E_{max} + \frac{g}{h}|r|^2} \tag{2.52}$$

here, $\boldsymbol{F}_{dis}$ is the collision impulse.

The method deals with any kinds of temporal disturbance besides collision. The walking locomotion is stabilized with this method as long as the disturbance is smaller than the value equivalent to the limit of collision impulse shown in (2.52).

2.4.5 Experiment and simulation results

Fig. 2-14 shows a simulation result of steady walking starting from its initial position.

Fig. 2-15 shows a simulation result with 2 Ns impulse applied on lateral direction. The parameters in the simulation were all the same to the result in Fig. 2-14. Even after the collision, walking motion carried on by replanning the trajectory. The new trajectory after the impulse was compliant to it. The result shows that the robot modified its stepping point against the collision, and momentum on the inverted pendulum was canceled after the swing leg stepped. Finally, the robot converged to the desired trajectory.

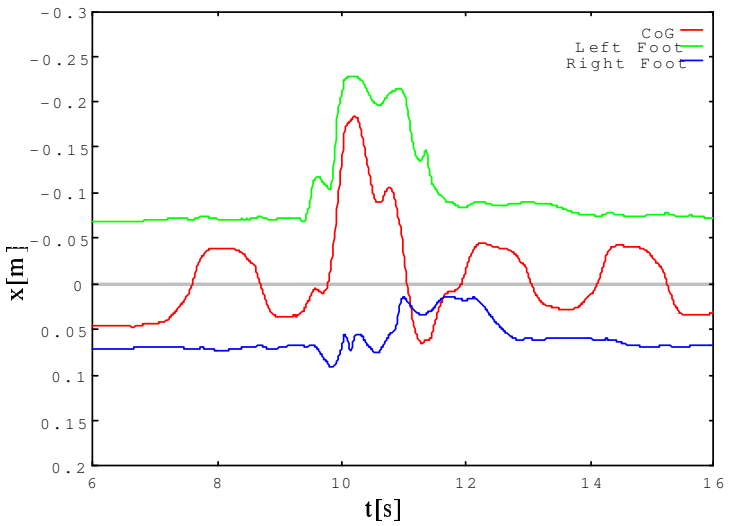


Fig.2-15: Simulation result with 2 Ns impulse given at $t = 9.5s$

Fig. 2-16 shows an experimental result with the same parameters to the simulation. A transient error occurred at first while the robot converged to the steady walking by planning proper stepping points.

2.5 Modification of stepping point for obstacle avoidance

This section discusses how to select a series of footholds for obstacle avoidance. At first, trajectory planning is modified to have higher walking stability since a long stride for obstacle avoidance may destabilize the walk. In the process of modification, VSP is applied to plan a trajectory with continuous acceleration on the inverted pendulum. Secondly, models of an obstacle and a stepping point are introduced to simplify collision detection. Finally, a method to plan a series of stepping points for obstacle avoidance is proposed.

2.5.1 Trajectory planning based on VSP

Trajectory planning of COG

The trajectory planning method applied in the previous sections has a supporting-point changeover. Acceleration on the inverted pendulum alters discontinuously at the moment of

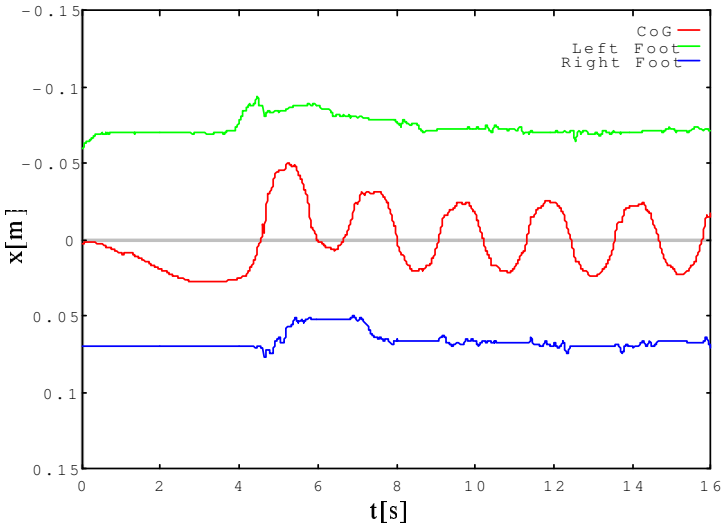


Fig.2-16: Experimental result

the supporting-point changeover while a discontinuous acceleration command destabilizes walking motion. In this section, the trajectory planning method is modified based on VSP to provide a continuous acceleration command. In the modified method, VSP moves continuously from the previous supporting point to the new one instead of having a discrete changeover of the supporting point. VSP moves with constant velocity $\dot{\boldsymbol{u}}$ during the double support phase. Examples of VSP and COG trajectories are shown in Figs. 2-17 and 2-18. One walking cycle is composed of three parts: the latter half of a single support phase, a double support phase, and the former half of the next single support phase.

A trajectory of COG based on dynamics of the pendulum is shown in (2.53), (2.54) and (2.55).

$$\boldsymbol{r}_{s1}(t) = \frac{\boldsymbol{r}_0}{2}(e^{\omega t}+e^{-\omega t}) + \frac{\dot{\boldsymbol{r}}_0}{2\omega}(e^{\omega t}-e^{-\omega t}) \tag{2.53}$$

$$\boldsymbol{r}_{d}(t) = \frac{\boldsymbol{r}_0}{2}(e^{\omega t}+e^{-\omega t}) + \frac{\dot{\boldsymbol{r}}_0}{2\omega}(e^{\omega t}-e^{-\omega t}) + \frac{\dot{\boldsymbol{u}}}{2\omega}(-e^{\omega(t-\frac{Ts}{2})}+e^{-\omega(t-\frac{Ts}{2})}) + \dot{\boldsymbol{u}}t \tag{2.54}$$

$$\begin{aligned}\boldsymbol{r}_{s2}(t) = {} & \frac{\boldsymbol{r}_0}{2}(e^{\omega t}+e^{-\omega t}) + \frac{\dot{\boldsymbol{r}}_0}{2\omega}(e^{\omega t}-e^{-\omega t}) + \frac{\dot{\boldsymbol{u}}}{2\omega}(-e^{\omega(t-\frac{Ts}{2})}+e^{-\omega(t-\frac{Ts}{2})}) \\ & + \frac{\dot{\boldsymbol{u}}}{2\omega}(e^{\omega(t-T_d-\frac{Ts}{2})}-e^{-\omega(t-T_d-\frac{Ts}{2})}) + \dot{\boldsymbol{u}}T_d\end{aligned} \tag{2.55}$$

here, $\boldsymbol{r}_{s1}(t)$, $\boldsymbol{r}_d(t)$, and $\boldsymbol{r}_{s2}(t)$ denote COG position in the current single support phase, the next double support phase, and the next single support phase, respectively. Time t is based on the

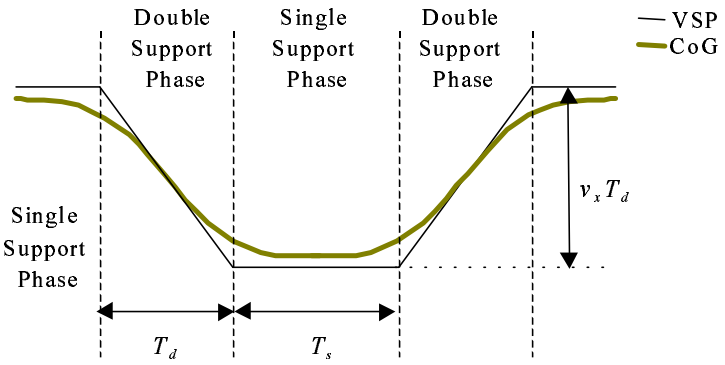


Fig.2-17: VSP and COG trajectories on lateral direction

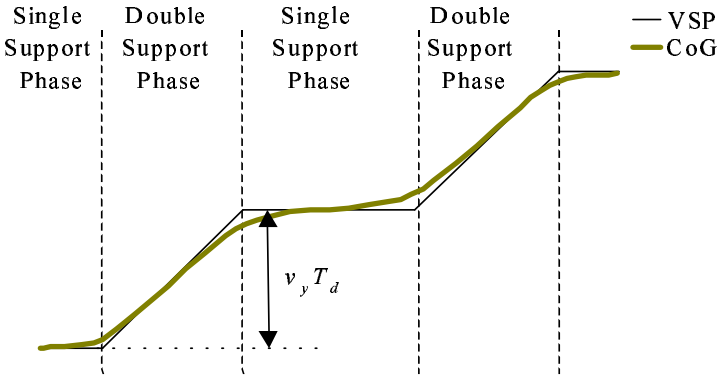


Fig.2-18: VSP and COG trajectories on traveling direction

middle of the current single support phase. T_s and T_d are periods of a single support phase and a double support phase, respectively. $\boldsymbol{r}_0$, which is $\boldsymbol{r}$ at $t = 0$, is measured at the beginning of every walking cycle while its value can be arbitrarily manipulated by VSP.

Trajectory planning of swing leg

Equations (2.31) and (2.32) are modified for trajectory planning of multiple steps ahead as follows:

$$\boldsymbol{l}(t) = \begin{cases} \frac{\boldsymbol{l}_{ch}(i)-\boldsymbol{l}_{ch}(i-2)}{2}\left(1-\cos\left(\frac{\pi}{T_s}(t+\frac{T_s}{2})\right)\right)+\boldsymbol{l}_{ch}(i-2) & (t<\frac{T_s}{2}) \\ \boldsymbol{l}_{ch}(i) & (\frac{T_s}{2}\le t\le\frac{T_s}{2}+T_d) \\ \frac{\boldsymbol{l}_{ch}(i+1)-\boldsymbol{l}_{ch}(i-1)}{2}\left(1-\cos\left(\frac{\pi}{T_s}(t-T_d-\frac{T_s}{2})\right)\right)+\boldsymbol{l}_{ch}(i-1) & (t>\frac{T_s}{2}+T_d) \end{cases} \quad (2.56)$$

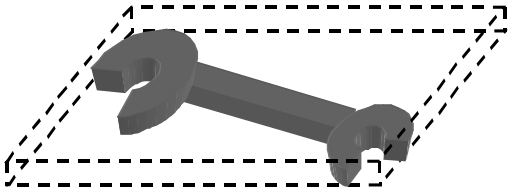


Fig.2-19: Obstacle modeled as cuboid

$$l_h(t) = \begin{cases} \frac{h_f}{2}\left(1-\cos\left(\frac{2\pi}{T_s}(t+\frac{T_s}{2})\right)\right) & (t < \frac{T_s}{2}) \\ 0 & (\frac{T_s}{2} \leq t \leq \frac{T_s}{2}+T_d) \\ \frac{h_f}{2}\left(1-\cos\left(\frac{2\pi}{T_s}(t-T_d-\frac{T_s}{2})\right)\right) & (t > \frac{T_s}{2}+T_d). \end{cases} \quad (2.57)$$

Here, $\boldsymbol{l}_{ch}(i)$ denotes the ith stepping point.

2.5.2 Modeling of obstacle and stepping point

In order to plan a series of stepping points, the biped robot needs to estimate whether the robot will collide with an obstacle. The estimation must be in real time, whereas calculation amount of control is large due to multiplicity of the control system. Therefore an obstacle is modeled as an off-limits on the ground so as to simplify the estimation. Possibility of collision is estimated from the relationship between the off-limits and the stepping point.

At first, the obstacle is modeled as a smallest cuboid that includes the obstacle as shown in Fig. 2-19. The cuboid has edges parallel to the lateral direction. Furthermore, the cuboid is modeled as an off-limits on the ground.

The estimation of collision is a bit complicated since both the foot surface and the cuboid have own width. Therefore the off-limits is expanded over the width of foot surface as shown in Fig. 2-20. The off-limits is also expanded over a margin depending on the orbit of the swing leg. It is called an orbital margin. A safety margin is also included in the off-limits. An example of an orbital margin and a safety margin is shown in Fig. 2-21. Possibility of collision is raised when a stepping point is in the off-limits. In sum, the issue of collision estimation is simplified down to observation of the relationship between the off-limits and the stepping point.

$\boldsymbol{B}$, a vector to describe the off-limits, is figured out from (2.58) to (2.61).

$$B_x^{max} = A_x^{max} + M_x^o + M_x^s + \quad F_l \tag{2.58}$$

$$B_x^{min} = A_x^{min} - M_x^o - M_x^s - \quad F_r \tag{2.59}$$

$$B_y^{max} = A_y^{max} + M_y^o + M_y^s + \quad F_h \tag{2.60}$$

$$B_y^{min} = A_y^{min} - M_y^o - M_y^s - \quad F_t \tag{2.61}$$

where, $\boldsymbol{A}$ denotes limit positions of the cuboid, $\boldsymbol{M}^o$ and $\boldsymbol{M}^s$ are vectors that denote the orbital margin and the safety margin, respectively. Subscripts x and y denote the x-coordinate and the y-coordinate, respectively, and superscripts max and min denote a maximum value and a minimum value, respectively. F denotes distance from the edge to the representative point of the foot. Subscripts l, r, h, and t denote the left edge, the right edge, the heel, and the toe, respectively. As described in Section 2.2, the representative point is the center of the foot surface in this study.

The orbital margin $\boldsymbol{M}^o$ is figured out from (2.62), which is an equation developed from (2.56) and (2.57).

$$\boldsymbol{M}^o = \frac{\boldsymbol{l}_{n+2} - \boldsymbol{l}_n}{2} \left(1 - cos\left(\frac{1}{2} cos^{-1}\left(-\frac{2A_z^{max}}{h_f} + 1\right)\right)\right) \tag{2.62}$$

In case the obstacle is too tall to step over, a trajectory to step aside from the obstacle is planned. If the obstacle is short enough, a trajectory to step it over is planned.

2.5.3 Planning of stepping point

The next stepping point is planned based on the method introduced in Section 2.4. Then the stepping point may be in an off-limits when an obstacle is on the way of the robot. The stepping should be replanned in such a case so that the next stepping point is out of the off-limits. Sudden stride alteration may occur, however, as shown in Fig. 2-22(a). A series of stepping points should be planned with a long-term vision to reduce the alteration since large stride alteration may destabilize walking motion. In order to reduce the alteration of the stride, it is desirable to step just in front of the obstacle and step it over with the least stride. Hence a stamping point, a sub-goal for a future stepping point located in front of the obstacle, is introduced as shown in Fig. 2-22(b). An area for a stamping point is figured out by (2.63).

$$B_y^{max} - \omega_{max} \leq l_{stamp,y} \leq B_y^{min} \tag{2.63}$$

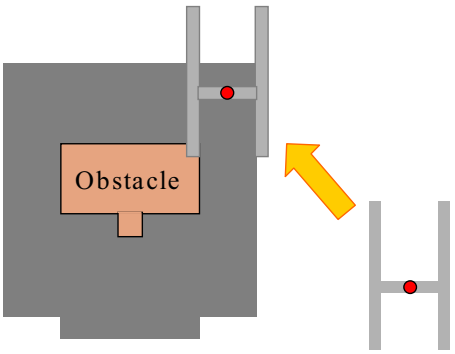


Fig.2-20: Margin for foot width

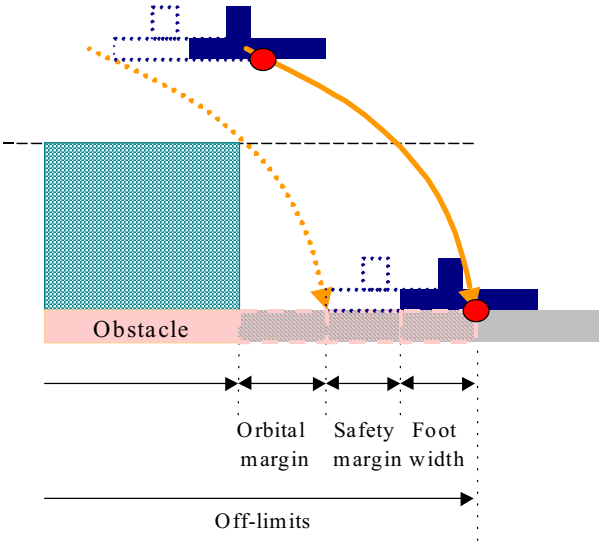


Fig.2-21: Orbital margin and safety margin

where, $l_{stamp,y}$ denotes the y-coordinate of the stamping point.

The area shown in (2.63) is termed "stamp area". As described above, the stamping point in the stamp area is a sub-goal of stepping points and a stride to step on the stamping point with the least steps is derived as follows:

$$\omega_{stamp} = \frac{l_{stamp,y} - l_{n,y}}{2\left(\left[\frac{l_{stamp,y}-l_{n,y}}{2\omega_{max}}\right]+1\right)}. \tag{2.64}$$

Here, $[\cdot]$ derives a maximum integer value smaller than the argument inside. When the robot steps on the stamping point, the robot next has to step over the obstacle. The shortest stride to step over the obstacle is derived as follows:

$$\omega_{stepover} = \frac{B_y^{max} - l_{stamp,y}}{2}. \tag{2.65}$$

2.5.4 Experimental results

We performed experiments in an environment with an obstacle. The robot shown in Fig. 2-1 was at first constrained on the sagittal plane. A result on 2-dimensional walking is shown in Fig. 2-23. Periods of a double support phase and a single support phase were both 1.0 s. The maximum height of the swing leg trajectory was 0.08 m. In the experiment, a spanner wrench was put 0.85 m ahead of the robot as an obstacle. The parameters of the obstacle and a modeled

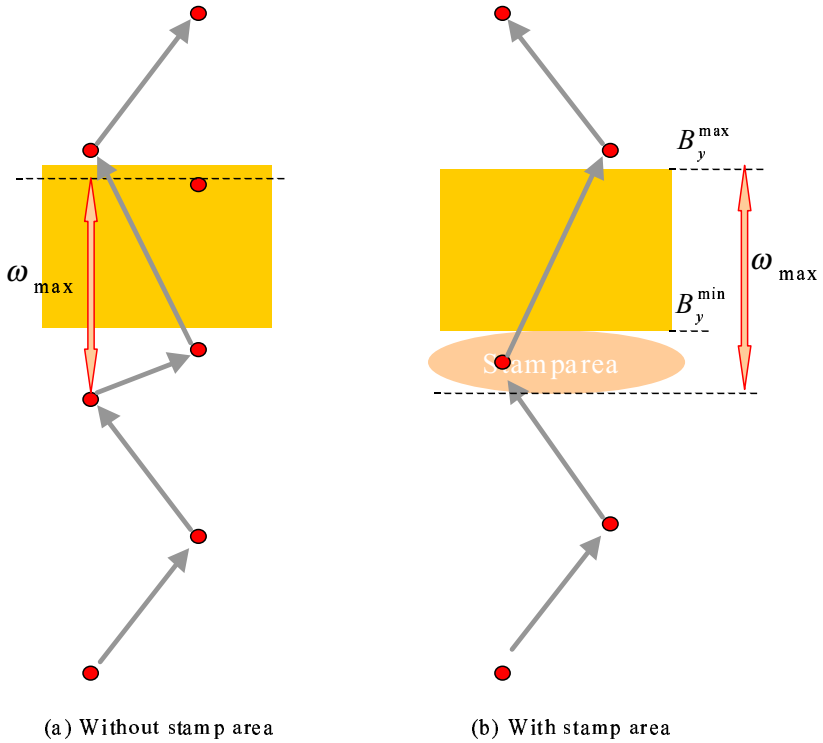


Fig.2-22: Sudden stride alteration avoided with stamp area

cuboid are shown in Table 2.3. The information of the obstacle was given at $t = 8.0$ s while this robot does not have any sensor to detect an object. The result shows that strides were modified in real-time by manipulating VSP. Walking motion was stabilized by providing a continuous acceleration command based on VSP manipulation.

Table 2.3: Parameters of obstacle and modeled cuboid

Parameters of obstacle		Parameters of cuboid	
max. width [m]	0.04	depth [m]	0.07
length [m]	0.18	width [m]	0.17
height [m]	0.006	height [m]	0.006

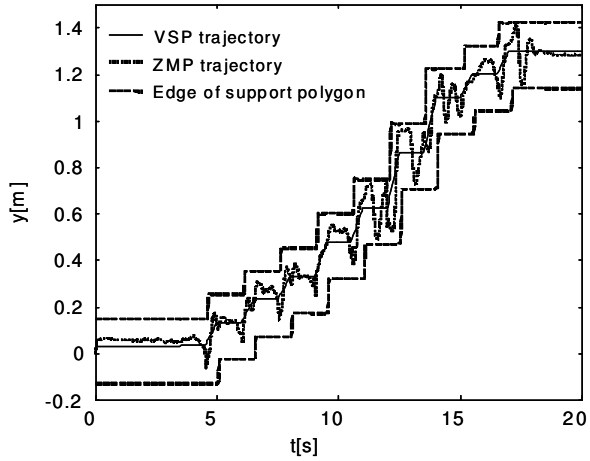


Fig.2-23: ZMP response value on obstacle avoidance locomotion

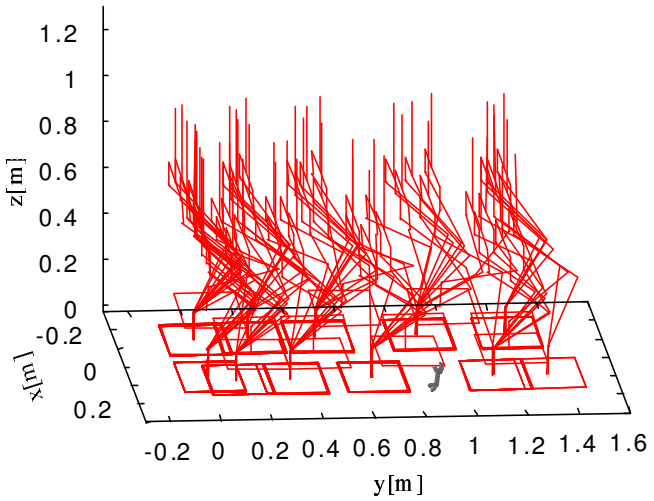
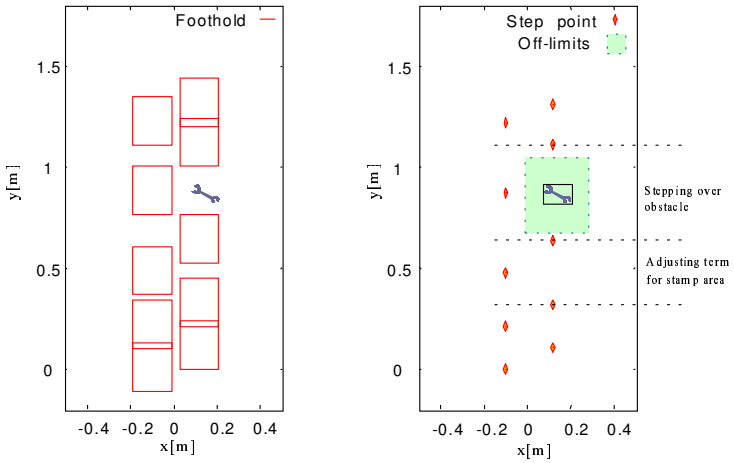


Fig.2-24: 3D stick diagram of locomotion with obstacle

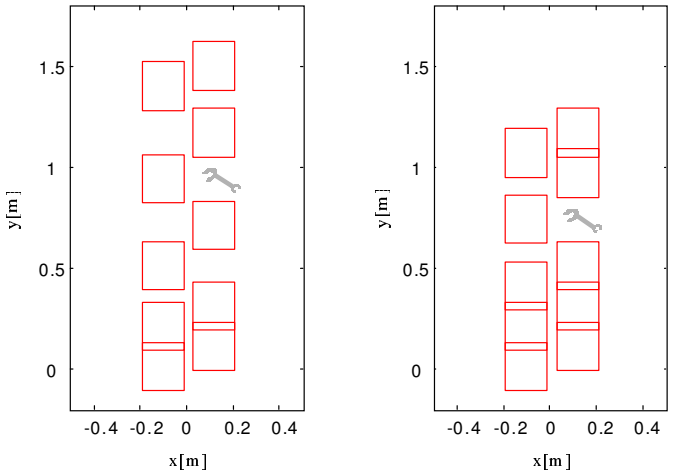
The method is expanded to obstacle avoidance in 3-dimensional walking. Here, trajectory planning on the lateral direction and the traveling direction were performed individually. The stick diagram is shown in Fig. 2-24. Periods of a double support phase and a single support phase were both 3.0 s here. Fig. 2-25(a) shows positions of footholds and the obstacle while Fig. 2-25(b) shows the modeled off-limits and stepping points. The results show that the robot successfully stepped over the obstacle by the stepping point planning based on the model of an

off-limits and stepping points. Fig. 2-26 shows results with an obstacle at a different position. It was confirmed that stride modification dealing with various environments was realized in real-time by setting the stamping point as a sub-goal.



(a)Footholdsandobstacle (b)Steppointsandoff-limits

Fig.2-25: Stepping over an obstacle



(a)Theobstaclewasset 0.95mahead.

(b)Theobstaclewasset 0.75mahead.

Fig.2-26: Obstacle arrangement

2.6 Walk based on environmental mode

This section describes a method to walk in unknown environment. Walking on the unleveled ground is a good example to verify adaptability to unknown environment. In such a case, the foot surface of the robot has to contact with the ground to acquire environmental information. Compliance control is indispensable for contact to unknown environment while it may degrade tracking performance to the desired trajectory. Hybrid control based on environmental information realizes a compliant motion without degrading the tracking performance. The hybrid control system is developed based on an idea of an environmental mode, that is, modal information of environment decomposed by Hadamard matrix.

2.6.1 Extraction of environmental mode

Fig. 2-27 shows an example of four-point contact between a plane and environment. Environmental information is sensed through the contact points in parallel with the contact motion. In this research, such a process is executed by using feet of the biped robot. State of contact is detected by force sensors implemented at the contact points. The environmental modes extracted in the process are heaving, rolling, pitching and twisting as shown in Fig. 2-27.

Heaving mode corresponds to up-down motion while rolling mode and pitching mode correspond to rotation in the roll-axis and the pitch-axis respectively. Twisting mode denotes unevenness of environment. The following description is to show how to extract the environmental modes from the sensors at the contact points. Sensor information at the contact points are defined as e_1, e_2, e_3, e_4 while subscripts denotes the contact points here. 4th order environmental modes are derived from them by (2.66).

$$\begin{bmatrix} e_1 \\ e_2 \\ e_3 \\ e_4 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} \begin{bmatrix} e_h \\ e_r \\ e_p \\ e_t \end{bmatrix} = \boldsymbol{H}_a \begin{bmatrix} e_h \\ e_r \\ e_p \\ e_t \end{bmatrix} \tag{2.66}$$

where h, r, p, t in subscripts denote heaving, rolling, pitching and twisting respectively. $\boldsymbol{H}_a$ in

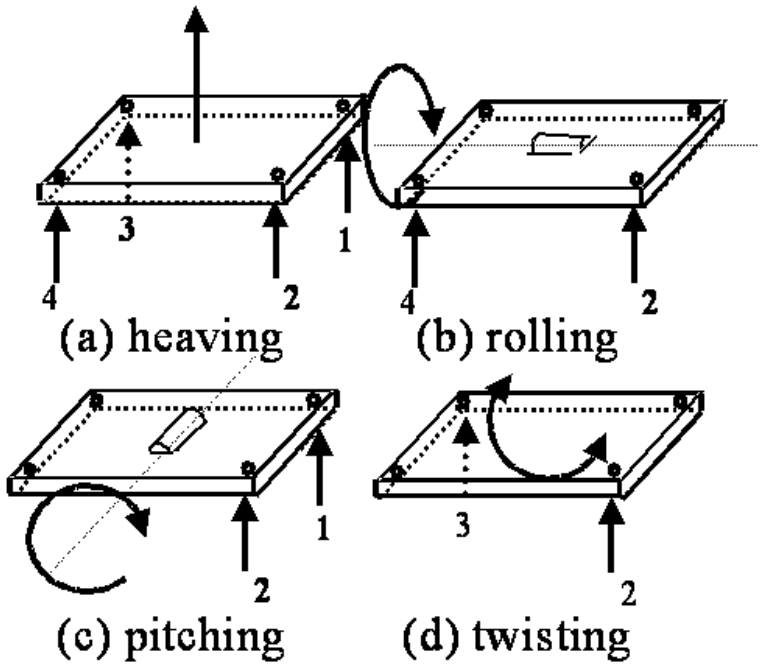


Fig.2-27: Environmental mode

(2.66) is called an Hadamard matrix [51]. The 4th order Hadamard matrix is defined as follows:

$$\boldsymbol{H}_a = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}. \tag{2.67}$$

Hadamard matrix is known to have a following property.

$$\boldsymbol{H}_a \boldsymbol{H}_a^T = n \times \boldsymbol{I}_n \tag{2.68}$$

where $\boldsymbol{I}_n$ is an nth order unit matrix and $\boldsymbol{H}_a$ is an Hadamard matrix.

An inverse matrix of the Hadamard matrix is derived easily from the property. Equation

(2.69) shows the process to extract environmental modes from $e_1 \sim e_4$.

$$\begin{bmatrix} e_h \\ e_r \\ e_p \\ e_t \end{bmatrix} = \boldsymbol{H}_a^{-1} \begin{bmatrix} e_1 \\ e_2 \\ e_3 \\ e_4 \end{bmatrix} = \frac{1}{4}\boldsymbol{H}_a \begin{bmatrix} e_1 \\ e_2 \\ e_3 \\ e_4 \end{bmatrix} \tag{2.69}$$

2.6.2 Classification of environmental mode

Two types of environmental modes, position modes and force modes, are extracted during the contact. A position mode represents irregularity of environment. On the other hand, force mode in the vertical direction represents stiffness and that in a horizontal direction represents friction. Position modes are extracted with a noncontact sensor while force modes are only extracted in response to the contact motion.

2.6.3 Walk on rough terrain

Extraction of environmental modes is utilized for biped walk on an unleveled ground. A 2-dimensional 6 DOF biped robot shown in Fig. 2-28 is used. Note that it is different from the robot used in the previous sections. 2-dimensional motion is only considered in the experiment since the robot is restrained in the sagittal plane. A decentralized control system shown in Fig. 2-29 is developed to control such a robot. The control system is divided into a left leg subsystem and a right leg subsystem. Each leg subsystem is composed of three joint subsystems: a thigh subsystem, a knee subsystem, and an ankle subsystem. The heaving mode and the pitching mode of the ground are acquired in such a system. The environmental modes are extracted by (2.70).

$$\begin{bmatrix} e_h \\ e_p \end{bmatrix} = \frac{1}{2}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} e_1 \\ e_2 \end{bmatrix} \tag{2.70}$$

Walking motion consists of 3 phases: a single support phase, a contact phase and a double support phase. These phases are shifted to the next when the task of the term completes or modal information exceeds its threshold. In other words, event-driven motion control is applied.

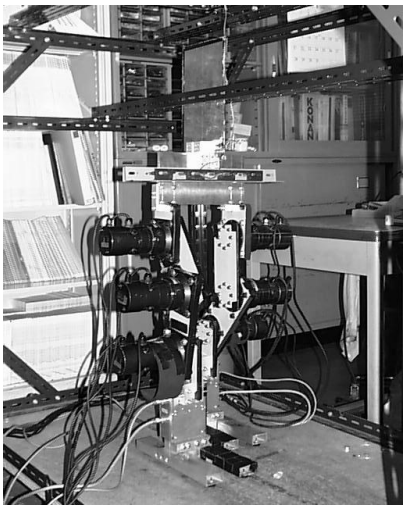


Fig.2-28: Biped robot

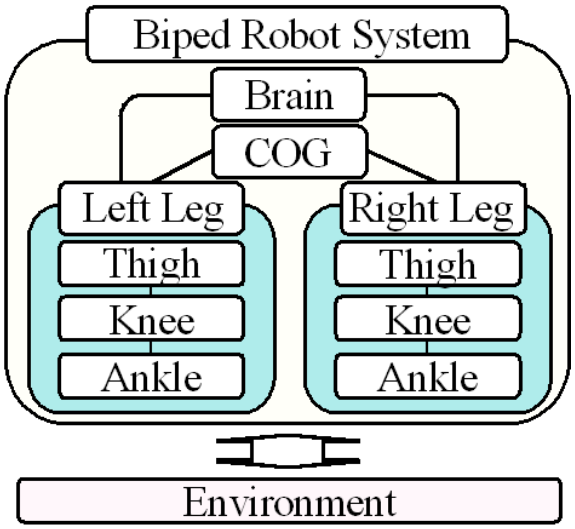


Fig.2-29: Composition of system

The walking cycle is set relatively long in this method in order to avoid large collision in unknown environment.

Single support phase

The single support phase in this method is to search for the ground by the swing leg. A command trajectory of the tip on each leg is given in this term. Here, tip is a point representing the leg which exists at the center of a foot surface. The trajectory is a spline curve connecting a starting point, a middle point and an end point. It satisfies the command values of stride and walking cycle. The foot surface is tilted for about 30 degrees to step on the ground from its heel. At the same time, supporting leg is controlled so as to keep its hip position.

Reference joint angle acceleration $\ddot{\boldsymbol{q}}^{ref}$ is derived from the PD controller shown in (2.71). An example of the odd leg is only shown here while the other leg also has the same processes.

$$\ddot{\boldsymbol{x}}^{ref} = \boldsymbol{K}_p(\boldsymbol{x}^{cmd} - \boldsymbol{x}^{res}) + \boldsymbol{K}_v(\dot{\boldsymbol{x}}^{cmd} - \dot{\boldsymbol{x}}^{res}) \tag{2.71}$$

$$\dot{\boldsymbol{q}}^{ref} = \boldsymbol{J}^{-1}\dot{\boldsymbol{x}}^{ref} \tag{2.72}$$

where, $\boldsymbol{x} = [x, y, \theta]^T$ is a vector that denotes position and attitude of a leg tip. $\boldsymbol{q} = [q_1, q_2, q_3]^T$ is a joint angle vector of a leg. $\boldsymbol{K}_p, \boldsymbol{K}_v$ are position gain and velocity gain respectively.

Current reference I^{ref} on each joint subsystem is derived by (2.73).

$$I^{ref} = \frac{M_n}{K_t}\ddot{q}^{ref} \tag{2.73}$$

here, M_n is a nominal value of joint inertia, K_t is a motor torque constant. Acceleration control is acquired on each joint subsystem with applying DOB. The block diagram of the entire system during the single support phase is shown in Fig. 2-30.

The single support phase shifts to the contact phase when heaving or pitching modes are detected from the ground.

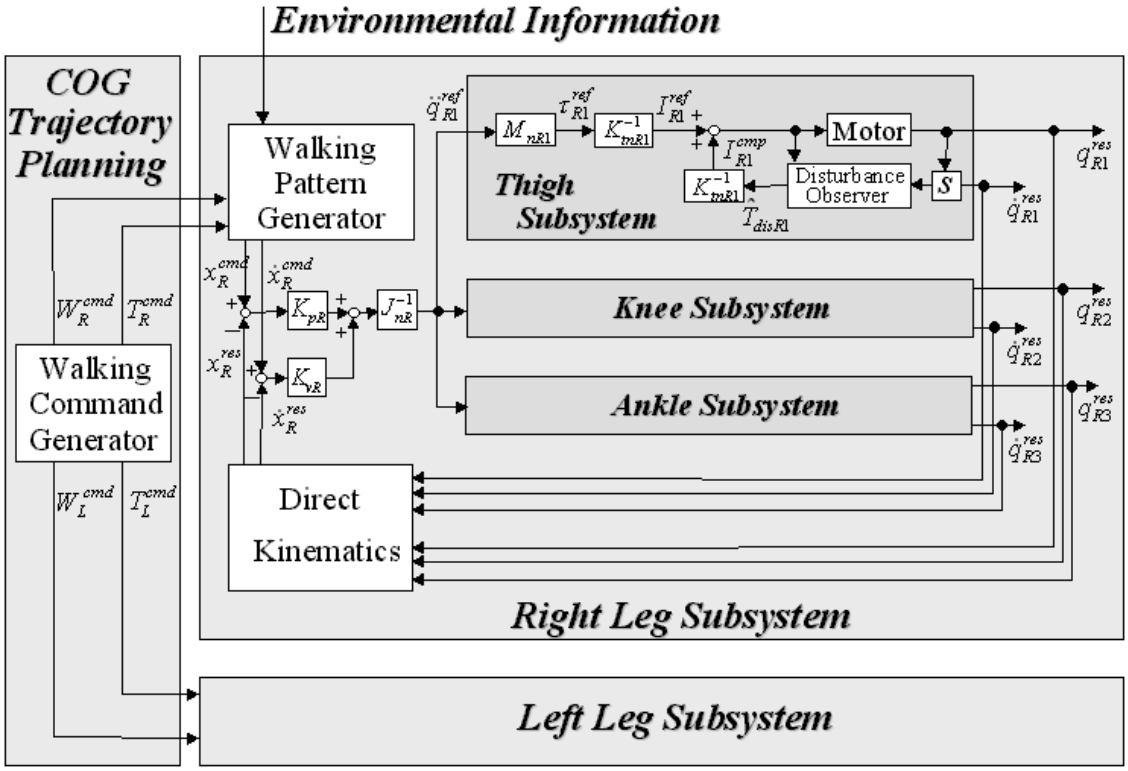


Fig.2-30: Double and single support phase

Contact phase

The contact phase starts when the heel of the swing leg contacts with the ground during the single support phase. Force environmental modes F_h and F_p are acquired then. The ankle

joint subsystem on the swing leg handles F_p, force on pitching mode, by compliance control. Command values of the ankle joint are given by (2.74).

$$M_c\ddot{q}_{cR3}^{cmd} + D_c\dot{q}_{cR3}^{cmd} + K_c q_{cR3}^{cmd} = K_f F_p \tag{2.74}$$

Reference value of ankle joint is given by (2.75).

$$\ddot{q}_3^{ref} = -K_{p3}\dot{q}_c^{cmd} - K_{v3}q_c^{cmd} - \ddot{q}_c^{cmd} \tag{2.75}$$

here, K_f is a feedback gain for external force. Thigh and knee subsystems control the position of the ankle joint so that it tracks an trajectory of arc with the heel at the center. Control of the support leg is the same to the single support phase. The block diagram when the right foot come to contact with ground is shown in Fig. 2-31.

The ankle subsystem deals with pitching mode which should be compliant to the ground. On the other hand, the thigh and the knee subsystems deals with the trajectory of the ankle which needs high tracking performance. The contact phase shifts to the double support phase when force on the pitching mode becomes 0 due to the toe contact. As a result, the swing leg absorb the collision force and acquire line contact with the ground. At the same time, ground condition is extracted as position environmental modes.

2.6.4 Double support phase

Hip link moves to just above the center of the fore foot during the double support phase. The endpoint of the hip trajectory is figured out based on the heaving mode P_h extracted in the contact phase. Position command of tip on each leg is derived from the endpoint trajectory. The block diagram is the same to the single support phase.

The double support phase shifts to the single support phase when the hip reaches to the endpoint.

2.6.5 Experiment

An experiment was performed by the robot shown in Fig. 2-28 to show the validity of the method. Link parameters and control parameters are shown in Table 2.4 and Table 2.5, respectively. Walking parameters for trajectory planning is shown in Table 2.6.

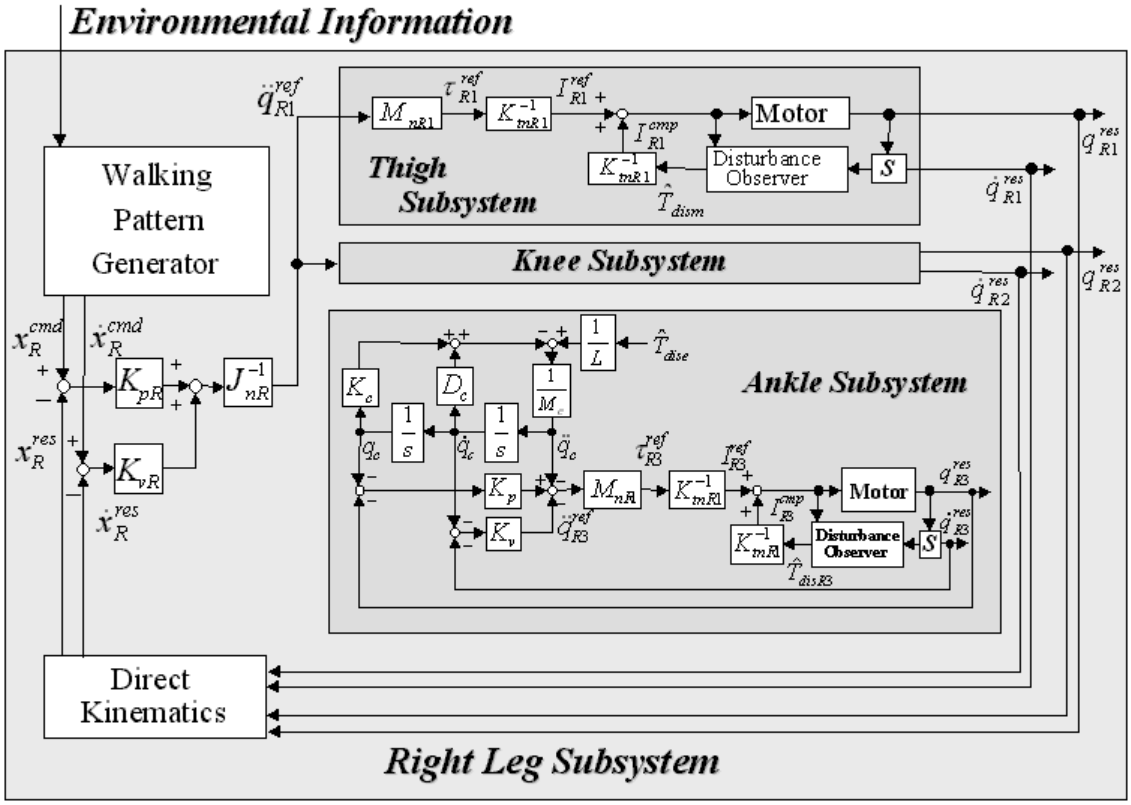


Fig.2-31: Contact phase

Force information is acquired from force sensors attached on the feet. Fig. 2-32 shows a stick diagram of walking motion when the robot walked on unleveled ground. The ground was flat around the start point while it was a slope with 10° from 0.3 m ahead. Environmental modes of heaving and pitching, acquired in the experiment, are shown in Table 2.7 and Fig. 2-33, respectively. The result shows that the robot walked on unleveled ground with quarrying environmental modes. Compliance control on pitching mode reduced collision impulse. From the result that the upper body leaned back gradually, we may infer that errors on environmental modes were accumulated with each step. The problem should be solved with attaching an additional attitude sensor.

Table 2.4: Link parameter of biped robot

	Mass[kg]	Length[m]
Trunk	1.2	0.25
Thigh link	3.6	0.16
Knee link	3.6	0.16
Ankle link	6.8	0.295
Toe		0.185
Heel		0.115

Table 2.5: Control parameters

Control Parameter		Value
Position gain	K_p	900.0
Velocity gain	K_v	60.0
Cutoff frequency of DOB [rad/s]	G_g	100.0
Sampling period [s]	S_t	0.001

2.7 Summary of chapter

This chapter discussed control of a biped robot in terms of motion planning for adaptation to human environment. Section 2.3 introduced a method applying VSP. The robot can plan a

Table 2.6: Walking parameters

Walking Parameter		Value
Single Support Phase[s]	T_{single}	3.6
Double Support Phase[s]	T_{double}	2.4
Walk Length[m]	W	0.20
Waist Height[m]	H_{waist}	0.50
Swing Leg Height(max.)[m]	H_{sw}	0.15
Desired Swing Leg Angle[m]	$H_{swangle}$	0.15

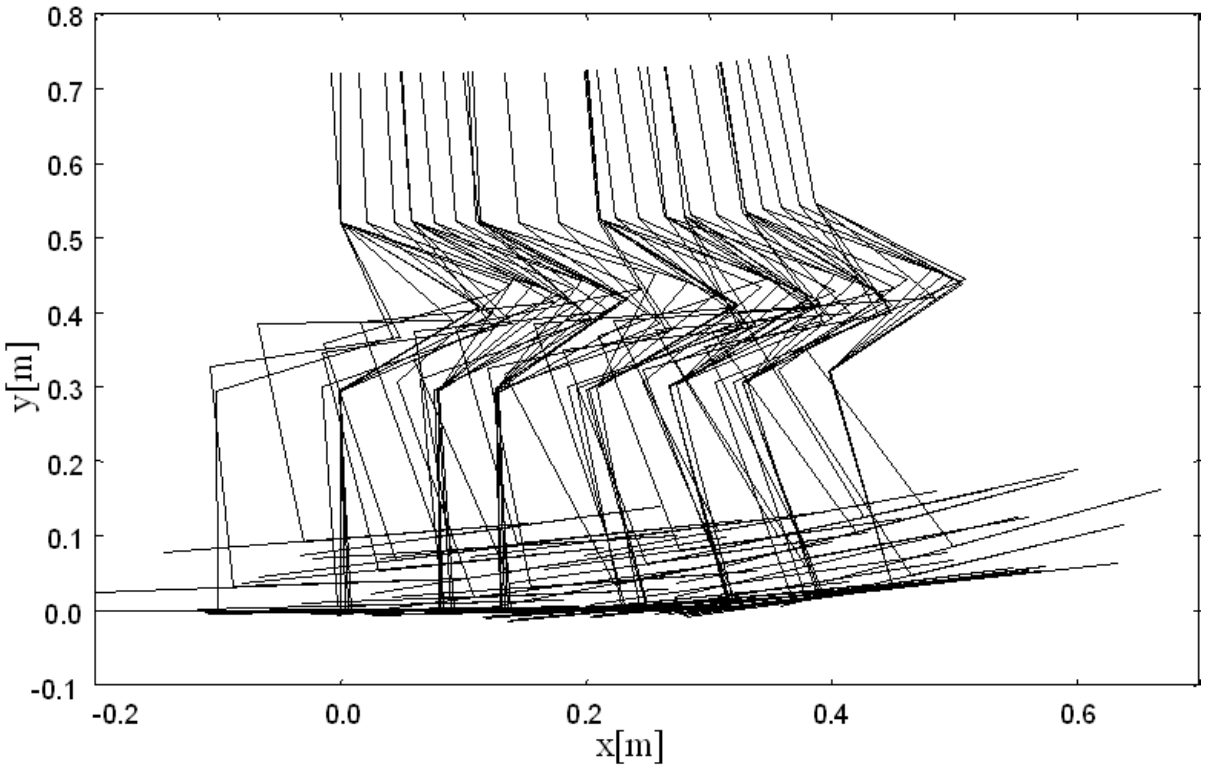


Fig.2-32: Walking experiment on floor with unknown slope

broad range of biped motion while the inverted pendulum mode is strictly constrained without the idea of VSP . It was shown in Section 2.4 that real-time modification of a stepping point leads to stabilization of biped motion. It stabilized even the biped motion with large bumping force on the upper body. Obstacles in structured environment could be avoided by particular trajectory planning shown in Section 2.5.

These results shown in this section indicate that adaptation to environment in most cases is achieved by motion planning if environmental information is known, i.e. if the robot works in structured environment.

On the other hand, Section 2.6 shows that some compliance should be introduced into the control system if unknown environmental information exists. Compliance control may degrade tracking performance of the robot although it is important for adaptation to unknown environment. Hence the modal decomposition method is introduced to separate the controller into a

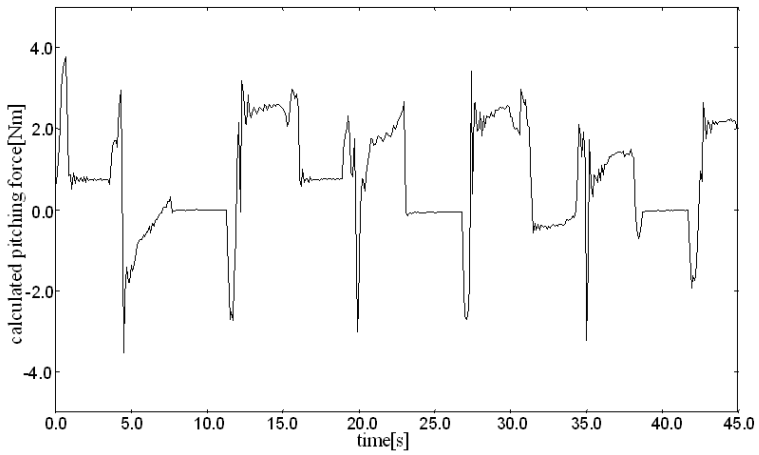


Fig.2-33: Transition of Environmental Mode (Pitching)

Table 2.7: Abstracted Environmental Mode

Step	Heaving [m]	Pitching [rad]	Step	Heaving [m]	Pitching [rad]
1	0.00000	0.00000	5	0.01134	0.15783
2	0.00009	0.00208	6	0.01736	0.19656
3	0.00302	0.01008	7	0.02042	0.17747
4	0.00479	0.11818			

mode to be compliant and a mode to reject disturbance. The modal decomposition is based on environmental modes derived by an Hadamard matrix. This example shows that robots in human environment should have a hybrid control system consists of force and position control in order to acquire both environmental adaptability and tracking performance.

Chapter 3

Flexible design of bilateral control

3.1 Introduction

The previous chapter showed the priority of motion planning for robots in structured environment. At the same time, it also showed the need of compliance control for the robots in unstructured environment. The idea of environmental mode was introduced for constructing a hybrid control system to acquire both environmental adaptability and tracking performance.

Then, another idea inspired by the environmental mode is proposed. The idea is that a sort of a mode may exist in a robot control system as is the case with an environmental mode. The mode will be termed "function mode" later. This chapter describes a design framework based on the idea. A function mode should be able to deal with an environmental mode as a part of the task to the robot. Furthermore, it should also be able to deal with a cooperative task by multiple robots. Such a framework is verified in a bilateral control system, the simplest form of a robot control system for a cooperative task.

Bilateral control is a method for a robotic teleoperation system. It is superior in the respect that it transfers haptic sense to a remote place. Hence it has been studied for a long time to carry out a skilled operation in the remote place. Although it is simply composed of two manipulators, its design is complicated. Many types of controllers such as position-position, position-force, force-position, and force-force architectures were investigated. Lawrence [52] utilized "four-channel" architecture that describes general structure of bilateral control systems. Modal decomposition methods simplify the design of the bilateral control system [15, 53, 54]. Although many types of bilateral control systems exist, many new architectures are still investigated.

At the same time, promising indices have also been proposed. Yokokohji and Yoshikawa defined an ideal response of the bilateral control system in [55]. Hannaford specified this response with the hybrid matrix [56]. The condition of "transparency", another concept of an ideal response, was evaluated using the hybrid matrix [52]. Iida and Ohnishi proposed indices of "reproducibility" and "operationality", which give quantitative evaluation [57]. The ideal response is divided into two independent features evaluated by the indices.

Most of previous studies on bilateral control aimed at acquisition of transparency. Ideally, such a system can be represented by an infinitely stiff and weightless mechanical connection between the end-effectors of a master arm and a slave arm [58]. In order to actualize it by control, high feedback gains in both position and force control are required. However, it is indicated in [58] that the highest level of force feedback is not universally beneficial. Lee and Li presented an interesting design method for a bilateral control system to behave as a common passive rigid mechanical tool [54]. The study infers that a teleoperator should appear to the human as a mechanical extension of his/her body.

In sum, the main goal of the study in this chapter is acquisition of an adjustable bilateral control system instead of the ideal system. Here, the adjustable system means a control system that allows the following arrangement.

- modification of its apparent mechanical parameters
- dynamical task shift

An idea of functionality is proposed as a new design framework for an adjustable bilateral control system. Simplicity and explicitness of the proposed framework provide a way to design the adjustable system.

Contents of this chapter are as follows: The idea of functionality is proposed in Section 3.2. This study introduces coordinate transformation to associate functions with robots. The coordinate transformation method is described in Section 3.3. Section 3.4 shows examples of controller design and indicates some features of the method. Experimental results are shown in Section 3.5. Finally this chapter is concluded in Section 3.6.

3.2 Definition of function

Functionality is an idea for design of a bilateral control system. At first, the system role is defined as follows:

Definition 1 "System role" is a description of the requirement from a user to a robot control system.

The system role represents a momentary feature of the control system and does not include sequential information. The control system should be designed to satisfy this system role. It is difficult, however, to associate a system role directly with a controller since the system role consists of abstract words. The idea of functionality is introduced below to concretize the system role.

Definition 2 "Function" is a minimum component of a system role. Conversely, the system role is described as a combination of functions.

In this study, a bilateral control system is designed based on functions. Fig. 3-1a) shows a design framework of conventional bilateral control systems. In conventional methods, a controller corresponds directly to a manipulator. Two controllers receive their command values from a command generator. The command generator, which determines the type and the feature of the control system, should be designed so as to meet the system role. However, the behavior of the total system is hardly analogized from the structure of the command generator. Its design depends on the empirical knowledge of the designer since there is no explicit procedure to decide the architecture. Unclear correspondence between the controllers and the system role leads to the difficulty of design.

Hence we propose a design framework shown in Fig. 3-1b). Here, the system role is divided into functions. Controllers are designed to satisfy individual functions. There are two categories of functions in the bilateral control system. One is a function of coupling and the other is a function of entire motion.

It is able to control master and slave manipulators as if they are coupled with a spring. It is

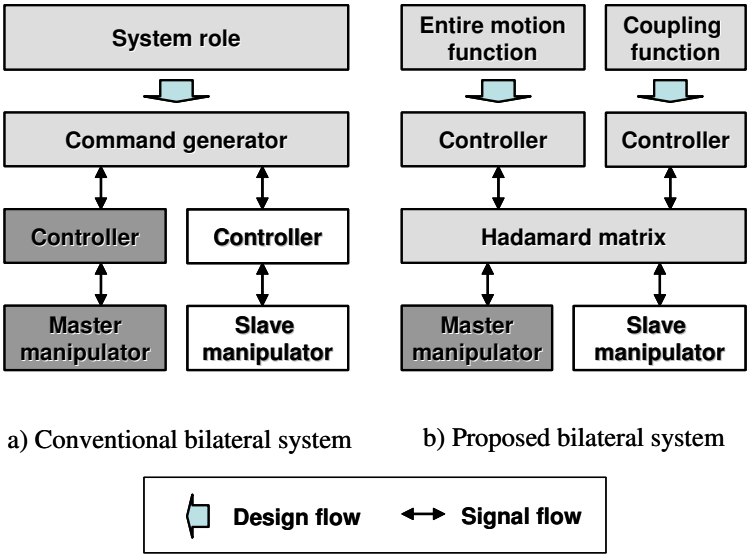


Fig.3-1: Design framework of bilateral system

also able to realize a rigid coupling. This kind of part that a controller plays will be treated as a function. We define these parts as a spring coupling function and a rigid coupling function, respectively. These functions are classified as coupling functions.

Meanwhile, functions to control the entire motion exist when master and slave manipulators are treated as a coupled system. The effect of friction can be compensated if an accurate friction model is derived. This is defined as a friction compensation function. It is also able to manipulate the apparent inertia of the entire system. This is defined as an inertia manipulation function. These functions are classified as entire motion functions.

Functions can be realized with control while some of the functions are also achievable with mechanical tools. In each case, they are treated as the same functions. The examples of functions are shown in Fig. 3-2.

Coupling functions are accomplished by controlling the position gap of the two manipulators, master and slave. At the same time, entire motion functions relate to the sum of the two manipulator positions. Consequently, coordinate transformation should be applied to design the controller based on functions.

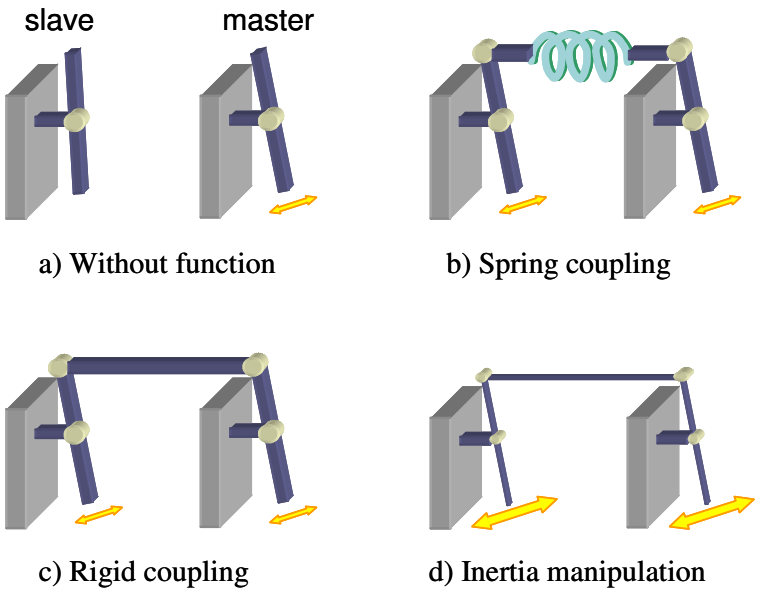


Fig.3-2: Examples of functions

3.3 Coordinate transformation

3.3.1 Coordinate transformation in equal ratio

In order to design the controllers based on functions, the robot coordinate should be transformed into a new coordinate based on functions. Here, the coordinate based on the information of each control object is defined as a robot coordinate. On the other hand, the transformed coordinate based on the information of functions is defined as a function coordinate. An Hadamard matrix is useful for the transformation of a bilateral control system [51]. It transforms each variable into common and differential modes, which relate to the functions. In other words, this coordinate transformation performs modal decomposition. Coordinate transformation is executed by (3.1).

$$\begin{bmatrix} x_+ \\ x_- \end{bmatrix} = \frac{1}{2}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}\begin{bmatrix} x_m \\ x_s \end{bmatrix} = \frac{1}{2}\boldsymbol{H}_2\begin{bmatrix} x_m \\ x_s \end{bmatrix} \tag{3.1}$$

where the subscript m denotes the master manipulator, the subscript s denotes the slave manipulator, the subscript + denotes the common coordinate and the subscript – denotes the

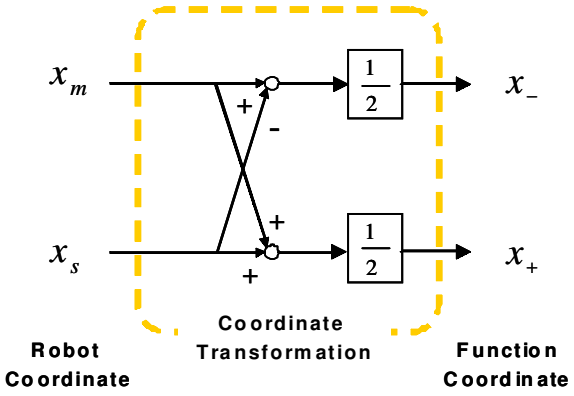


Fig.3-3: Coordinate transformation

differential coordinate. x shows the position of a manipulator. $\boldsymbol{H}_2$ is the quadratic Hadamard matrix.

In this chapter, kinematics and dynamics of master and slave manipulators are considered in 1 DOF for simplicity. x_m and x_s are defined as a robot coordinate system. x_+ and x_- are defined as a function coordinate system.

Velocity and force are also transformed with the Hadamard matrix as follows:

$$\begin{bmatrix} \dot{x}_+ \\ \dot{x}_- \end{bmatrix} = \frac{1}{2}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}\begin{bmatrix} \dot{x}_m \\ \dot{x}_s \end{bmatrix} = \frac{1}{2}\boldsymbol{H}_2\begin{bmatrix} \dot{x}_m \\ \dot{x}_s \end{bmatrix} \tag{3.2}$$

$$\begin{bmatrix} f_+ \\ f_- \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}\begin{bmatrix} f_m \\ f_s \end{bmatrix} = \boldsymbol{H}_2\begin{bmatrix} f_m \\ f_s \end{bmatrix} \tag{3.3}$$

$$\begin{bmatrix} \tau_+ \\ \tau_- \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}\begin{bmatrix} \tau_m \\ \tau_s \end{bmatrix} = \boldsymbol{H}_2\begin{bmatrix} \tau_m \\ \tau_s \end{bmatrix} \tag{3.4}$$

where f denotes external force given to the manipulator and τ denotes input force.

Fig. 3-3 shows a block diagram of the coordinate transformation. With the Hadamard matrix, a position plain on robot coordinates is transformed to a position plain on function coordinates. A function is related to a base on the function coordinate space.

3.3.2 Scaling matrix

Coordinate transformation expanded for scaling control is shown by (3.5).

$$\begin{aligned}\begin{bmatrix} x_+ \\ x_- \end{bmatrix} &= \frac{\beta}{\alpha+\beta}\begin{bmatrix} 1 & \beta \\ \frac{1}{\alpha} & -1 \end{bmatrix}\begin{bmatrix} x_m \\ x_s \end{bmatrix} \\ &= \frac{\beta}{\alpha+\beta}\boldsymbol{H}_s\begin{bmatrix} x_m \\ x_s \end{bmatrix} \end{aligned} \tag{3.5}$$

where α denotes a scaling factor in common coordinate and β denotes a scaling factor in differential coordinate. $\boldsymbol{H}_s$ is a transformation matrix expanded for scaling control. We term this scaling matrix. When $\alpha = 1$ and $\beta = 1$, this bilateral control system becomes in equal ratio and (3.5) becomes equivalent to (3.1).

A scaling matrix has the feature that its inverse is in proportion to itself. Therefore inverse coordinate transformation is easily derived by (3.6).

$$\begin{bmatrix} x_m \\ x_s \end{bmatrix} = \begin{bmatrix} 1 & \beta \\ \frac{1}{\alpha} & -1 \end{bmatrix}\begin{bmatrix} x_+ \\ x_- \end{bmatrix} = \boldsymbol{H}_s\begin{bmatrix} x_+ \\ x_- \end{bmatrix} \tag{3.6}$$

Velocity and force are also transformed with a scaling matrix as follows:

$$\begin{aligned}\begin{bmatrix} \dot{x}_+ \\ \dot{x}_- \end{bmatrix} &= \frac{\alpha}{\alpha+\beta}\begin{bmatrix} 1 & \beta \\ \frac{1}{\alpha} & -1 \end{bmatrix}\begin{bmatrix} \dot{x}_m \\ \dot{x}_s \end{bmatrix} \\ &= \frac{\alpha}{\alpha+\beta}\boldsymbol{H}_s\begin{bmatrix} \dot{x}_m \\ \dot{x}_s \end{bmatrix} \end{aligned} \tag{3.7}$$

$$\begin{bmatrix} f_+ \\ f_- \end{bmatrix} = \begin{bmatrix} 1 & \beta \\ \frac{1}{\alpha} & -1 \end{bmatrix}\begin{bmatrix} f_m \\ f_s \end{bmatrix} = \boldsymbol{H}_s\begin{bmatrix} f_m \\ f_s \end{bmatrix} \tag{3.8}$$

$$\begin{bmatrix} \tau_+ \\ \tau_- \end{bmatrix} = \begin{bmatrix} 1 & \beta \\ \frac{1}{\alpha} & -1 \end{bmatrix}\begin{bmatrix} \tau_m \\ \tau_s \end{bmatrix} = \boldsymbol{H}_s\begin{bmatrix} \tau_m \\ \tau_s \end{bmatrix}. \tag{3.9}$$

3.3.3 Dynamics

Dynamics in robot coordinates are developed to the dynamics in function coordinates so as to show that it is reasonable to design the control system in each scaled function coordinate.

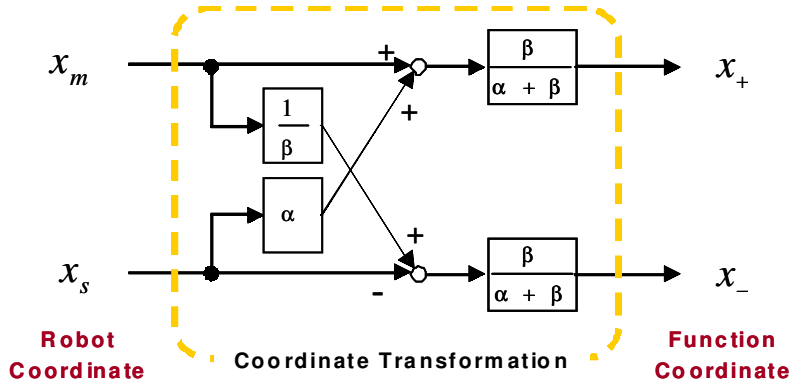


Fig.3-4: Coordinate transformation from robot coordinate to function coordinate

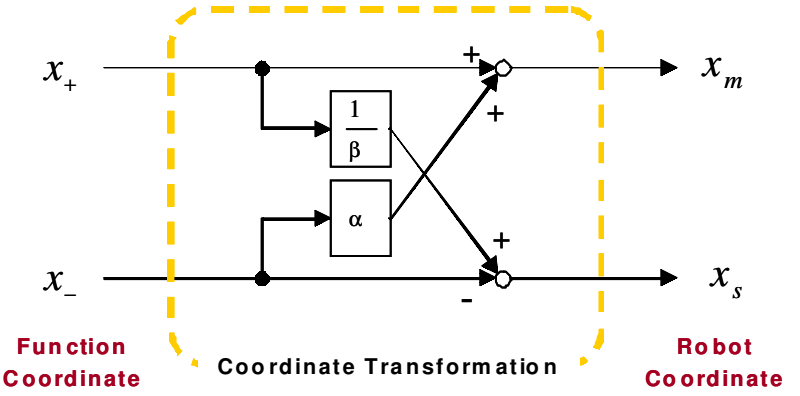


Fig.3-5: Coordinate transformation from function coordinate to robot coordinate

Dynamic equation on master and slave coordinate under scaling control is shown as follows:

$$M_m\ddot{x}_m + \mu_m\dot{x}_m = \tau_m + f_m \tag{3.10}$$

$$M_s\ddot{x}_s + \mu_s\dot{x}_s = \tau_s + f_s. \tag{3.11}$$

Here, M denotes mass of the manipulator and μ denotes friction coefficient.

By assuming that the models of master and slave manipulators are equivalent, dynamic equations in function coordinates are figured out from (3.10) and (3.11).

$$M_t\frac{\alpha}{\alpha+\beta}(\ddot{x}_m + \beta\ddot{x}_s) + \mu_t\frac{\alpha}{\alpha+\beta}(\dot{x}_m + \beta\dot{x}_s)$$

$$= (\tau_m + \beta\tau_s) + (f_m + \beta f_s) \tag{3.12}$$

$$M_t \frac{\alpha}{\alpha+\beta}(\frac{1}{\alpha}\ddot{x}_m - \ddot{x}_s) + \mu_t \frac{\alpha}{\alpha+\beta}(\frac{1}{\alpha}\dot{x}_m - \dot{x}_s)$$

$$= (\frac{1}{\alpha}\tau_m - \tau_s) + (\frac{1}{\alpha}f_m - f_s) \tag{3.13}$$

where, $M_t = \frac{\alpha+\beta}{\alpha}M_m$ and $\mu_t = \frac{\alpha+\beta}{\alpha}\mu_m$.

$$M_t\ddot{x}_+ + \mu_t\dot{x}_+ = \tau_+ + f_+ \tag{3.14}$$

$$M_t\ddot{x}_- + \mu_t\dot{x}_- = \tau_- + f_- \tag{3.15}$$

From (3.14) and (3.15), it is shown that dynamics on common and differential coordinates could be treated same as two independent physical systems. It is able to add both inputs of a common coordinate controller and a differential coordinate controller to master or slave manipulators since the inputs are independent in a transformed position plain. With this fact, we can design a controller on each function coordinate independently. In sum, the controller design becomes simple and explicit.

3.4 Controller design

In this section, some examples of a function-based controller design method are described. Then features of the design method are indicated. The examples are based on coordinate transformation without scaling while the scaling matrix is applicable.

Fig. 3-6 shows the first example. A PD controller is mounted in the differential coordinate. The torque input from the PD controller is figured out as follows:

$$\tau_- = -K_-x_- - D_-\dot{x}_-. \tag{3.16}$$

Substituting (3.16) to (3.15),

$$M_t\ddot{x}_- + \mu_t\dot{x}_- = f_- - K_-x_- - D_-\dot{x}_-$$

$$M_t\ddot{x}_- + (D_- + \mu_t)\dot{x}_- + K_-x_- = f_-. \tag{3.17}$$

As shown in (3.17), PD controller works the same as a mechanical spring with stiffness K_- and viscosity D_-. Hence this controller serves as a spring coupling function.

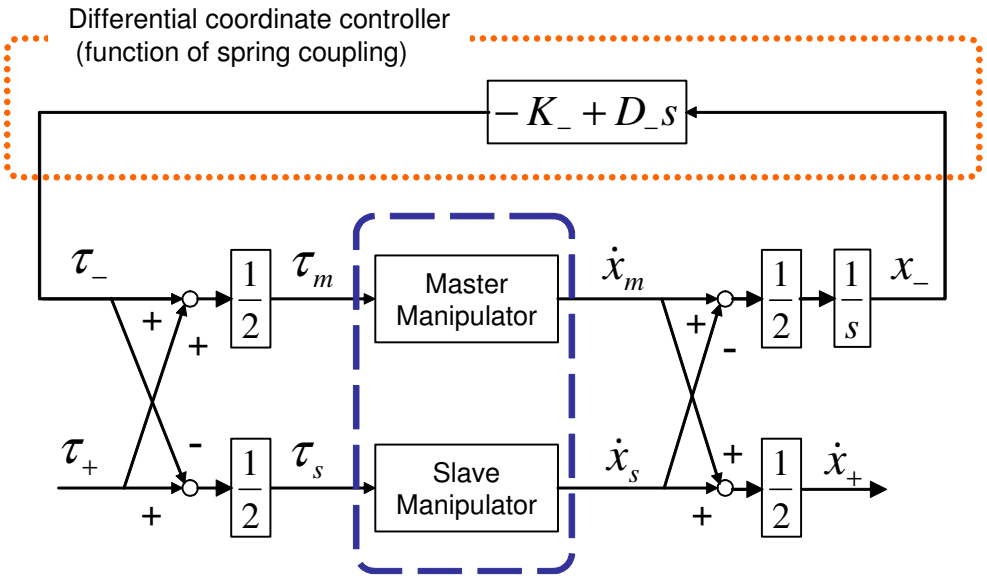


Fig.3-6: Spring coupling

Fig. 3-7 shows a PD controller with disturbance observer in the differential coordinate. This controller realizes a rigid coupling function. The disturbance observer in the differential coordinate is composed as shown in Fig. 3-8. Here, g denotes the cutoff frequency of the disturbance observer.

The input torque of this control system is figured out as follows:

$$\tau_- = -(K_- + D_- s)x_- - \frac{g}{s+g}(f_- - \mu_t \dot{x}_-). \tag{3.18}$$

Substituting (3.18) to (3.15),

$$M_t \ddot{x}_- + D_- \dot{x}_- + K_- x_- = \frac{s}{s+g}(f_- - \mu_t \dot{x}_-). \tag{3.19}$$

In low frequency range, every kind of disturbance including external force is completely canceled. Therefore, this function of coupling works as a rigid coupling. Since x_-, the gap of two manipulators, rapidly converges to zero and it would not be interfered with any other disturbances, it is reasonable to assume that stiffness of the function is infinity in the frequency range lower than the cutoff frequency of disturbance observer. At the same time, this function works as a spring coupling function in the higher frequency range since disturbance observer would not detect disturbance in the high frequency range.

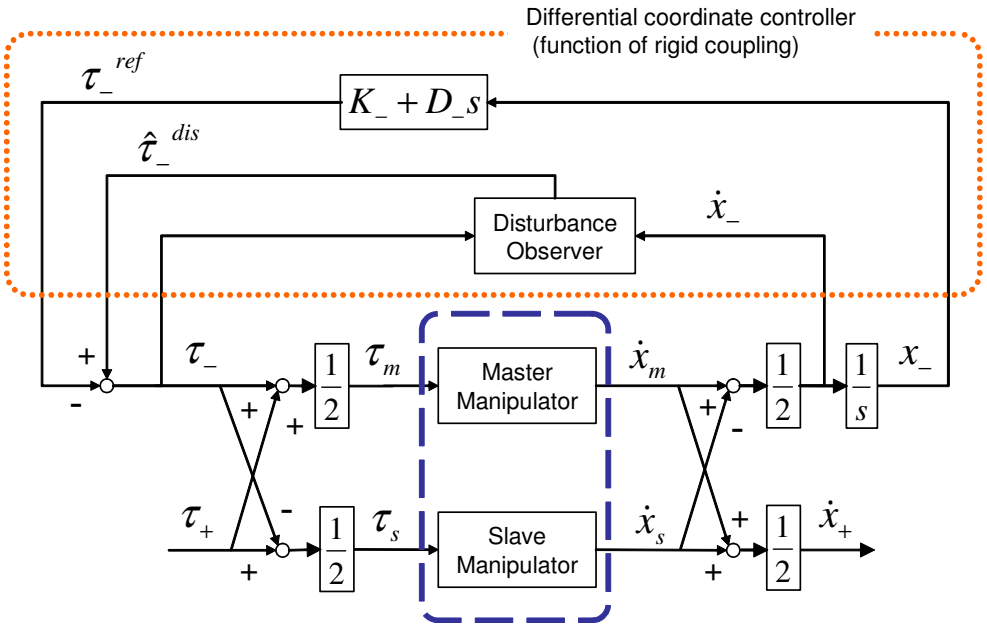


Fig.3-7: Rigid coupling

In the controllers shown above, there is no torque input in the common coordinate. This means no control is applied to common coordinate motion and both manipulators will move freely as the external force affects either manipulator.

Fig. 3-9 shows a controller based on a function of friction compensation. The controller in the differential coordinate works as a rigid coupling function. Sum of the friction torque is estimated and feedback input is given in the common coordinate.

Substituting $\tau_+ = (\mu_t - \mu_v)\dot{x}_+$ to (3.14),

$$M_t\ddot{x}_+ + \mu_v\dot{x}_+ = f_+ \tag{3.20}$$

where, μ_v is the virtual friction coefficient. (3.20) shows that the apparent friction coefficient is arbitrarily configured. For example, if the friction estimation is accurate and $\mu_v = 0$ is given, both manipulators move passively as if no friction occurs on the joint. Note that static friction remains, however, in practice.

Fig. 3-10 shows a controller based on a function of inertia manipulation. Sum of external force f_m and f_s is measured by external torque observer (ETOB) [59] in the common coordinate. τ_+,

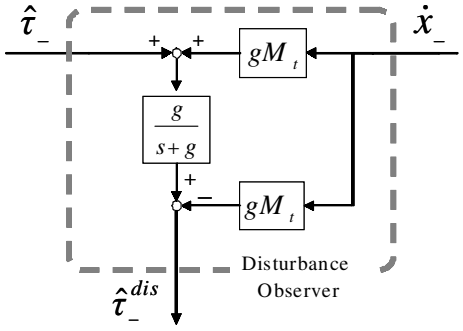


Fig.3-8: Disturbance observer in differential coordinate

torque input in the common coordinate, is figured out as follows:

$$\tau_+ = K_f f_+. \tag{3.21}$$

Substituting (3.21) to (3.14),

$$\begin{aligned} M_t\ddot{x}_+ + \mu_t\dot{x}_+ &= (1+K_f)f_+ \\ \frac{M_t}{1+K_f}\ddot{x}_+ + \frac{\mu_t}{1+K_f}\dot{x}_+ &= f_+ \\ M_v\ddot{x}_+ + \mu_v\dot{x}_+ &= f_+ \end{aligned} \tag{3.22}$$

where,

$$M_v = \frac{M_t}{1+K_f}, \mu_v = \frac{\mu_t}{1+K_f}.$$

Here, M_v denotes virtual mass realized by the inertia manipulation function.

As force feedback gain K_f becomes larger, virtual mass becomes smaller. Additionally, the virtual friction coefficient also becomes smaller. The inertia manipulation function interferes with the function of friction compensation since both functions exist in the same coordinate. The fact indicates that multiple functions in one coordinate should be designed as one combined system.

Fig. 3-11 shows a controller based on a position limit function. PD controller in the common coordinate works when the manipulators exceed a position limit. It pushes the manipulators back to the limit position. In other words, it represents a mechanical stopper with stiffness K_+ and viscosity D_+.

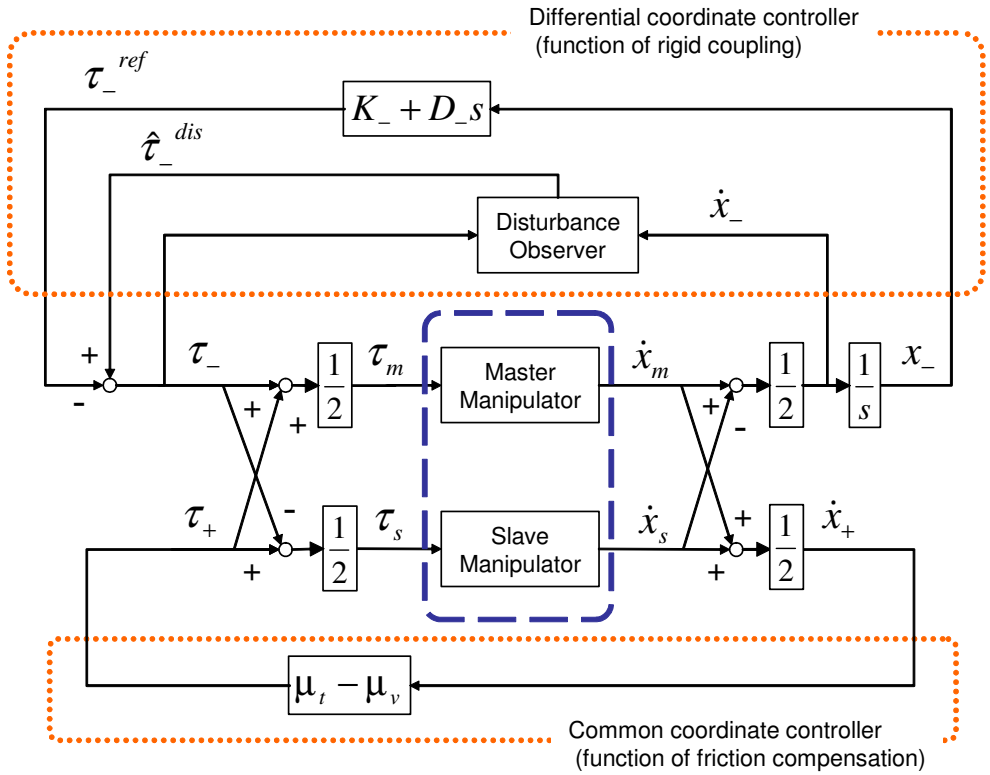


Fig.3-9: Friction compensation

Some other functions such as a function for gravity compensation and a function for velocity limit may also be required in some situations. Hence many kinds of functions are assumable. Additionally, any control scheme can be applicable to realize a function. In sum, the framework of functionality can deal with various architectures and various control schemes.

These examples of controllers show the simplicity and the explicitness of function-based controller design. Since a function is a minimum component, characteristics of the function are simple. Functions in different coordinates are completely independent to each other. Only when multiple functions exist in one coordinate, the interference of functions in the coordinate should be considered. The design problem is explicit since individual controllers correspond to each function without any interference of functions in the other coordinate.

The simplicity and the explicitness provide a way to design an adjustable system. The functions represent apparent mechanical properties such as inertia of the manipulator, friction of

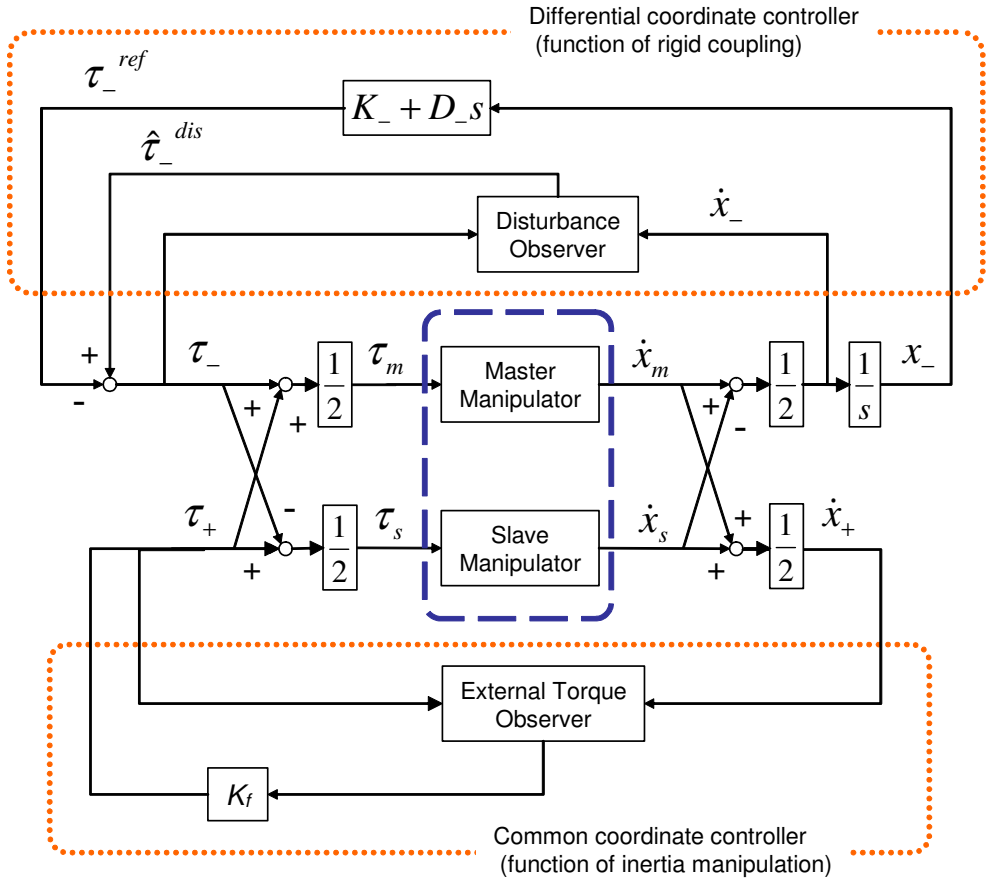


Fig.3-10: Inertia manipulation

the manipulator, stiffness of a stopper, and stiffness of a coupling. These properties are realized with the controllers designed in a simple and explicit way. Furthermore, task shifts are easily executed with this method. In case of the task shifts, conventional methods require redesign of the entire system while individual functions are easily mounted and unmounted in this method. The design problem is localized to individual function coordinates. It should be noted that the designer has to consider transient characteristics and smoothness of command values at the moment of task shift.

The framework in this study is based on the idea of modal decomposition. The modal decomposition method is already applied to some control systems such as mobile robots including wheel chairs [16], twin drive systems [51] and flywheels [60]. The framework is also applicable

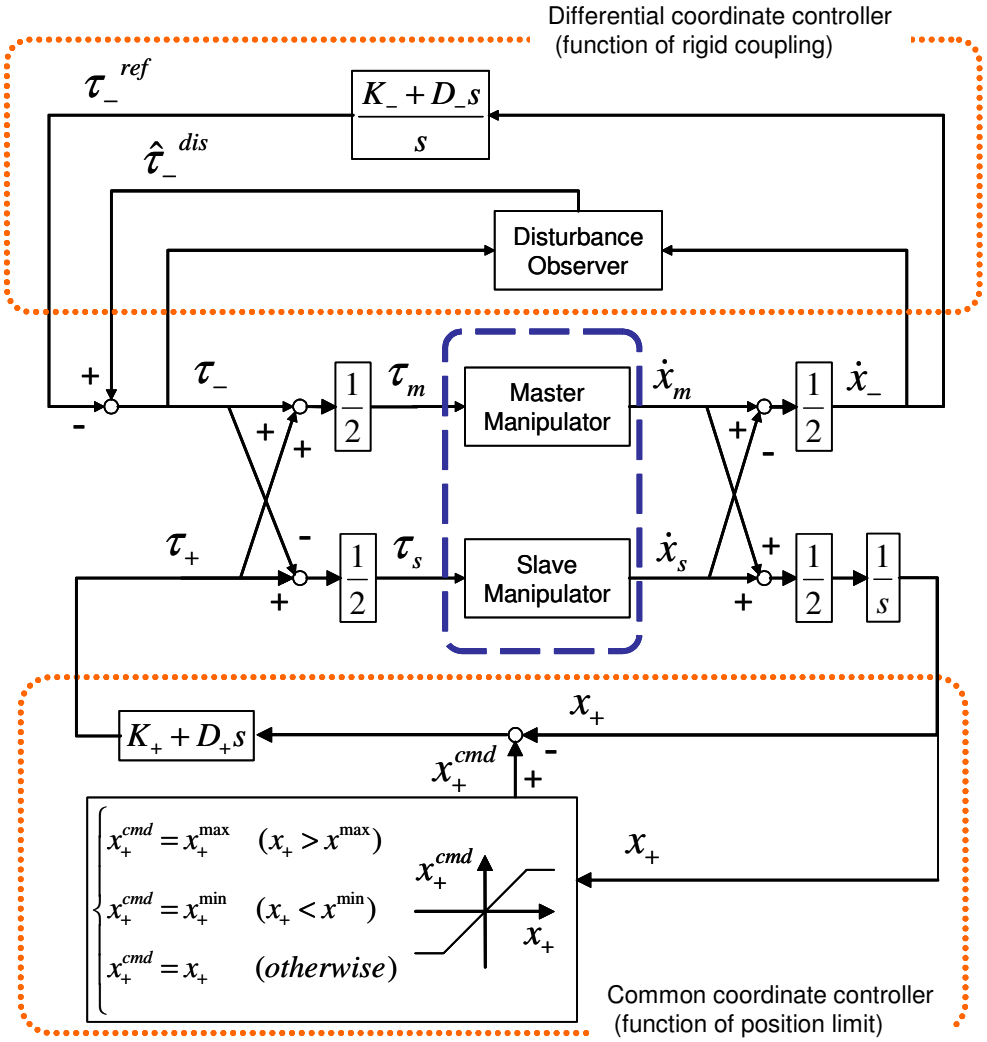


Fig.3-11: Position limit

to them although this chapter only deals with a bilateral control system.

3.5 Experiment

3.5.1 Description of experimental system

The overview of the bilateral control system is shown in Fig. 3-12. This experimental system is composed of two equivalent 1DOF manipulators connected to a PC through motor drivers.

The parameters of manipulators are shown in Table 3.1. A gravity term is negligible since the rotational plane of the manipulator is horizontal. The control parameters are shown in Table 3.2.

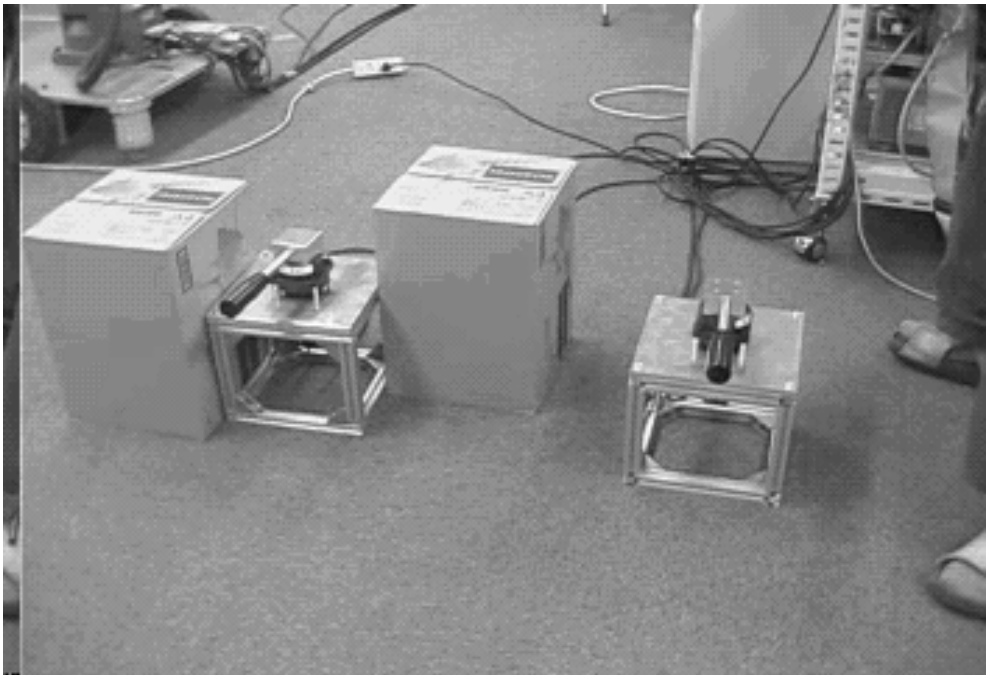


Fig.3-12: Experimental system

Table 3.1: Manipulator parameters

Arm length	[m]	0.16
Rated power output	[W]	50
Rated motor torque	[mNm]	159.0
Reduction ratio		1/33
Number of encoder pulse	[P/R]	2048
MOI at reducer output shaft	[kgm^2]	0.00535

Table 3.2: Control parameters

	Sampling period	[ms]	0.1
g	Cutoff frequency of DOB	[rad/s]	1100
g_f	Cutoff frequency of ETOB	[rad/s]	700
K_-	Position gain of coupling function		20
D_-	Velocity gain of coupling function		0.8
K_+	Position gain of position limit function		20
D_+	Velocity gain of position limit function		0.8
K_f	Force gain in common coordinate		1.5
x_{max}	Position limit	[rad]	0.3

Table 3.3: Functions in each stage

	Common coordinate	Differential coordinate
stage 1	none	Spring coupling
stage 2	none	Rigid coupling
stage 3	Friction compensation	Rigid coupling
stage 4	Friction compensation & Inertia manipulation	Rigid coupling
stage 5	Position limit	Rigid coupling

3.5.2 Experimental result

Fig. 3-13 compares positions of each manipulator while Fig. 3-14 shows the difference of them. Fig. 3-15 compares external force of the manipulators. Here, slave force is shown upside-down to compare the force in opposite directions. Tasks of the bilateral control system were shifted several times during the experiment. The combinations of functions are represented by five stages, from stage 1 to stage 5, as shown in Table. 3.3.

Stage 1 to stage 4 were switched arbitrarily by the operator. On the contrary, stage 5 starts accidentally when the manipulators exceed a position limit. Function-based controller design

deals not only with task shifts but also with exception handling. In each stage, the operator conducted two motions: a free motion and a touching motion. Firstly, he reciprocated the master manipulator twice in the free motion. Then, the slave manipulator followed the master one. Secondly, he pushed the obstacle in the slave side twice in the touching motion.

During the free motion in stage 1 and 2, the force responses of both manipulators did not accord each other since the master manipulator detected certain amount of force. The force is called manipulation force. It mainly consists of sum friction force and sum inertia force of the two manipulators. Therefore the manipulation force reduced in stage 3, the stage with a friction compensation function. Finally the manipulation force became the smallest in stage 4 since both friction force and inertia force were reduced.

During the touching motion in stage 1, pushing force increased in proportion to the position difference between the both manipulators. It shows that the spring coupling function worked the same as a mechanical spring coupling. During the touching motion in stage 2 to 5, force responses of both manipulators agreed very well. Position responses of both manipulators also agreed very well. These facts show that the rigid coupling function worked the same as a mechanical rigid coupling.

Fig. 3-14 shows that position difference of both manipulators slightly altered at the moment of contact. The amount of alteration in stage 2, 3 and 4 was almost the same while almost same amount of external force was applied in each stage. In sum, the functions in the common coordinate did not interfere with the rigid coupling function.

3.6 Summary of chapter

This chapter proposed an idea of "functionality", a new framework for a bilateral control system. Its features are summarized as follows:

- Function-based controller design is simple and explicit;
- A function represents an apparent mechanical property of a bilateral control system; and
- The decoupled design enables task shifts for various purposes.

Consequently, this framework is well suited for designing an adjustable bilateral system.

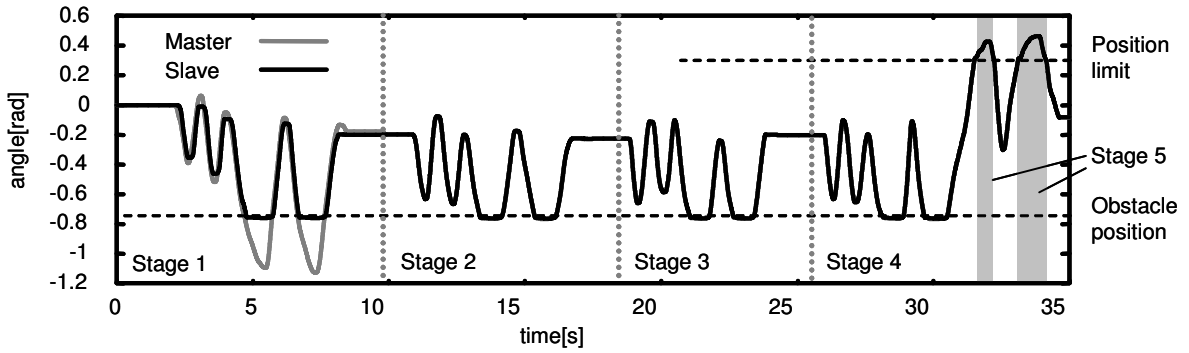


Fig.3-13: Position response

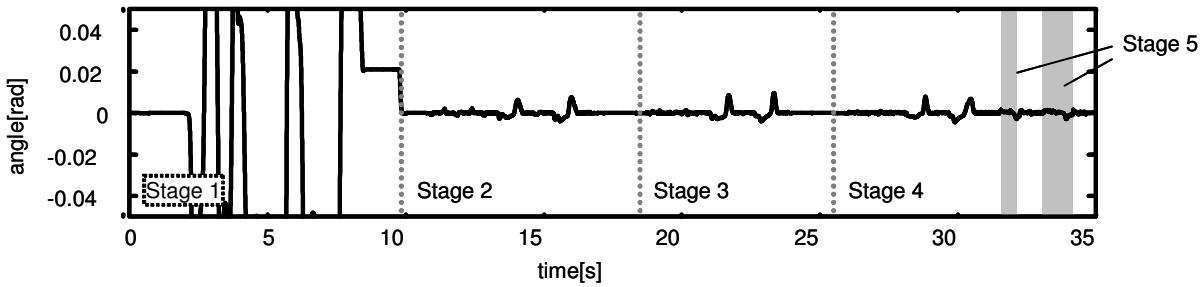


Fig.3-14: Difference of position response

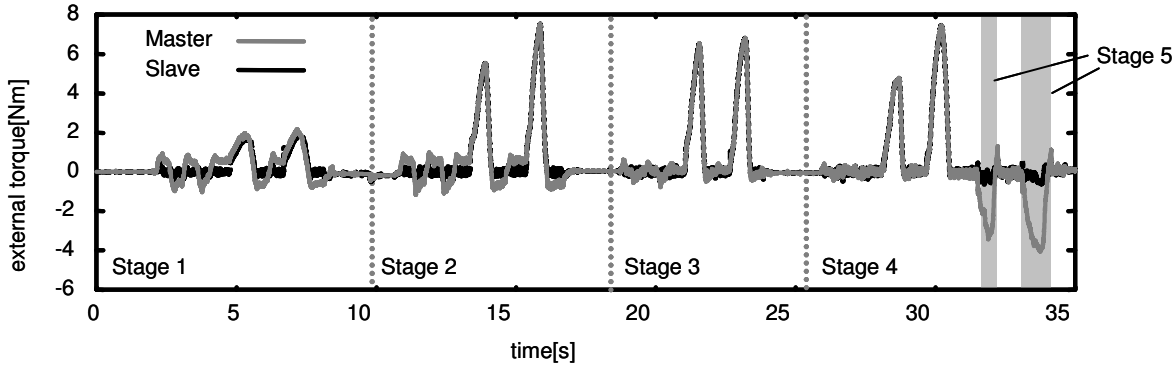


Fig.3-15: External force response

Chapter 4

Flexible design of decentralized control system

4.1 Introduction

The previous chapter introduced the idea of function. The idea helps decoupling the entire system into multiple subsystems for hybrid control. The hybrid control system acquires both manipulability and operationality as other parallel type controllers do [57, 61]. This chapter extends the idea to decentralized control.

The largest problem for motion control in human environment is complexity of controller design. There are several reasons for the complexity. Firstly, human environment has so many modes that the robot needs a hyper-DOF mechanism for adaptation. Secondly, the environmental modes keep changing in practice. Finally, a system role for a sophisticated control system also keeps changing. This chapter claims that the idea of functionality is a good candidate to solve the issue of the complexity. The framework based on functionality is extended to decentralized control.

Decentralized control is a promising method for large scale systems. It is preeminent in many features such as flexibility, fault tolerance, expandability, rapid response etc.. Many studies applied it to robot control systems [18, 23, 24, 62]. Among them, interesting concepts such as subsumption architecture [18], multi-agent system [62] and cell structure [24] have been proposed. Artificial intelligence is often introduced to solve the design issue of these methods. Decentralized control is also utilized for fault tolerant systems [63]. More explicit and simple framework in view of controller design is desired although the methods for decentralized control systems are interesting as concepts.

Arimoto and Nguyen [64] have shown that overall control input can be designed by linear superposition of all signals under the condition of unique stationary resolution of the controlled

position variables. Okada, Tatani, and Nakamura [22] have proposed a method to symbolize robot motion based on singular value decomposition.

This study aims at establishing a design framework for decentralized control systems that deals with task shift, faults, performance limits, and environmental variation in a unified manner. This paper has proposed function-based controller design for a bilateral control system in Chapter 3. In this chapter, the method is expanded to a decentralized control system with multiple subsystems.

Functionality is an idea to divide the system role into components. The proposed method is novel in the point that individual controllers are directly related to functions instead of control objects. This idea solves the problem of design complexity in decentralized control systems.

This chapter is organized as follows: The concept of a function for decentralized control is shown by a 1-dimensional example in Section 4.2. Section 4.3 shows examples and some interesting features of functions. In Section 4.4, design procedures are explained step by step in detail. Section 4.5 shows simulation results and experimental results. The concept is expanded for a 3-dimensional system in Section 4.6. Section 4.7 describes how to implement cooperative grasping motion with a three-dimensional function-based control system. Section 4.8 shows experimental results of the cooperative grasping motion. Finally, this study is concluded in Section 4.9.

4.2 Concept of functionality for decentralized control

4.2.1 Supposed system

Fig. 4-1 shows a supposed control system to explain functions in a simple example of a 1-dimensional system. It consists of four robots named robot_1, robot_2, robot_3 and robot_4 respectively.

Two robots make a pair and each pair grasps a load. The pair of robot_1 and robot_2 grasps load_1 while the pair of robot_3 and robot_4 grasps load_2. Distance between two pairs is controlled to be constant. An operator gives the command force to the load or the robot. Operator force is detected by reaction force observer (RFOB) [59] and assisted. Therefore the operator feels the load lighter. The operator can detect the collision in the further site because both pairs compose a kind of bilateral control system. It is assumed that communication among

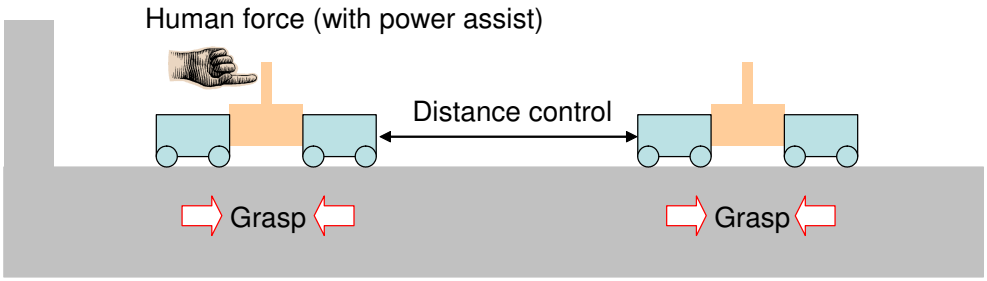


Fig.4-1: Supposed system

Table 4.1: Functions for robot control system

Type	Role of function	Based information
grasp	apply grasp force to a load	force difference
coupling	control distance of two subsystems	position difference
friction compensation	compensate friction of entire system	velocity sum
Inertia manipulation	assist manipulation force from human	force sum
torque limit	limit excessive input	torque of a robot
velocity limit	slow down overspeed actuator	velocity of a robot
position limit	avoid collisions and prohibited area	position of a robot

robots is complete.

4.2.2 Coordinate space based on function

This subsection describes relationship between functions and robots. Then a new coordinate space based on functions is introduced to sort out this relationship. In this study, a control system in human environment is divided into independent functions and the combination of functions composes a large hybrid control system. The examples of functions for decentralized control are listed in Table 4.1. As shown in the table, majority of the functions are based on information of multiple robots. It also needs to provide inputs to the robots at the same time. Hence, function-based information should be extracted from each robot so as to design function-

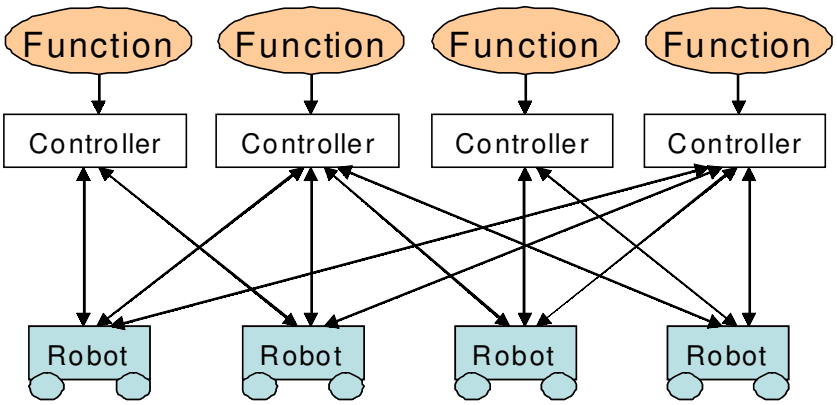


Fig.4-2: Relationship between functions, controllers and robots

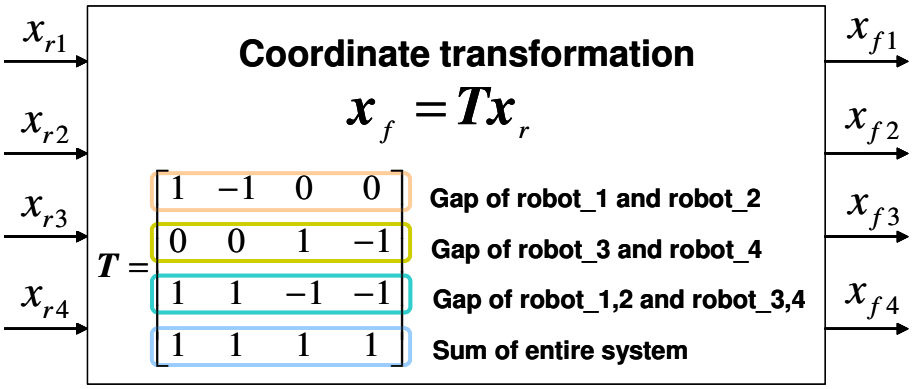


Fig.4-3: Outline of coordinate transformation

based controllers. Function-based controllers have complicated relationship with robots while they are related directly with functions as shown in Fig. 4-2.

In this study, robot information is transformed to new coordinate space based on functions to associate function-based controllers with control objects. The controllers are designed in the new coordinate space based on functions. This new coordinate space is called function coordinate space. On the other hand, the original coordinate space is called robot coordinate space. The outline of the coordinate transformation is shown in Fig. 4-20.

The overview of the control system is shown in Fig. 4-4. The robot coordinate is transformed

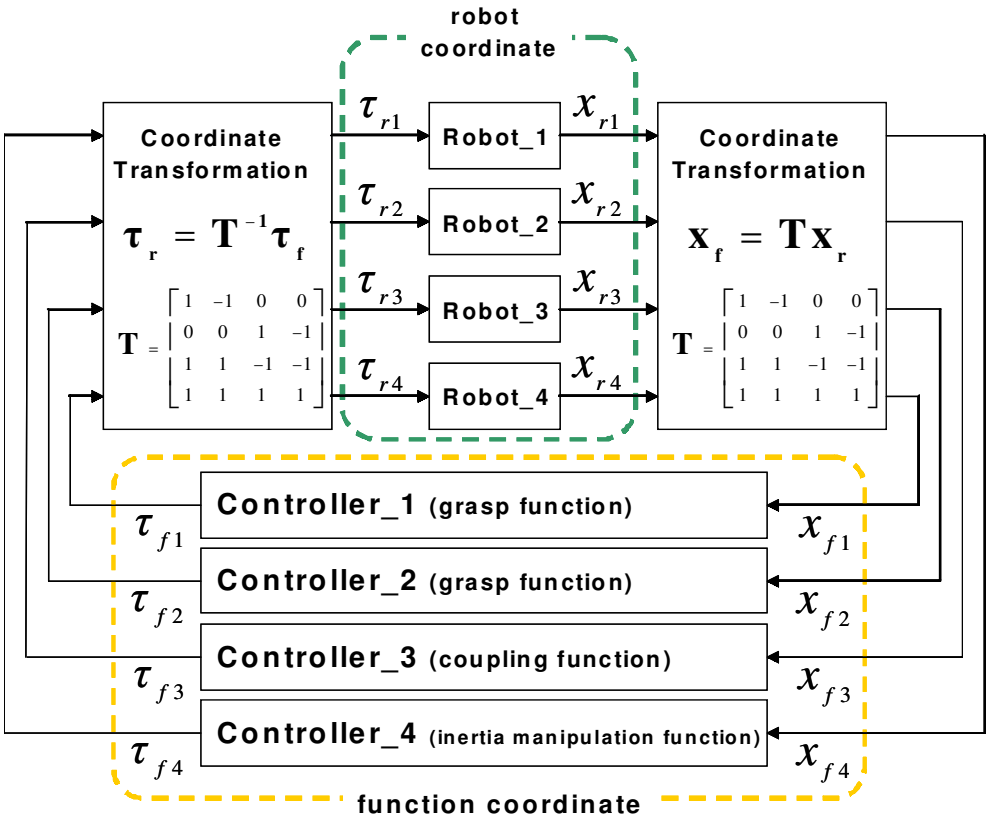


Fig.4-4: Overview of control system

to function coordinate by a transformation matrix.

The coordinate transformation from robot coordinates to function coordinates is shown in (4.1).

$$\begin{aligned} \boldsymbol{x}_f &= \boldsymbol{T}\boldsymbol{x}_r \\ \boldsymbol{x}_f &= \begin{bmatrix} x_{f1} & x_{f2} & \cdots & x_{fN} \end{bmatrix}^T \\ \boldsymbol{x}_r &= \begin{bmatrix} x_{r1} & x_{r2} & \cdots & x_{rM} \end{bmatrix}^T \end{aligned} \tag{4.1}$$

where x is the position information. Subscript r denotes the robot coordinate and f denotes the function coordinate. Numbers in subscripts denote the order of functions or robots. N is the total number of mounted functions. M is the total number of robots. $\boldsymbol{T}$ shows the transformation matrix. The method to derive $\boldsymbol{T}$ is described later.

Velocity, acceleration, input force $\boldsymbol{\tau}$ and external force $\boldsymbol{f}$ are all transformed in the same way.

$$\dot{\boldsymbol{x}}_f = \boldsymbol{T}\dot{\boldsymbol{x}}_r \tag{4.2}$$

$$\ddot{\boldsymbol{x}}_f = \boldsymbol{T}\ddot{\boldsymbol{x}}_r \tag{4.3}$$

$$\boldsymbol{\tau}_f = \boldsymbol{T}\boldsymbol{\tau}_r \tag{4.4}$$

$$\boldsymbol{f}_f = \boldsymbol{T}\boldsymbol{f}_r \tag{4.5}$$

4.2.3 Dynamics in function coordinate space

The dynamic equation in function coordinate space is derived to show that it is proper to design the controllers in the function coordinate space.

The dynamic equation in the original robot coordinate space is as follows:

$$\begin{aligned} \boldsymbol{m}_r\ddot{\boldsymbol{x}}_r + \boldsymbol{\mu}_r\dot{\boldsymbol{x}}_r &= \boldsymbol{\tau}_r + \boldsymbol{f}_r \qquad (4.6)\\ \boldsymbol{m}_r &= \begin{bmatrix} m_{r1} & & & \\ & m_{r2} & & \\ & & \ddots & \\ & & & m_{rM} \end{bmatrix} \\ \boldsymbol{\mu}_r &= \begin{bmatrix} \mu_{r1} & & & \\ & \mu_{r2} & & \\ & & \ddots & \\ & & & \mu_{rM} \end{bmatrix}. \end{aligned}$$

Here, m_r stands for the mass of robots and μ_r stands for the friction coefficient. The numbers in subscripts denote the robot number.

Assuming if all of the robots have the same mechanism,

$$m\ddot{\boldsymbol{x}}_r + \mu\dot{\boldsymbol{x}}_r = \boldsymbol{\tau}_r + \boldsymbol{f}_r \tag{4.7}$$

where $m = m_{r1} = m_{r2} = \cdots = m_{rM}$ and $\mu = \mu_{r1} = \mu_{r2} = \cdots = \mu_{rM}$.

Then the dynamic equation in the function coordinate space is derived as (4.8).

$$\begin{aligned} \boldsymbol{T}m\ddot{\boldsymbol{x}}_r + \boldsymbol{T}\mu\dot{\boldsymbol{x}}_r &= \boldsymbol{T}\boldsymbol{\tau}_r + \boldsymbol{T}\boldsymbol{f}_r \\ m\boldsymbol{T}\ddot{\boldsymbol{x}}_r + \mu\boldsymbol{T}\dot{\boldsymbol{x}}_r &= \boldsymbol{T}\boldsymbol{\tau}_r + \boldsymbol{T}\boldsymbol{f}_r \\ m\ddot{\boldsymbol{x}}_f + \mu\dot{\boldsymbol{x}}_f &= \boldsymbol{\tau}_f + \boldsymbol{f}_f \qquad (4.8) \end{aligned}$$

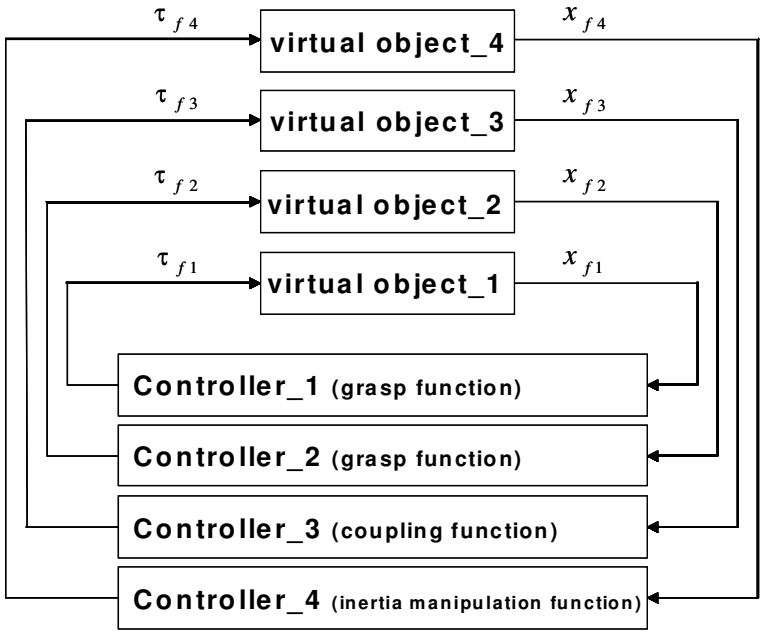


Fig.4-5: Control system in function coordinate space

It shows that the input from the function-based controller works as if it is controlling a virtual object in the function coordinate space. It also shows that the dynamics in each coordinate does not interact with each other. Therefore, the control system shown in Fig. 4-4 could be treated as a decoupled system shown in Fig. 4-5. As shown in the dynamic equation, essential behavior of the function in the function coordinate space is simple although one function-based controller affects multiple robots. The behavior carried out by each function-based controller is named "function mode". The behavior of the entire system in robot coordinate space shows up as superposition of function modes.

4.3 Example and feature of function

This section shows examples of functions and categorizes them into two types: task function and performance limit function. Task functions are to execute the required tasks. Performance limit functions are to deal with performance limits. Some interesting features of these functions are described after the examples.

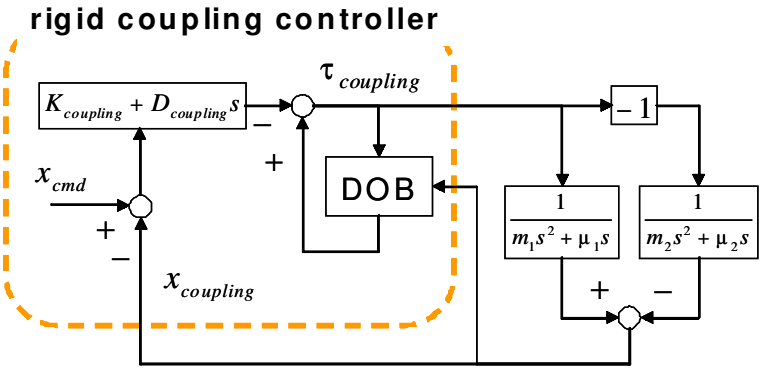


Fig.4-6: Rigid coupling controller with actual objects

4.3.1 Examples of task functions

Rigid coupling function

Fig. 4-7 shows a controller for a rigid coupling function. This controller provides an input so as to control the position difference of two control objects. x_{RC} shows the position of the virtual object, that is position difference of two control objects in fact. τ_{RC} shows the input force to the virtual object. When the rigid coupling is the ith function, x_{RC}, τ_{RC} and f_{RC} stands for x_{fi}, τ_{fi} and f_{fi} in (4.8) respectively.

Equation (4.9) shows the force input from the rigid coupling controller. Equation (4.10) is derived substituting (4.9) to (4.8). This equation shows the function mode of the rigid coupling function. It realizes rigid coupling in the frequency range lower than g, the cutoff frequency of disturbance observer. It works as a spring coupling with stiffness K_{RC} and friction coefficient D_{RC} in the frequency range higher than g.

$$\tau_{RC} = -(K_{RC} + D_{RC}s)x_{RC} - \frac{g}{s+g}(f_{RC} - \mu\dot{x}_{RC}) \tag{4.9}$$

$$m\ddot{x}_{RC} + D_{RC}\dot{x}_{RC} + K_{RC}x_{RC} = \frac{s}{s+g}(f_{RC} - \mu\dot{x}_{RC}) \tag{4.10}$$

Here subscript RC denotes the rigid coupling function.

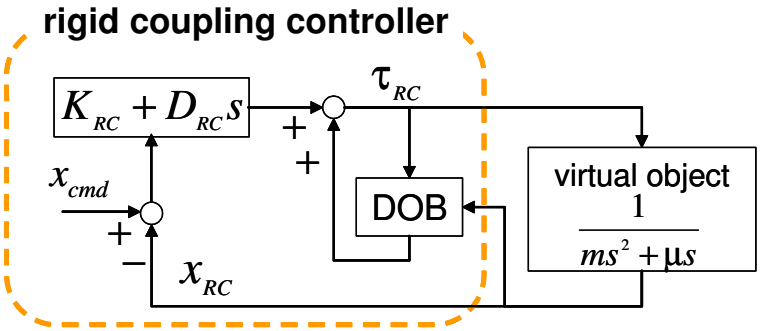


Fig.4-7: Rigid coupling controller with virtual object

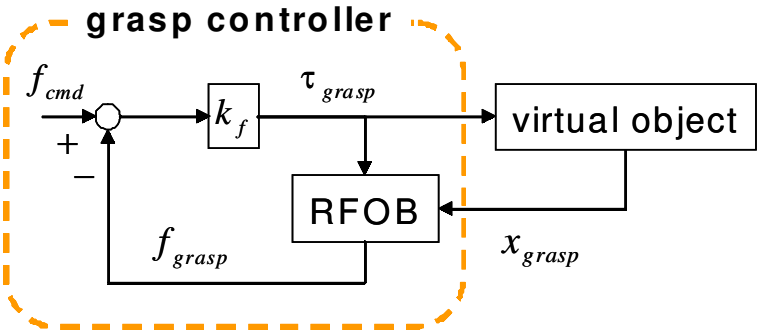


Fig.4-8: Grasp controller

Grasp function

Fig. 4-8 shows a controller for a grasp function. Here, subscript *grasp* denotes the grasp function and k_f shows the force gain. It controls the internal force on a load between two robots. This function is based on the difference of external force of two robots. The robots can keep grasping while disturbance on the load is smaller than $\frac{1}{2}\tau_{cmd}$.

Inertia manipulation function

Fig. 4-9 shows a controller for an inertia manipulation function. Here, subscript IM denotes the inertia manipulation function. It assists the external force on a virtual object. The virtual object may correspond to a robot or multiple robots. The assist force is given by (4.11).

$$\tau_{IM} = k_f f_{IM} \tag{4.11}$$

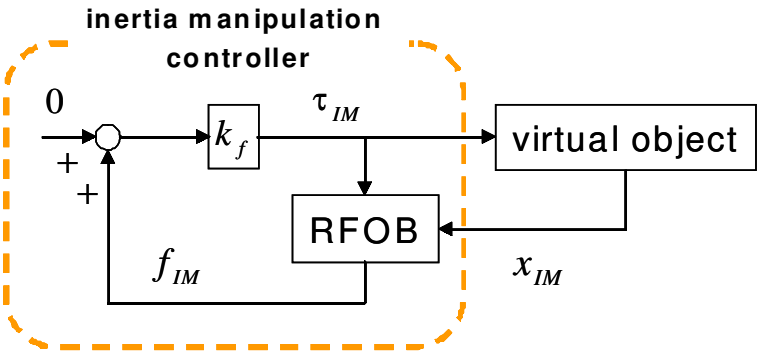


Fig.4-9: Inertia manipulation controller

Substituting (4.11) to (4.8),

$$\begin{aligned} m\ddot{x}_{IM} + \mu\dot{x}_{IM} &= (1+k_f)f_{IM} \\ \frac{m}{1+k_f}\ddot{x}_{IM} + \frac{\mu}{1+k_f}\dot{x}_{IM} &= f_{IM} \\ m_v\ddot{x}_{IM} + \mu_v\dot{x}_{IM} &= f_{IM} \\ m_v = \frac{m}{1+k_f}, \quad \mu_v &= \frac{\mu}{1+k_f}. \end{aligned} \tag{4.12}$$

As shown in (4.12), the inertia manipulation function realizes virtual mass m_v and virtual friction coefficient μ_v. k_f manipulates their values.

4.3.2 Examples of performance limit function

Position limit function

Fig. 4-10 shows a position limit controller. Here, subscript PL denotes the position limit function. It is a PD controller with conditions. When a robot goes out of its working area, position limit controller works so as to bring back the robot into the working area. This function is mainly applied for a collision avoidance.

Velocity limit function

Fig. 4-11 shows the velocity limit controller. Here, subscript VL denotes a velocity limit function. It is a velocity controller with conditions. When a robot runs over the velocity limit, it decreases the velocity.

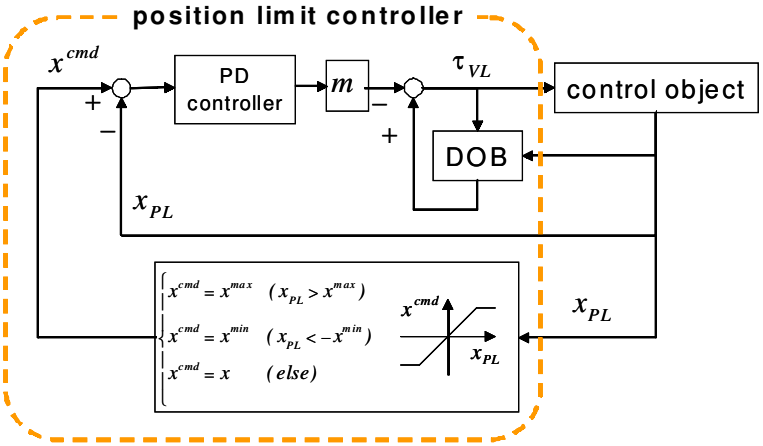


Fig.4-10: Position limit controller

Torque limit function

Fig. 4-12 shows a torque limit controller. Here, subscript TL denotes the torque limit function. This controller does not have a feedback loop. It only modifies the torque input when reference torque exceeds the torque limit. In fact, torque limit controller itself does nothing to handle exceptions, however, it helps to allocate torque inputs by becoming active. When it becomes active, other functions do not provide excessive inputs to the robot that exceeds torque limit. In this study, it modifies the force input instead of the torque input since we suppose linear actuators.

4.3.3 Function activity

Performance limit functions work under regulated conditions. They provide inputs when the state value of the system exceeds the performance limit. We define this state of functions as "active". On the other hand, function-based controllers do not provide any inputs when the system state is within the performance limit range. This state is named "inactive".

Extra DOF of the system arises when a function is inactive because no input is given from the function. Therefore it becomes possible to add other functions when inactive functions exist. On the contrary, some functions could not be executed when the total DOF of functions exceeds the total DOF of robots. The state of a function unexecuted due to limited DOF is named

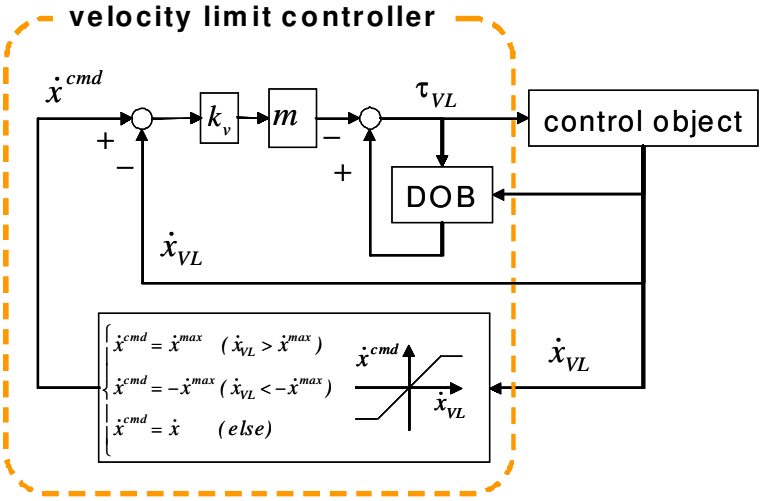


Fig.4-11: Velocity limit controller

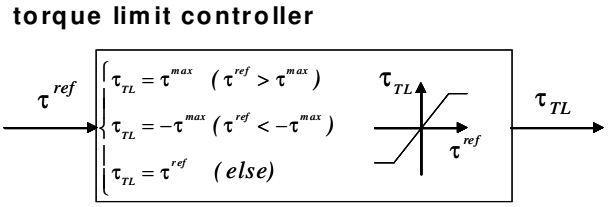


Fig.4-12: Torque limit controller

"standby".

4.3.4 Exception handling

This method deals with faults and environmental variations by manipulating the parameter of the performance limit function. For example, the velocity limit should be set to 0 m/s when the gear of the robot is jammed. A position limit should be given when an unknown object comes in. An actuator breakdown is described as a state of a 0 Nm torque limit. This kind of faults and environmental variations are treated as exceptions. The state beyond the performance limit is also treated as an exception. In this way, exceptions such as faults, performance limits and environmental variations are handled in a unified manner.

In conventional methods, exception handling required much effort since every combination of multiple exceptions needs programming. In this method, each exception corresponds to each performance limit function without any interferences. Hence design effort of exception handling becomes small.

4.4 Procedure of design

Previous section showed the examples of functions and introduced significant features of them. This section describes the procedure of design based on these features. The design flow of function-based control system is shown in Fig. 4-22. Each step is described in detail in this section.

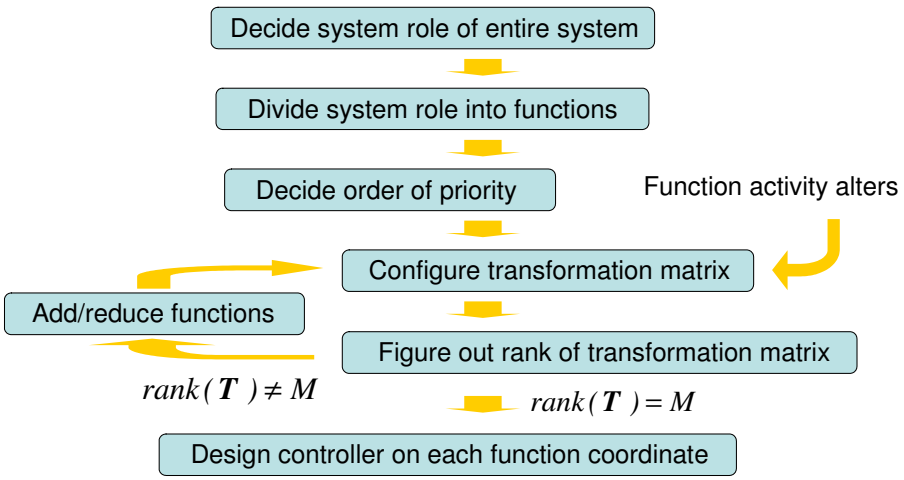


Fig.4-13: Flow of controller design

4.4.1 Specification of system role

As a first step to design a function-based control system, the system role of the system should be decided and described. The system role should cover the tasks of the system, fault tolerance and performance limits.

For example, the system role of the supposed system is described as follows:

“Transparent bilateral control between two pairs is achieved while each pair is grasping the

load. Robots operate within the performance limits such as velocity limits and torque limits. No collision between robots occurs. ”

4.4.2 Division of system role

As shown in the previous example, the system role is described in words that are useful expression for people. although it is abstract and complicated. Now it should be described in a physical fashion to associate with controllers. It makes the controller design explicit. It is also effective to divide the system role into minimum units to simplify each controller. Therefore it is divided to functions.

The system role described above is divided into several functions as follows:

- Grasp function for robot_1 and robot_2
- Grasp function for robot_3 and robot_4
- Rigid coupling function for pairs of (robot_1, robot_2) and (robot_3, robot_4)
- Inertia manipulation function for entire system
- Torque limit functions for each robot
- Velocity limit functions for each robot
- Position limit functions for each robot.

4.4.3 Setup of function priority

Performance limit functions become active when exceptions occur. Therefore the number of active functions often alters during the operation. Standby functions should be selected to sustain important functions when the number of active functions increases. Hence we introduce the concept of priority order.

Basis to make a priority order are as follows:

1. Mechanical limit
2. Safety
3. Importance of the task.

Functions to deal with mechanical limits are the most important since mechanical limits are the absolute condition of a robot control system. Most of the performance limit functions belong to this category. Functions to assure the safety comes the next because safety is the priority for users. A function for collision avoidance and others belong to this category. A grasp function also has a high priority since it is dangerous to drop the grasped load in some situations. Except these, function priority is given from the importance of the task.

The priority order of functions in the supposed system becomes as follows: torque limit functions, velocity limit functions and position limit functions come first since they are to deal with mechanical limits. Two grasp functions come in the next place to assure the safety. The order of other two functions are given from the importance of the task. The importance is decided arbitrary.

- Torque limit functions for each robot
- Velocity limit functions for each robot
- Position limit functions for each robot
- Grasp function for robot_1 and robot_2
- Grasp function for robot_3 and robot_4
- Rigid coupling function for pairs of (robot_1, robot_2) and (robot_3, robot_4)
- Inertia manipulation function for entire system

4.4.4 Configuration of transformation matrix

This section describes how to configure the transformation matrix $\boldsymbol{T}$. As described in Section 4.2.2, a function mode affects multiple robots while the function mode itself is simple. Transformation vector stands for the proportion of the affection from the function mode to the robots. $\boldsymbol{T}_{function}$ in (4.13) shows the transformation vector.

$$\boldsymbol{T}_{function} = \begin{bmatrix} T_1 & T_2 & \dots & T_N \end{bmatrix} \tag{4.13}$$

For example, a function mode of a rigid coupling function between robot_1 and robot_2 shows up to the position difference $x_{r1} - x_{r2}$. Since the function is based on the differential information

of control objects, 1 and -1 are substituted respectively to the elements related to the control objects. Equation (4.14) shows the transformation vector of this example.

$$\boldsymbol{T}_{RC} = \begin{bmatrix} 1 & -1 & 0 & 0 \end{bmatrix} \tag{4.14}$$

Transformation vectors of all functions are figured out in this way. When the function is based on the sum information of control objects, 1 is substituted to the related element. Other elements not related to the functions are 0.

A transformation matrix $\boldsymbol{T}$ is figured out based on the functions in priority order.

$$\boldsymbol{T} = \begin{bmatrix} \boldsymbol{T}_{function1} \\ \boldsymbol{T}_{function2} \\ \boldsymbol{T}_{function3} \\ \boldsymbol{T}_{function4} \end{bmatrix} \tag{4.15}$$

Transformation vectors of the active functions are ranked in the priority order. The transformation matrix should be reconfigured when activity of any functions alters. Transformed coordinates do not interfere with each other when rows of the transformation matrix are independent. Rank of the transformation matrix should be figured out to confirm whether functions are independent to each other. Equation (4.16) is the condition to show that functions are independent to each other .

$$rank(\boldsymbol{T}) = N \tag{4.16}$$

On the other hand, (4.17) shows the condition that the sum DOF of the functions and the sum DOF of the robots are equal.

$$rank(\boldsymbol{T}) = M \tag{4.17}$$

If $rank(T) < M$, it is possible to add other functions since redundancy exists. Rank of the transformation matrix is also utilized for deciding the number of mounted functions.

4.5 Experiment and Simulation in one-dimensional system

4.5.1 Simulation for normal conveying operation

We verified the performance of the proposed control system in simulation. Table 4.2 shows parameters on the simulation.

Table 4.2: Parameters of control system in simulation

m	Weight of robot [Kg]	2.0
m_l	Weight of load [Kg]	0.5
	Sampling period [ms]	0.1
g	Cutoff frequency of DOB[rad/s]	100.0
K_{RC}	Position gain for rigid coupling function	600.0
D_{RC}	Velocity gain for rigid coupling function	70.0
k_f	Force gain for grasp function	1.0
	and inertia manipulation function	

Fig. 4-14 shows the results when external force $F_{ext} = 0.28 \times \sin(t)\,\mathrm{N}$ was given to load_1. The transformation matrix in normal condition is shown as follows:

$$\boldsymbol{T} = \begin{bmatrix} \boldsymbol{T}_{grasp1} \\ \boldsymbol{T}_{grasp2} \\ \boldsymbol{T}_{RC} \\ \boldsymbol{T}_{IM} \end{bmatrix} = \begin{bmatrix} 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & 1 & 1 & 1 \end{bmatrix}. \tag{4.18}$$

Force responses of robot_1 and robot_2 stand for the grasp force on load_1. Similarly, force responses of robot_3 and robot_4 stand for the grasp force on load_2. It is shown from the result that grasp functions were successfully executed on both loads. Force responses fluctuate due to the external force, however, they never became lower than 0. Position responses and velocity responses show that the coupling function and the inertia manipulation function worked well. The distance between both pairs was controlled to be constant despite of the existence of the external force.

4.5.2 Simulation for conveying operation over velocity limit

In the next place, external force $F_{ext} = 0.35 \times \sin(t) + 0.06\,\mathrm{N}$ was given to load_1. Then velocity went over the limit of robot_1, 2.0 m/s. With 0.1 m/s of safety margin, the velocity limit function became active when the velocity response exceeded 1.9 m/s. From the priority order, the velocity limit function was the first function while it is active. Two grasp functions and a coupling function took back seats. The inertia manipulation function was standby due to the excess of functions. The result is shown in Fig. 4-15. The transformation matrix when the

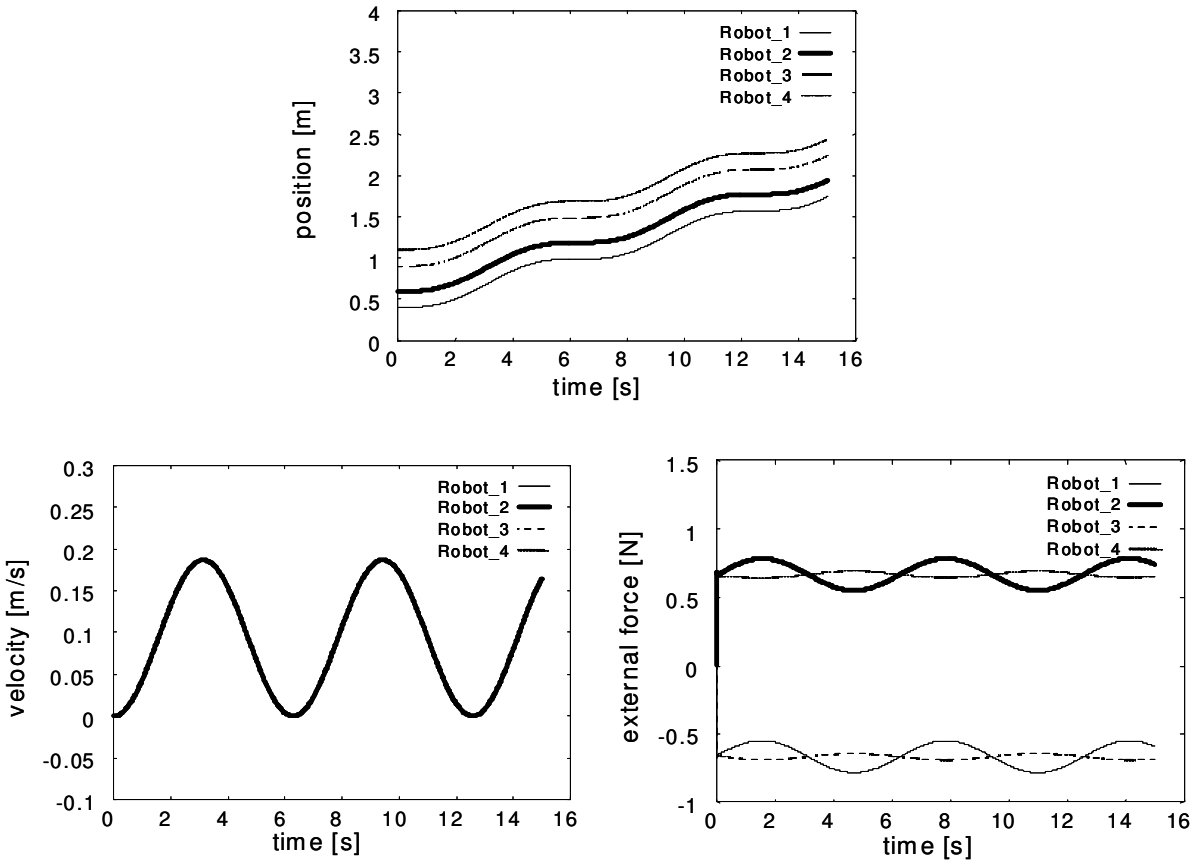


Fig.4-14: Simulation result on conveying operation (position, velocity and force response)

velocity limit function became active is shown as follows:

$$\boldsymbol{T} = \begin{bmatrix} \boldsymbol{T}_{VL} \\ \boldsymbol{T}_{grasp1} \\ \boldsymbol{T}_{grasp2} \\ \boldsymbol{T}_{RC} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & -1 \\ 1 & 1 & -1 & -1 \end{bmatrix}. \tag{4.19}$$

Velocity responses show that robot_1 velocity never overran the limit. Velocity of other robots were also repressed in tune with the robot_1 so as to achieve grasp and coupling functions. Force response show that the grasp function was executed without any failure even when the velocity limit function became active. Slight error occurred on force response when the velocity limit function got active because of the rapid input variation in the velocity limit function. This fact indicates the need of input smoothness. In sum, these results show that changeover of the

function rarely interfere the other functions if smooth inputs are provided.

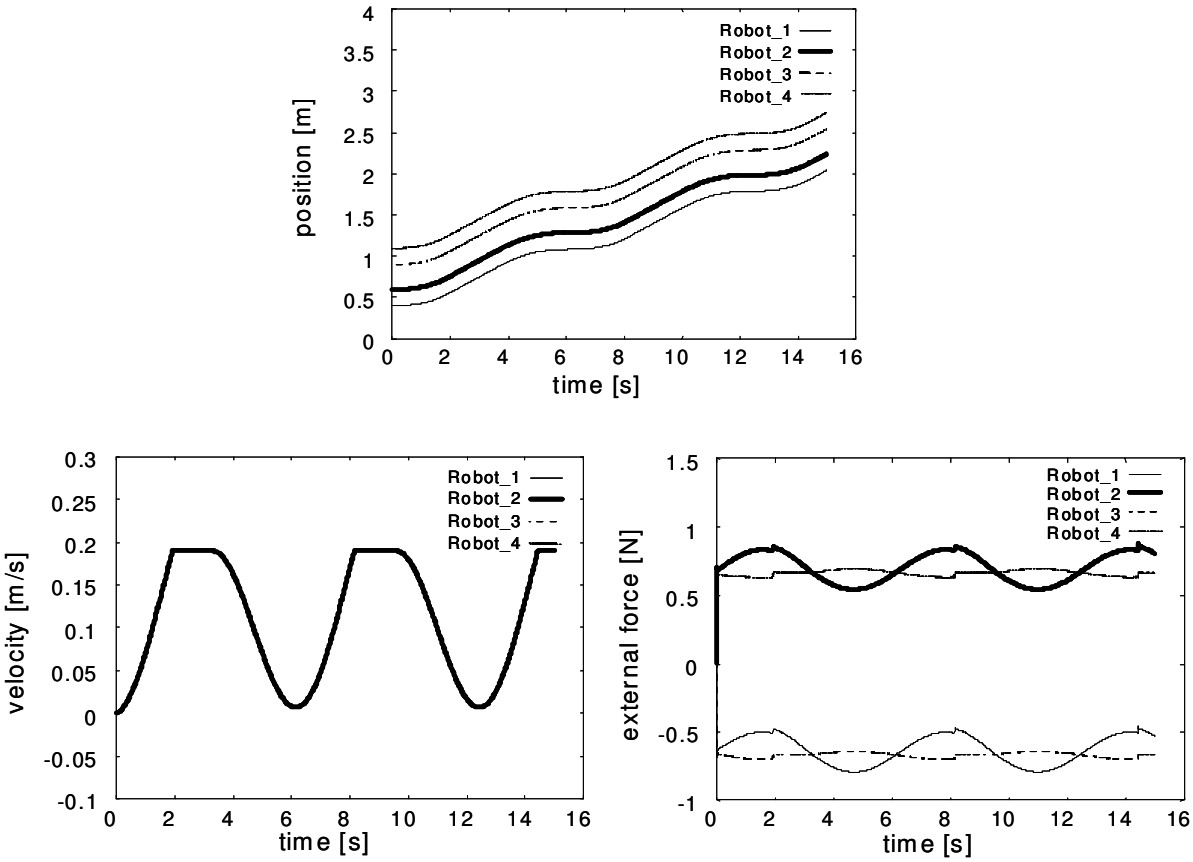


Fig.4-15: Conveying operation over velocity limit (position, velocity and force response)

4.5.3 Simulation for conveying operation with an actuator breakdown

We realized the situation that the actuator on a robot_4 broke down while robot_1 is running at maximum velocity. It is able to handle this kind of a fault situation by setting the torque limit of distressed robot as 0. A torque limit function on the distressed robot became active.

The transformation matrix then is shown as (4.20).

$$\boldsymbol{T} = \begin{bmatrix} \boldsymbol{T}_{TL} \\ \boldsymbol{T}_{VL} \\ \boldsymbol{T}_{grasp1} \\ \boldsymbol{T}_{RC} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 1 & 1 & -1 & -1 \end{bmatrix} \tag{4.20}$$

The results are shown in Fig. 4-16.

The grasp function for load_2 was replaced since the function cannot be executed without the input force on robot_4. The torque limit function on robot_4 was active instead.

At the same time, the velocity of robot_1 came to the limit. Then the velocity limit function on robot_1 became active and the inertia manipulation function was replaced.

The position response and the force response show that the grasp function on load_1 and the coupling function worked out for several faults. It shows that function-based controller design attained fault tolerant control system. Based on the priority order, the control system sustained the important functions and exception handling functions became active while low priority functions were replaced. Consequently the entire control system could keep the operation with the minimal task reduction under existence of multiple faults.

4.5.4 Experiment for grasping motion

In order to show the validity of function-based controllers, an experiment is executed on the experimental system shown in Fig. 3-12. Table 4.3 denotes the parameters of the manipulators. Fig. 4-17 is the illustration of the experiment. Fig. 4-18 shows the position information during the experiment. Additionally, Fig. 4-19 shows the force information.

Steps on the experiment are as follows. Here, t shows the time.

1) $t < 5$ A rigid coupling function and an inertia manipulation function were applied. The transformation matrix was

$$\boldsymbol{T} = \begin{bmatrix} \boldsymbol{T}_{RC} \\ \boldsymbol{T}_{IM} \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}. \tag{4.21}$$

Manipulators were moved freely by the hand.

2) $5 < t < 12$ There was no changeover of tasks but a load was pinched by the hand motion.

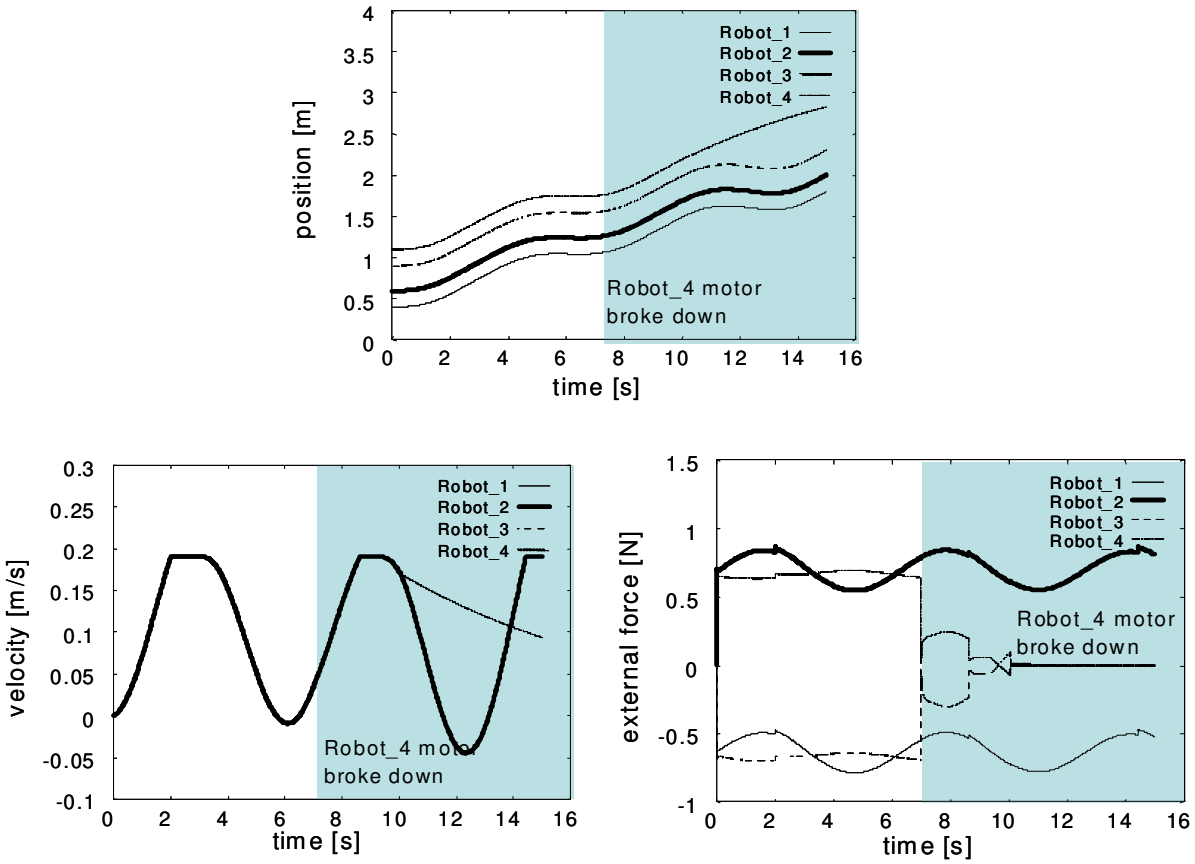


Fig.4-16: Conveying operation with an actuator breakdown(position, velocity and force response)

3) $t = 12$ The inertia manipulation function was shifted to a grasp function. The inner force value at the moment of task changeover was given as a command grasping force. It was about 1.2 Nm. The transformation matrix became

$$\boldsymbol{T} = \begin{bmatrix} \boldsymbol{T}_{RC} \\ \boldsymbol{T}_{grasp} \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}. \tag{4.22}$$

4) $12 < t < 18$ No changeover of tasks occurred. The hand was taken off from the manipulators.

5) $t = 18$ The rigid coupling function was shifted to an inertia manipulation function. The

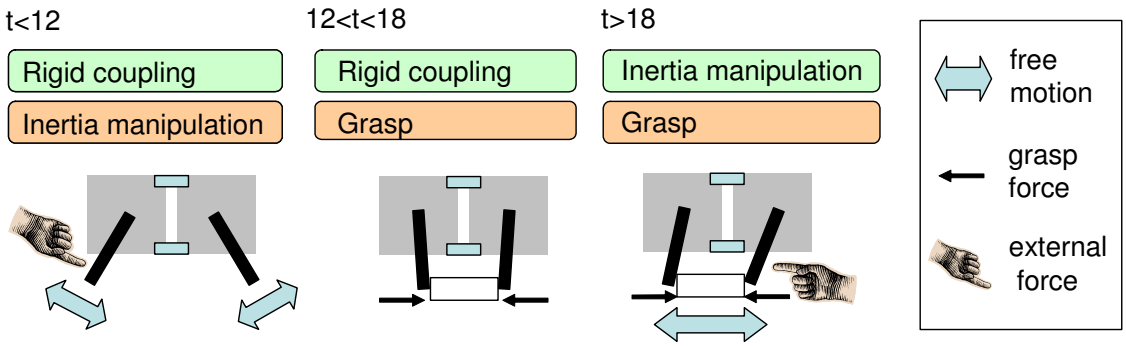


Fig.4-17: Steps on experiment

transformation matrix was

$$\boldsymbol{T} = \begin{bmatrix} \boldsymbol{T}_{IM} \\ \boldsymbol{T}_{grasp} \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}. \tag{4.23}$$

6) $t > 18$ The load moved freely while it was grasped by the manipulators. Fig. 4-19 shows differential force, which is the grasp force, was almost constant after the grasp function was applied. It means grasp motion was sustained. Transformation vector in (4.23) shows that the inertia manipulation function is based on the sum information of both manipulators. Hence, external force on the manipulators was assisted and the virtual inertia on entire motion became smaller.

Table 4.3: Manipulator parameters

Arm length[m]	0.16
Rated power output[W]	50
Rated motor torque[mNm]	159.0
Reduction ratio	1/33
Number of encoder pulse[P/R]	2048
MOI at reducer output shaft[kgm^2]	0.00535

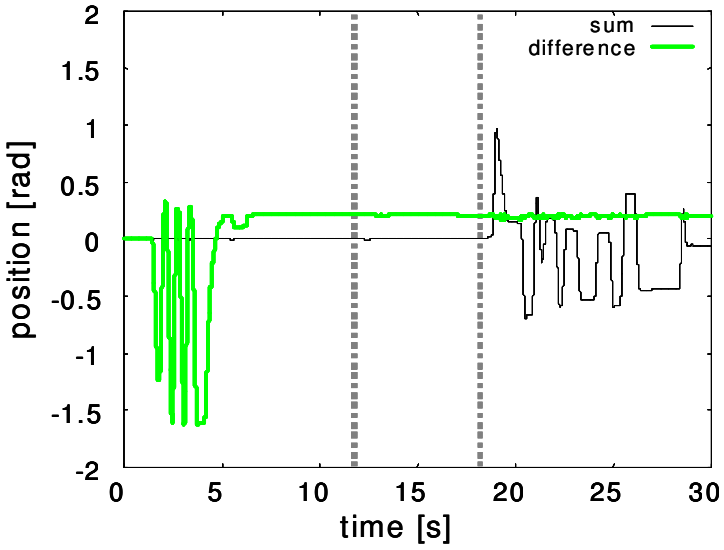


Fig.4-18: Position response when function changeover occurs(function-based information)

Table 4.4: Control parameters

	Sampling period [ms]	0.1
g	Cutoff frequency of DOB [rad/s]	300
K_{RC}	Position gain for rigid coupling function	15
D_{RC}	Velocity gain for rigid coupling function	1.2
k_f	Force gain	1.5

The experimental result mentions that function-based controller design simplify the way to shift tasks. Since each function is independent to each other, tasks could be adjusted without considering the entire system.

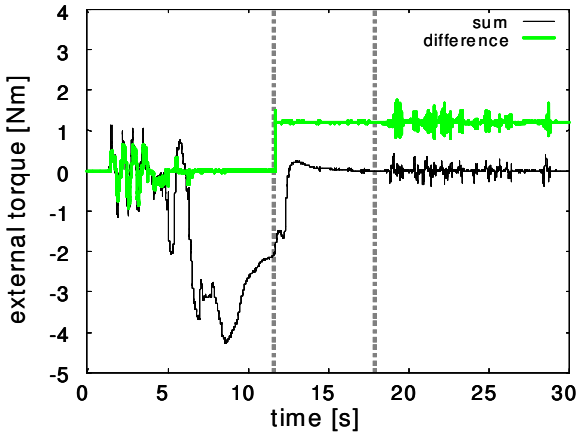


Fig.4-19: Force response when function changeover occurs(function-based information)

4.6 Generalized form in 3-dimensional space

4.6.1 Coordinate transformation based on function

The controller design based on functionality needs coordinate transformation. This subsection describes an extended form of the coordinate transformation.

There exists many kinds of functions for tasks, exception handling, and so on. Various kinds of information such as arm tip position, motor angles and modal information are required for the functions. Multi-layered transformation is therefore introduced. An outline of the transformation is shown in Fig. 4-20.

The coordinate transformation introduced in Section 4.2.2 was to derive function coordinate space from workspace information of each robot. Note that workspace of a one-dimensional mobile robot coincides with space of motor angle. A Jacobian matrix is known for transformation from joint space to workspace. Transformation from real motor coordinate space to virtual motor coordinate space of sum and difference motor is introduced for a twin-drive system. [65].

Several coordinate spaces are transformed through transformation matrices. ${}^{f}\boldsymbol{T}_{r}$, a transformation matrix from real motor coordinate space to function coordinate space, is derived by multiplying the matrices between each space.

At first, function coordinate space is transformed from arm coordinate space (i.e. workspace

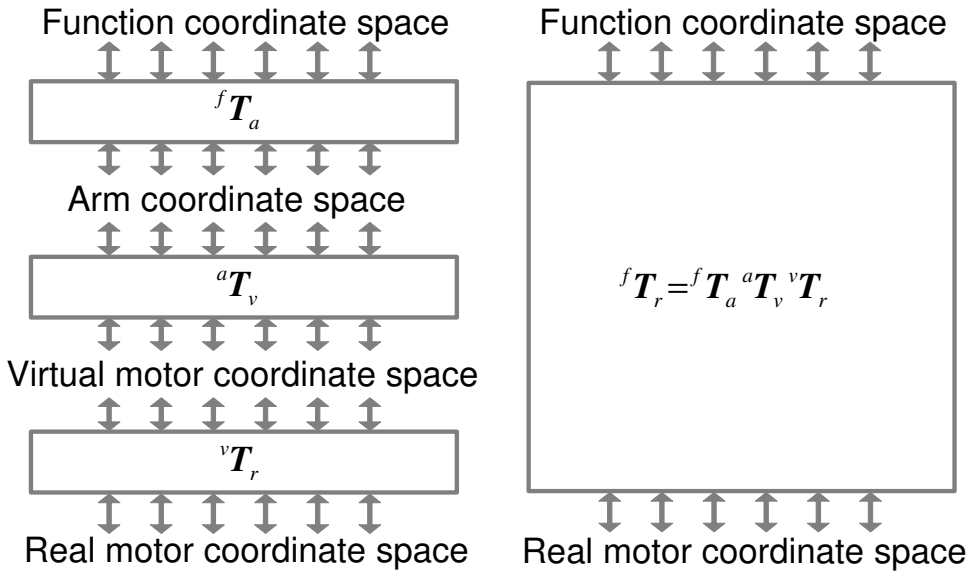


Fig.4-20: Outline of coordinate transformation

of each robot) as follows:

$$\boldsymbol{x}_f = {}^{f}\boldsymbol{T}_a \boldsymbol{x}_a \tag{4.24}$$

$$\dot{\boldsymbol{x}}_f = {}^{f}\boldsymbol{T}_a \dot{\boldsymbol{x}}_a \tag{4.25}$$

$$\ddot{\boldsymbol{x}}_f = {}^{f}\boldsymbol{T}_a \ddot{\boldsymbol{x}}_a \tag{4.26}$$

$$\boldsymbol{f}_f = {}^{f}\boldsymbol{T}_a \boldsymbol{f}_a \tag{4.27}$$

$$\boldsymbol{x}_a = [\boldsymbol{x}_{a1}, \boldsymbol{x}_{a2}, \cdots, \boldsymbol{x}_{am}]^T$$

$$\boldsymbol{f}_a = [\boldsymbol{f}_{a1}, \boldsymbol{f}_{a2}, \cdots, \boldsymbol{f}_{am}]^T .$$

Here, $\boldsymbol{x}_{ai} \in \boldsymbol{R}^3$ and it denotes position of an end effector on the ith robot. $\boldsymbol{f}_{ai} \in \boldsymbol{R}^3$ and it denotes external force on the end effector. The subscript f denotes function coordinate space and the subscript a denotes arm coordinate space. ${}^{f}\boldsymbol{T}_a \in \boldsymbol{R}^{N\times M}$, ${}^{a}\boldsymbol{T}_v \in \boldsymbol{R}^{M\times M}$, ${}^{v}\boldsymbol{T}_r \in \boldsymbol{R}^{M\times M}$, m is total number of robots, M is total DOF of robots, and N is total DOF of functions.

${}^{f}\boldsymbol{T}_a$ corresponds to the transformation matrix in Section 4.2. It is composed of $1, 0$ and -1 to calculate sum and difference information of related arm tip variables.

As shown from (4.24) to (4.27), position, velocity, acceleration and external force are all

transformed by ${}^{f}\boldsymbol{T}_{a}$. Position of arm tip is calculated by direct kinematics based on a real motor response. Force on arm tip is measured by a force sensor or RFOB. Then, position and force information for function-based controller are derived from (4.24) and (4.27), respectively. Velocity and acceleration information on function coordinates are derived from a real motor response by (4.28) and (4.29).

$$\dot{\boldsymbol{x}}_f = {}^{f}\boldsymbol{T}_r \dot{\boldsymbol{x}}_r \tag{4.28}$$

$$\ddot{\boldsymbol{x}}_f = {}^{f}\boldsymbol{T}_r \ddot{\boldsymbol{x}}_r \tag{4.29}$$

$${}^{f}\boldsymbol{T}_r = {}^{f}\boldsymbol{T}_a \, {}^{a}\boldsymbol{T}_v \, {}^{v}\boldsymbol{T}_r \tag{4.30}$$

where, ${}^{a}\boldsymbol{T}_v$ is a transformation matrix similar to a Jacobian matrix. It transforms virtual motor coordinate space to arm coordinate space. ${}^{v}\boldsymbol{T}_r$ is a transformation matrix from real motor coordinate space to virtual motor coordinate space.

${}^{v}\boldsymbol{T}_r$ is a specific transformation matrix only for a twin drive system. It is a unit matrix $\boldsymbol{I}$ for other general systems. In a one-dimensional system, the Jacobian matrix of each robot ${}^{a}\boldsymbol{T}_v$ is also a unit matrix $\boldsymbol{I}$.

${}^{f}\boldsymbol{T}_r$ can be explained as an extended Jacobian matrix. It is extended for a twin-drive system and cooperative work of a multi-robot system. It is therefore called "cooperative Jacobian matrix". ${}^{f}\boldsymbol{T}_a$, which is simply named "transformation matrix" in Section 4.2 is called "function matrix" for distinction.

Control input $\boldsymbol{u}_f$ is derived from controllers on function coordinate space. Here, $\boldsymbol{u}_f$ is in acceleration dimension. Torque input in real motor coordinate is derived from (4.31).

$$\boldsymbol{\tau}_r = \boldsymbol{M}_n \, {}^{f}\boldsymbol{T}_r^{+} \boldsymbol{u}_f \tag{4.31}$$

$${}^{f}\boldsymbol{T}_r^{+} = ({}^{f}\boldsymbol{T}_r^{T} \, {}^{f}\boldsymbol{T}_r)^{-1} \, {}^{f}\boldsymbol{T}_r^{T}$$

Here, $\boldsymbol{M}_n \in \boldsymbol{R}^{M \times M}$ and it is the nominal value of the inertia matrix of robots. The condition for deriving torque input is

$$rank(\boldsymbol{M}_n \, {}^{f}\boldsymbol{T}_r^{+}) = M. \tag{4.32}$$

Therefore, if any of functions are dependent on each other, a new function should be added. On the other hand, if functions are overfull, one of the functions with the lowest priority should be halted. The entire block diagram is shown in Fig. 4-21.

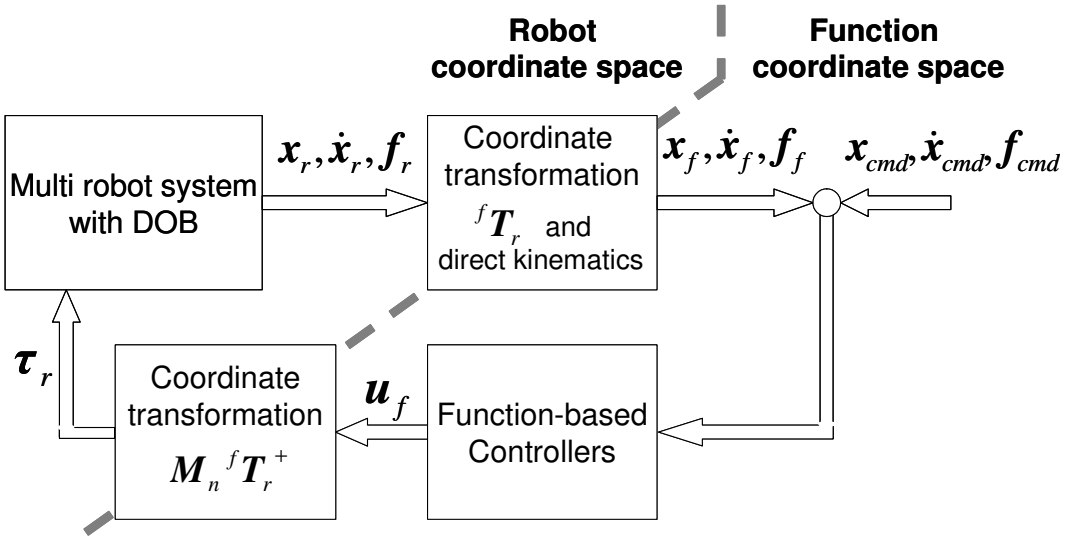


Fig.4-21: Block diagram of function-based control system

4.6.2 Dynamics in function coordinate space

It is to be anticipated from the name of cooperative Jacobian matrix that the coordinate transformation is for kinematics of a large scale system. Virtual dynamics in a function coordinate interferes with each other, contrary to the method in one-dimensional systems. The interference occurs due to the generalization to three-dimensional systems.

Disturbance observer is applied to all of real motors in this method to cancel the interferences. It is known that the plant works as a nominal system when acceleration control is acquired with disturbance observer [66]. Hence inputs from position/force controller based on functions are superposed without any interference.

4.6.3 Procedures of controller design

A design flow of function-based control system is shown in Fig. 4-22. Firstly, the system role is determined by a designer of the control system. Secondly, the designer divides the system role into functions. Thirdly, a priority order of functions is determined. Important functions should be secured even if the number of active functions alters. Then, the transformation matrix ${}^f\boldsymbol{T}_r$ is derived. The number of functions is modified so that rank of ${}^f\boldsymbol{T}_r$ agrees with total DOF of robots M. Otherwise, (4.32) is unsatisfied. Finally, function-based controllers are designed

individually.

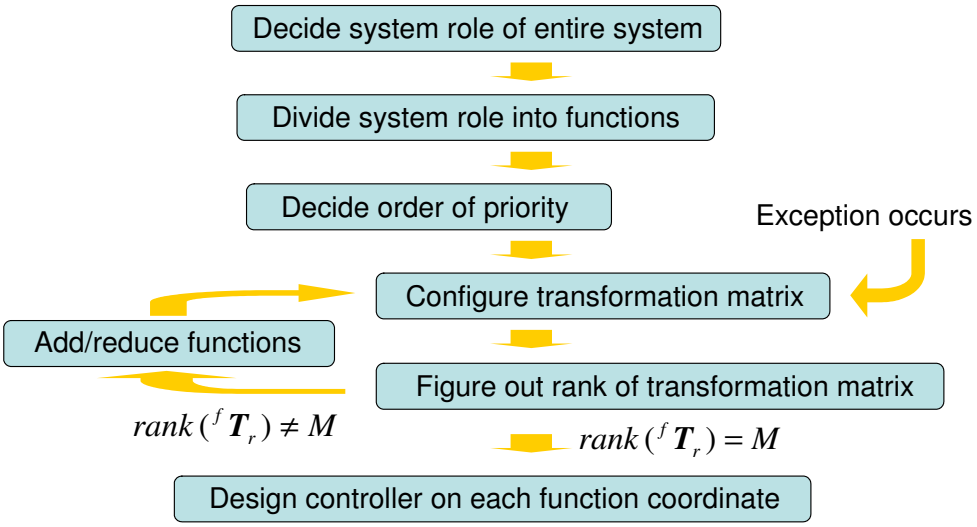


Fig.4-22: Flow of controller design

4.6.4 Reconfiguration for alteration of system role

When the system role alters, combination of functions and its transformation matrix should be modified. At first, new combination of task functions should be given by the designer. Here, a task function is a function to acquire the system role while a performance-limit function is a function to deal with an exception. In the next place, the transformation matrix should be modified along with the functions. Majority of task functions control relative position or relative force between arm tips. In this study, ${}^{f}\boldsymbol{T}_{a}$ denotes the relation between arm tips. In sum, ${}^{f}\boldsymbol{T}_{a}$ should be modified in a similar way in Section 4.2 by modifying $\boldsymbol{T}$ when the system role alters.

4.6.5 Reconfiguration for exception handling

Reconfiguration for exception handling is more difficult compared to that for alteration of the system role. There are three reasons: exceptions occur all of a sudden; the control system should choose the combination of functions autonomously; not only ${}^{f}\boldsymbol{T}_{a}$ but also ${}^{a}\boldsymbol{T}_{v}$ or ${}^{v}\boldsymbol{T}_{r}$ should be modified since performance-limit functions that deal with exceptions are often based on a real motor output or a virtual motor output. A method to modify a transformation matrix

is introduced below.

${}^f\boldsymbol{T}_r$ is described as follows:

$$ {}^f\boldsymbol{T}_r = \left[\begin{array}{cccc} {}^f\boldsymbol{t}_{r1}^T & {}^f\boldsymbol{t}_{r2}^T & \cdots & {}^f\boldsymbol{t}_{rN}^T \end{array} \right]^T . \tag{4.33} $$

${}^f\boldsymbol{t}_{ri} \in \boldsymbol{R}^M$, it extracts the coordinate of the ith function. It is called "function mode" and depends on the characteristics of the function. Function modes for task functions are derived all at once from (4.30).

On the other hand, performance-limit functions, which are activated in a special case also have their function modes. The function mode of the performance-limit function should be derived individually when the function is activated. The function mode of the performance-limit function is derived from various ways since performance-limits may exist in each layer of the multi-layered coordinate transformation. For example, a function mode of a velocity-limit function on the kth real motor is derived as follows:

$$ \begin{aligned} {}^f\boldsymbol{t}_{r,PL}^T &= \left[\begin{array}{cccc} t_1 & t_2 & \cdots & t_M \end{array} \right] \\ & \left\{ \begin{array}{ll} t_i = 1 & i = k \\ t_i = 0 & otherwise. \end{array} \right. \end{aligned} \tag{4.34} $$

Here, ${}^f\boldsymbol{t}_{r,PL}$ denotes a function mode of a performance-limit function.

A position-limit function for avoidance of a singular point is shown as another example of a performance-limit function. A joint angle of the twin drive system corresponds to a response value of a virtual differential motor. Hence a singular point is avoided by setting a position-limit on the virtual motor. A function mode of the position-limit function for the kth virtual motor is derived as follows:

$$ {}^f\boldsymbol{t}_{r,PL} = {}^v\boldsymbol{t}_{rk} \tag{4.35} $$

where, ${}^v\boldsymbol{T}_r = \left[\begin{array}{cccc} {}^v\boldsymbol{t}_{r1}^T & {}^v\boldsymbol{t}_{r2}^T & \cdots & {}^v\boldsymbol{t}_{rN}^T \end{array} \right]^T$.

A function mode of a position-limit function on an arm tip is derived as follows:

$$ {}^f\boldsymbol{t}_{r,PL} = {}^a\boldsymbol{t}_{rk} \tag{4.36} $$

where, ${}^a\boldsymbol{T}_r = \left[\begin{array}{cccc} {}^a\boldsymbol{t}_{r1}^T & {}^a\boldsymbol{t}_{r2}^T & \cdots & {}^a\boldsymbol{t}_{rN}^T \end{array} \right]^T$. It is assumed that the position limit is set for the kth element of $\boldsymbol{x}_a$.

A procedure for exception handling is shown as follows:

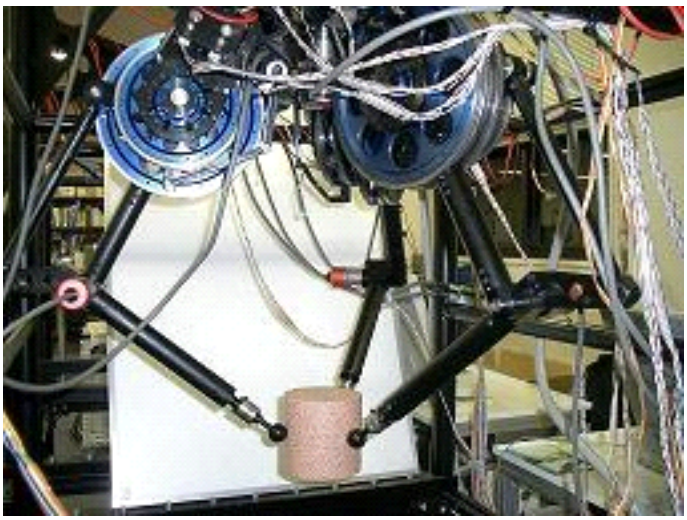


Fig.4-23: Parallel link manipulators

1. Keep observing variables for discriminating exceptions
2. Select a relevant performance-limit function when one of the variables exceeds its limit
3. Derive ${}^{f}\boldsymbol{t}_{r,PL}$, a function mode of the performance-limit function
4. Derive ${}^{f}\boldsymbol{t}_{r,low}$, the function mode of the lowest-priority function
5. If ${}^{f}\boldsymbol{t}_{r,PL} \cdot {}^{f}\boldsymbol{t}_{r,low} \neq 0$, substitute ${}^{f}\boldsymbol{t}_{r,PL}$ to ${}^{f}\boldsymbol{t}_{r,low}$ in ${}^{f}\boldsymbol{T}_{r}$
6. If ${}^{f}\boldsymbol{t}_{r,PL} \cdot {}^{f}\boldsymbol{t}_{r,low} = 0$, select the function with the next-lowest priority, derive its function mode ${}^{f}\boldsymbol{t}_{r,low}$, and go to 5).

4.7 Function-based controller design for cooperative grasping motion

A control system for parallel link manipulators is shown in this section as a typical example of a function-based system. A picture of manipulators is shown in Fig. 4-23. The modeling of the manipulators is shown in [67]. The entire system consists of three parallel link manipulators with 3 DOF. There are 6 motors on each manipulator since the manipulator consists of twin drive systems.

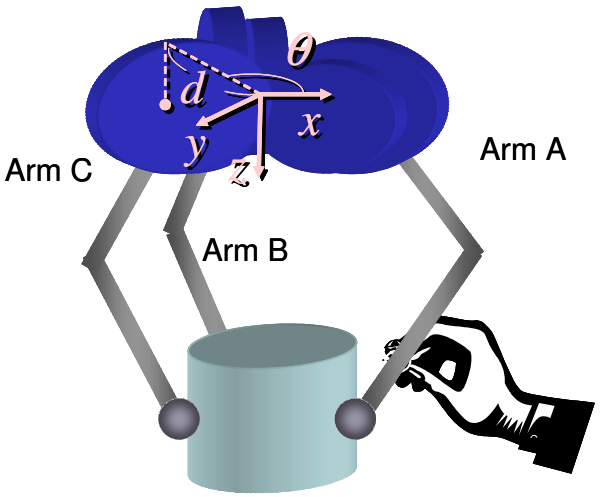


Fig.4-24: Overview of work

An overview of the work with the control system is shown in Fig. 4-24. An operator holds one of the manipulators and handles an object with it.

Three manipulators are fixed with orientation difference of 120 degrees respectively. Absolute position of the arm tip is presented by cylindrical coordinates as shown in (4.37).

$$\boldsymbol{x}_{ai} = [\ d_i \quad \theta_i \quad z_i\ ]^T \tag{4.37}$$

where d denotes distance from the z-axis based on the center of three manipulators, z denotes up-down position, and θ denotes rotation angle in a horizontal plane.

A control system to execute the following operation was developed as a typical example of a human support operation with task variation.

Firstly in Step 1, the arm tips of the three manipulators move in compliance with external force only in the grasping mode, a mode that denotes sum of d_A,d_B and d_C. Step 2 starts after the operator inserts a cylindrical object between the three arm tips. In Step 2, the object is cooperatively grasped by the three arms while its position and attitude is kept constant under external force. In Step 3, the object moves in compliance with external force only in the pitching mode while it is grasped. Its position is kept constant at that time. In Step 4, it moves only in the up-down mode while its attitude is kept constant and it is grasped. Task functions for acquiring the system roles in Step 1 to Step 4 are shown in Table 4.5. The overview of the coordinate transformation is shown in Fig. 4-25.

Table 4.5: Functions for parallel link manipulators

	Step 1	Step 2	Step 3	Step 4
Based on d				
Mode 1(Grasping)	**SC** (1)	**GR** (1)	**GR** (1)	**GR** (1)
Mode 2	RC (2)	RC (2)	RC (2)	RC (2)
Mode 3	RC (3)	RC (3)	RC (3)	RC (3)
Based on θ				
Mode 1(Rolling)	RC (9)	RC (9)	RC (9)	RC (9)
Mode 2	RC (8)	RC (8)	RC (8)	RC (8)
Mode 3	RC (7)	RC (7)	RC (7)	RC (7)
Based on z				
Mode 1(Up-down)	RC (6)	RC (6)	RC (6)	**SC** (6)
Mode 2(Pitching)	RC (5)	RC (5)	**SC** (5)	RC (5)
Mode 3	RC (4)	RC (4)	RC (4)	RC (4)
Based on virtual sum motors				
Mode 1	VC (10)	VC (10)	VC(10)	VC(10)
⋮	⋮	⋮	⋮	⋮
Mode 9	VC (18)	VC (18)	VC(in	VC(18)

Here, RC, SC, VC, and GR denote functions of rigid coupling, spring coupling, velocity control, and grasping, respectively. Numbers in parentheses denote the priority order of the function. The grasping function has higher priority to secure the object. Velocity control functions on sum coordinates keep velocity of virtual sum motors constant to avoid stick-slip phenomenon [65]. Outputs of the functions have relatively small effects on the operation. The velocity control functions therefore have lower priority. The priority order of other task functions is given arbitrarily. Performance-limit functions exist in addition to the task functions. Priority of performance-limit functions are set higher than that of task functions so that they are compulsively activated when exceptions occur.

The function matrix ${}^f\boldsymbol{T}_a$ for such functions is given as follows:

$$
{}^f\boldsymbol{T}_a = \begin{bmatrix} \boldsymbol{T}_d & & & \\ & \boldsymbol{T}_\theta & & \\ & & \boldsymbol{T}_z & \\ & & & \boldsymbol{I}_9 \end{bmatrix} {}^f\boldsymbol{S}_a \tag{4.38}
$$

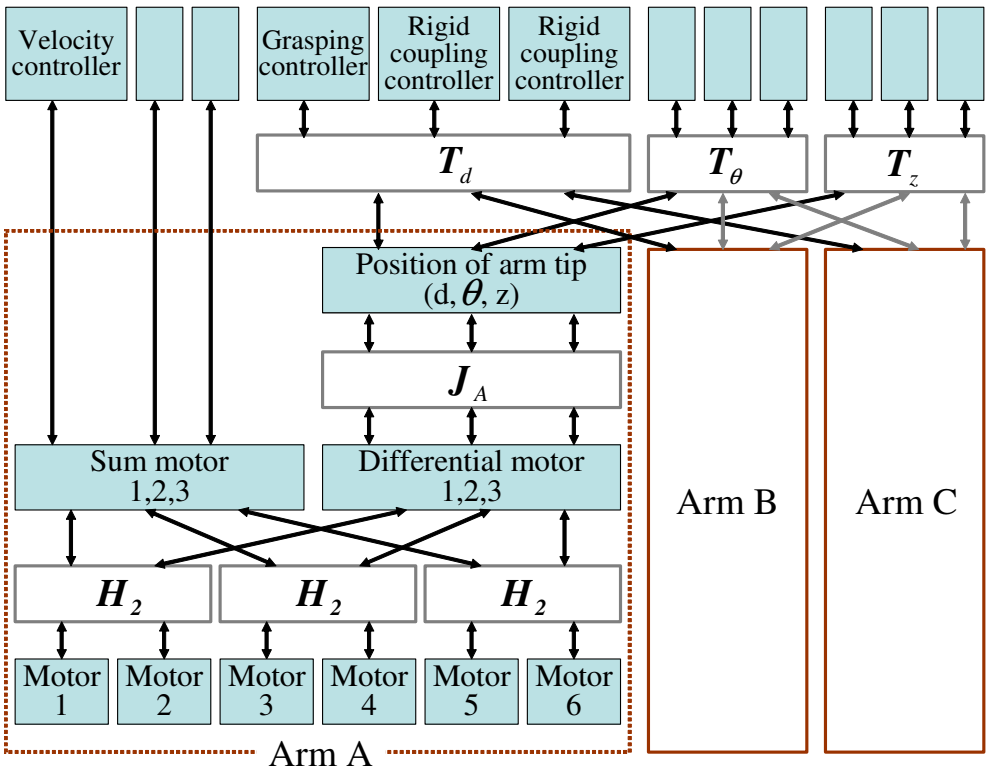


Fig.4-25: Overview of entire coordinate transformation

$$\boldsymbol{T}_d = \begin{bmatrix} 1 & 1 & 1 \\ 1 & -1 & 0 \\ 1 & 0 & -1 \end{bmatrix} \tag{4.39}$$

$$\boldsymbol{T}_\theta = \begin{bmatrix} 1 & 1 & 1 \\ 1 & -1 & 0 \\ 1 & 0 & -1 \end{bmatrix} \tag{4.40}$$

$$\boldsymbol{T}_z = \begin{bmatrix} 1 & 1 & 1 \\ 1 & -1 & 0 \\ 1 & 0 & -1 \end{bmatrix} \tag{4.41}$$

$$
{}^{f}\boldsymbol{S}_{a} = \begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
\end{bmatrix} \tag{4.42}
$$

where, ${}^{f}\boldsymbol{S}_{a}$ is a permutation matrix to change an order of variables from an arm-based order to a function-based order. $\boldsymbol{I}_{n}$, an nth order unit matrix, corresponds to virtual sum motor coordinates. $\boldsymbol{T}_{d}$ denotes a function matrix in d coordinates while $\boldsymbol{T}_{\theta}$ and $\boldsymbol{T}_{z}$ denote that in θ and z coordinates. The first row of $\boldsymbol{T}_{d}$, $\boldsymbol{T}_{\theta}$ and $\boldsymbol{T}_{z}$ are to derive sum of three manipulators' responses. The sum modes are named Mode 1. The second row and the third row are to derive the difference value of the manipulator A and others. The difference values correspond to Mode 2 and Mode 3.

${}^{a}\boldsymbol{T}_{v}$ in this study is shown as follows:

$$
{}^{a}\boldsymbol{T}_{v} = \begin{bmatrix}
{}^{a}\boldsymbol{T}_{vA} & & \\
& {}^{a}\boldsymbol{T}_{vB} & \\
& & {}^{a}\boldsymbol{T}_{vC}
\end{bmatrix} \tag{4.43}
$$

$$
{}^{a}\boldsymbol{T}_{vA} = \begin{bmatrix}
\boldsymbol{I}_{3} & \\
& \boldsymbol{J}_{A}
\end{bmatrix} \tag{4.44}
$$

$$
{}^{a}\boldsymbol{T}_{vB} = \begin{bmatrix}
\boldsymbol{I}_{3} & \\
& \boldsymbol{J}_{B}
\end{bmatrix} \tag{4.45}
$$

$$
{}^{a}\boldsymbol{T}_{vC} = \begin{bmatrix}
\boldsymbol{I}_{3} & \\
& \boldsymbol{J}_{C}
\end{bmatrix}. \tag{4.46}
$$

Here, $\boldsymbol{J}_{A}$, $\boldsymbol{J}_{B}$ and $\boldsymbol{J}_{C}$ denote Jacobian matrices for arm A, B and C, respectively.

${}^{v}\boldsymbol{T}_r$ in this study is shown as follows:

$$
{}^{v}\boldsymbol{T}_r = \begin{bmatrix} {}^{v}\boldsymbol{T}_{rA} & & \\ & {}^{v}\boldsymbol{T}_{rB} & \\ & & {}^{v}\boldsymbol{T}_{rC} \end{bmatrix} \tag{4.47}
$$

$$
{}^{v}\boldsymbol{T}_{rA} = {}^{v}\boldsymbol{T}_{rB} = {}^{v}\boldsymbol{T}_{rC}
$$

$$
= {}^{v}\boldsymbol{S}_r \begin{bmatrix} \boldsymbol{H}_2 & & \\ & \boldsymbol{H}_2 & \\ & & \boldsymbol{H}_2 \end{bmatrix} \tag{4.48}
$$

$$
{}^{v}\boldsymbol{S}_r = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.49}
$$

$$
\boldsymbol{H}_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \tag{4.50}
$$

where, ${}^{v}\boldsymbol{S}_r$ is a permutation matrix to change an order of variables from real motors to virtual motors. $\boldsymbol{H}_2$ is a second-order Hadamard matrix.

4.8 Experiment in three-dimensional system

Experimental results are shown in this section. Table 4.6 shows control gains in the experiment. Figs. 4-26 and 4-27 show responses in d coordinates and in z coordinates, respectively.

Table 4.6: Control parameters

Position gain	K_p	600.0
Velocity gain	K_v	70.0
Force gain	K_f	8.0
Cutoff-frequency of DOB	G_{dis}	30.0
Cutoff-frequency of RFOB	G_f	15.0

When the operator maneuvered the manipulator A in Step 1, all three manipulators moved only in grasping mode and accomplished open-close motion. An object was grasped in Step 2 after the operator inserted it. The object was tilted in the pitching mode in Step 3 when the operator applied force in the z direction. On the other hand, the object went up and down in Step 4 when the operator applied force in the same direction.

External force affected in all directions since the operator did not accurately maneuver. The object, however, moved only in the mode of spring coupling functions as shown from the force responses in Figs. 4-26 and 4-27. The result shows that rigid coupling functions worked as a guide to limit the human manipulation. The direction of free motion can be changed by modifying the combination of functions. The robot moved on the mode with a function that includes force control. Force responses in Fig. 4-26 show that grasping motion was retained then. The result shows that function modes for human manipulation and that for autonomous environmental adaptation were autonomously decoupled then. Furthermore, the mode of human manipulation could be flexibly shifted by modification of functions. Note that grasping an object stands for a simple example of environmental adaptation here. Interaction between each mode was small due to acceleration control based on DOB.

4.9 Summary of chapter

This chapter described function-based controller design for general decentralized control systems. The features of function-based controller design are as follows:

- Design is simple since a function is a minimum component.

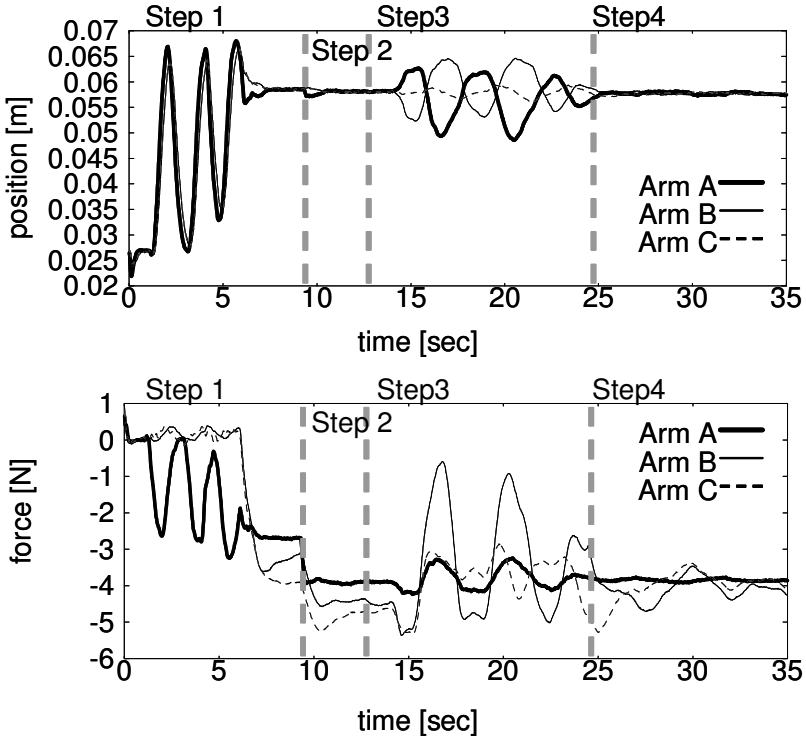


Fig.4-26: Responses in d coordinate

- Design is explicit since the controller directly corresponds to the function.
- Many kinds of exceptions such as faults, performance limits and environmental variations are handled in the unified fashion.
- The task shift could be executed without considering the entire system.

The most basic philosophy implied in this study is "controller design as combination of detachable components." The philosophy is that various kinds of controllers are designed in advance like peripheral equipments for a PC, as shown in Fig. 4-28, and they are activated when there is a request. A great patterns of tasks are realized with such a framework. Furthermore, the design is still simple and explicit. This kind of concept will be a key for future motion control since a robot in human environment has to keep modifying its controller in response to task shifts and exceptions.

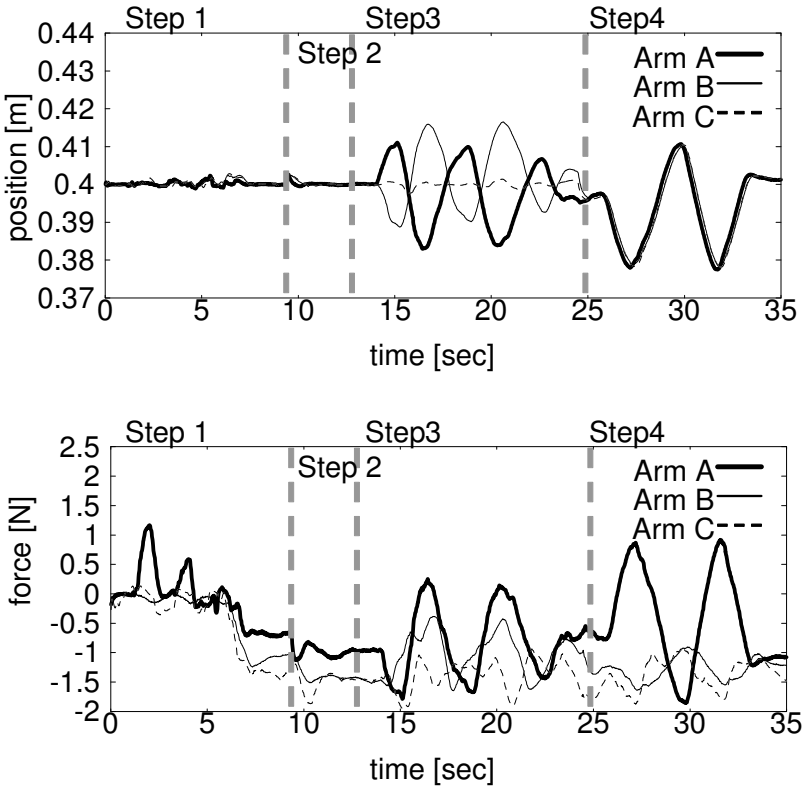


Fig.4-27: Responses in z coordinate

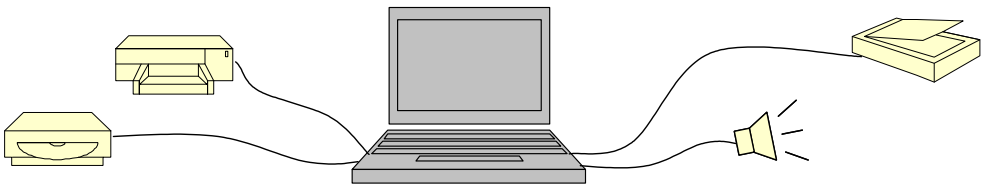


Fig.4-28: Design as detachable component

Chapter 5

Robustness improvement in acceleration control

5.1 Introduction

There are high demands on a control system with rapid response and high robustness. Acceleration control is indispensable for constructing such a system. Hence, many studies on motion control have focused on acceleration control. Disturbance observer (DOB) [28] is useful to develop an acceleration control system with simple parameter modification. Furthermore, acceleration control is also a key for hybrid control since both position and force are represented by a unified unit, acceleration. Joint acceleration controller based on DOB decomposes force control and position control in workspace. [66] As shown in the previous chapters, robots in human environment must have a hybrid control system. Hence, this chapter discusses how to improve control performance of acceleration control.

Accurate and rapid velocity measurement is vital for high-performance acceleration control. Mizuochi, Tsuji and Ohnishi have proposed a new multirate sampling method with a shorter output sampling period to acquire velocity information rapidly [68,69]. The method differs from many other multirate sampling methods [70–72] in the respect that an output sampling period is set shorter than an input sampling period. The study also showed the priority of the sampling period of an output over that of an input in acceleration control. Although the method much improves the control performance, the short sampling period exposes the issue of a quantization error on an optical encoder.

An adverse effect of the quantization error occurs in velocity measurement though there are many studies on velocity measurement or estimation by an optical encoder. Among these studies, there are two commonly used methods: M method and T method. M method, also called fixed-time method, counts the number of pulses generated from the optical encoder in fixed interval

of time and calculates velocity by finite-difference derivative. On the other hand, T method, or fixed-position method, calculates velocity as the interpulse angle divided by the time between sequential pulses. Accuracy deteriorates in a low-speed range with M method, while T method achieves high accuracy. T method is applicable only to the low-speed range however. Ohmae, Matsuda, Kamiyama and Tachikawa [73] proposed M/T method which works in all speed ranges and has a high accuracy in the low-speed range. The method has been applied in many studies since it is effective for practical use. This method is extended to a system termed constant sample-time digital tachometer (CSDT) [74]. It is more easily incorporated into a controller operating with a constant sampling period.

Velocity estimation method with Kalman filter improves the velocity standard deviations [75]. Instantaneous speed observer [76], a discrete-time observer to grasp the velocity between the encoder pulses, is an effective tool for accurate velocity estimation. These methods, however, require plant models.

The velocity measurement method in this study should satisfy the following terms.

- high accuracy
- wide speed range
- rapid response
- measurement without any models

Above all, this study lays weight on "high accuracy" because acceleration control badly requires accurate velocity measurement. Although M/T method almost satisfies the terms, it has high accuracy only in a low speed range. It was believed for long time that no one estimator algorithm is best for a system with a large dynamic range of speeds, large transients, and an imperfect encoder [77]. On the contrary, this paper proposes a velocity measurement method with high accuracy in all speed ranges.

Contents of this chapter are shown below. Section 5.2 is a description of an experimental setup in this study. In Section 5.3, the mechanism of acceleration control is described to show why acceleration control badly requires accurate velocity measurement. The multirate sampling method for the acceleration control system is described in Section 5.4. Velocity measurement is discussed in the next place since a short output sampling period in the multirate sampling

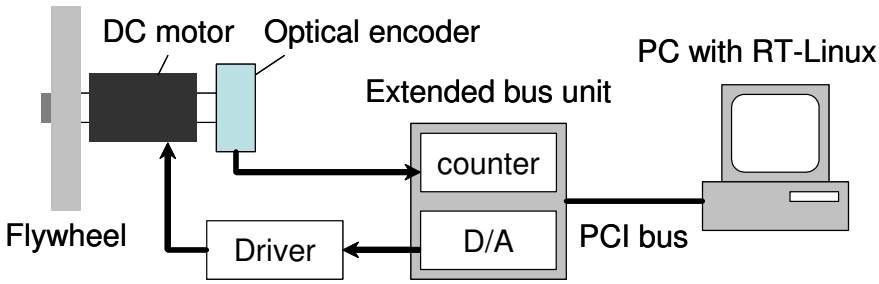


Fig.5-1: Overview of experimental setup

method raises the problem of quantization error on velocity information. Section 5.5 shows conventional methods of velocity measurement. Their resolution and measurement time are also introduced. S method, a novel method for velocity measurement is proposed in Section 5.6. Section 5.7 describes some technical issues of the method. Section 5.8 introduces an indicator to optimize a measurement time. The validity of S method is verified by experimental results in Section 5.9. Finally, this chapter is concluded in Section 5.10.

5.2 Experimental Setup

Fig. 5-1 shows an overview of an experimental system in this study. It is a 1 DOF flywheel driven by a DC motor. An optical encoder generates pulses in proportion to arm displacement. A counter board counts pulses from the optical encoder and PC reads a pulse number from the board. Here, pulse number denotes the number of pulse signals on every sampling period T_s. T_s is controlled to be constant with the real-time architecture of RT-Linux. We compared performance of the proposed multirate method with other methods on this system. The parameters on the experimental setup are shown in Table 5.1.

5.3 Acceleration control

5.3.1 Target of acceleration control

Output of a robot control system is position and/or force. A simple case is continuous path tracking, however, force control is indispensable to work in unknown environment. A simplified

Table 5.1: Experimental parameters

Sampling period	[ms]	1.0
Flywheel MOI	[Kgm^2]	0.003
Type of motor		Maxon RE40
Stall torque	[mNm]	2500
Torque constant	[mNm/A]	60.3
Type of optical encoder		Maxon HEDS 5540
Pulse per revolution	[pulse]	500

index which covers various motion is preferable, though there are various candidates of motion representation. One of such indices is stiffness. Suppose that x is a position of a robot and f is external force on that. From the kinematic and the dynamic equation, the following holds

$$f = g(\ddot{x}, \dot{x}, x). \tag{5.1}$$

The stiffness κ is defined in the partial differentiation

$$\kappa = \frac{\partial f}{\partial x}. \tag{5.2}$$

The ideal position control inhibits any deviation of position against any deviation of force. That means κ will be infinite in such a case. Naturally an integrator in the forward loop compensates the steady error and δx will be zero at infinite time. However, such function does not reflect in (5.2). On the other hand, the ideal force control inhibits any force deviation against any position deviation. Therefore, κ is zero in the ideal force control. In the compliance control, there must be a relation between position and force. For instance, a virtual compliance control will have a mechanical impedance computed in the controller according to the specified dynamics. Table 5.2 shows that κ is a good parameter as an index which represents a target of motion.

From the viewpoint of environmental adaptability, κ is also an effective tool for a hybrid control system based on functionality. However, it is quite difficult to realize an ideal stiffness κ with the hybrid control system since functional modes may interfere to each other in 3-dimensional space. Komada et al. have shown that disturbance observer can realize the nominal system for many types of plants in a hybrid control system [78]. The results in Chapter 4 also

Table 5.2: Stiffness as a motion index

Target of motion	Stiffness κ
position	∞
compliant	finite
force	0

shows that acceleration control based on disturbance observer cancels the interference. In sum, acceleration control is indispensable to acquire a hybrid control system with ideal stiffness κ.

5.3.2 Acceleration control system with disturbance observer

This subsection describes a mechanism of acceleration control. This study applies disturbance observer as basic technique for acceleration control. Fig. 5-2 is the block diagram of disturbance observer. Here, τ_l is a mechanical load, $\hat{\tau}_{dis}$ is estimated disturbance torque, G_{dis} is a cutoff frequency of disturbance observer, G_v is a cutoff frequency of the low-pass filter (LPF) for measured velocity, I_a is input current, K_t is a torque constant, θ is position response of the controlled object, ω is velocity response, J is inertia, s denotes a Laplace operator, a bar over a variable denotes a calculated value, a subscript n denotes a nominal value, a superscript ref denotes a reference value, and a superscript cmp denotes a compensation value.

The total disturbance torque τ_{dis} contains a mechanical load τ_l, varied self-inertia torque $\Delta J\ddot{\theta}$, and torque ripple from motor $\Delta K_t I_a^{ref}$. The disturbance torque τ_{dis} is represented as follows:

$$\tau_{dis} = \tau_l + \Delta J\ddot{\theta} - \Delta K_t I_a^{ref}. \tag{5.3}$$

This disturbance torque is figured out from input and output values as shown in (5.4).

$$\tau_{dis} = K_{tn} I_a^{ref} - J_n \omega s \tag{5.4}$$

where, the Laplace operator s denotes a derivative calculation. The first term $K_{tn} I_a^{ref}$ in (5.4) is based on input information, and the second term $J_n \omega s$ is based on output information. Disturbance torque is estimated through the LPF as shown in (5.5) in order to reduce noise.

$$\hat{\tau}_{dis} = \frac{G_{dis}}{s + G_{dis}} \left(K_{tn} I_a^{ref} - \frac{G_v}{s + G_v} J_n \omega s \right) \tag{5.5}$$

Disturbance observer estimates disturbance on the control system and compensates it. This disturbance estimation is based on the acceleration ωs derived from the output of the optical encoder. In other words, the control system has acceleration feedback in essence. Consequently the control system with disturbance observer is an acceleration control system in essence.

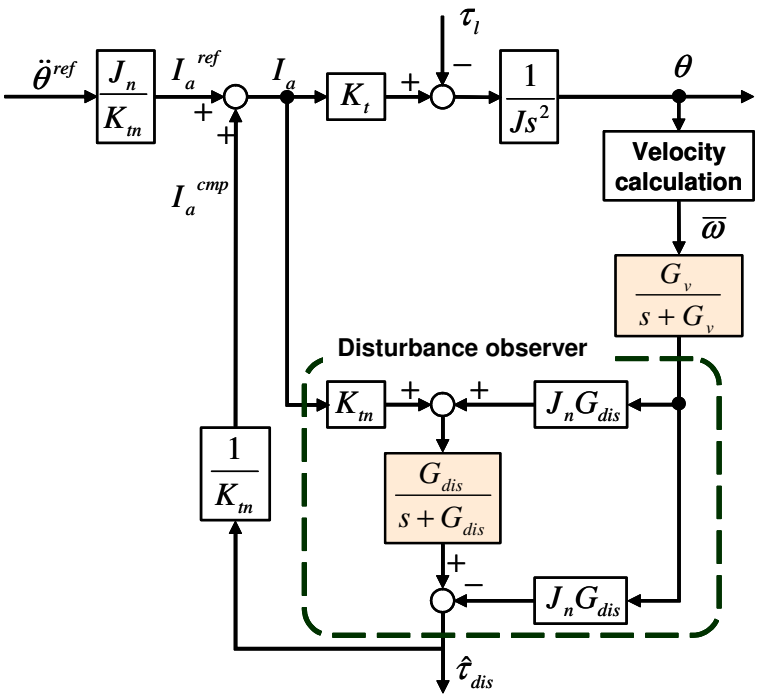


Fig.5-2: Disturbance Observer

5.3.3 Quantization error in acceleration control

This subsection describes the influence of a quantization error of optical encoders in order to show the importance of velocity measurement accuracy in acceleration control. Fig. 5-3 shows the equivalent transformation form of disturbance observer. Equation (5.6) shows $\hat{\tau}_{dis}$ estimated in practice.

$$\hat{\tau}_{dis} = \frac{G_{dis}}{s+G_{dis}}\left(K_t I_a^{ref} - \frac{G_v}{s+G_v} J_n \bar{\omega} s\right) \tag{5.6}$$

$\bar{\omega}$ includes a certain amount of noise due to the quantization error on the optical encoder. Its accuracy and delay depend on the calculation method. Finite-difference derivative amplifies the

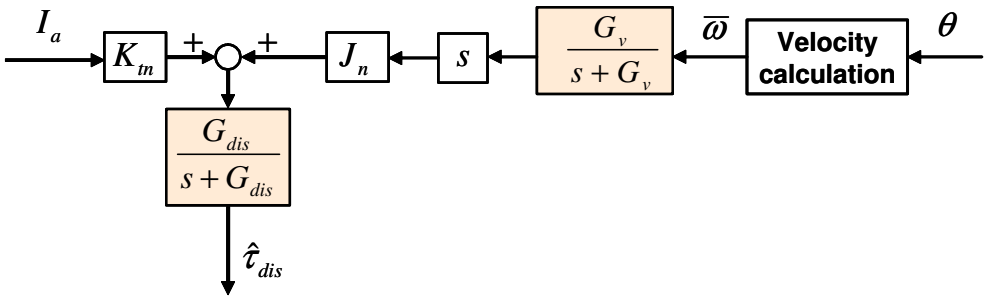


Fig.5-3: Equivalent transformation form of disturbance observer

noise. Fig. 5-4 shows velocity and acceleration values in simulation. Velocity values are derived by finite-difference derivative of position values. At the same time, acceleration values are derived by finite-difference derivative of the velocity values. Parameters in this simulation such as encoder resolution and the sampling period are equal to that of the experimental system. The result shows how large the noise on measured acceleration is, compared with the true value. Two LPFs are introduced to reduce this noise while they cause a delay on disturbance estimation. The cutoff frequencies of the LPFs should be high since this delay may deteriorate the performance of the control system. Accurate velocity measurement is indispensable to heighten the cutoff frequency.

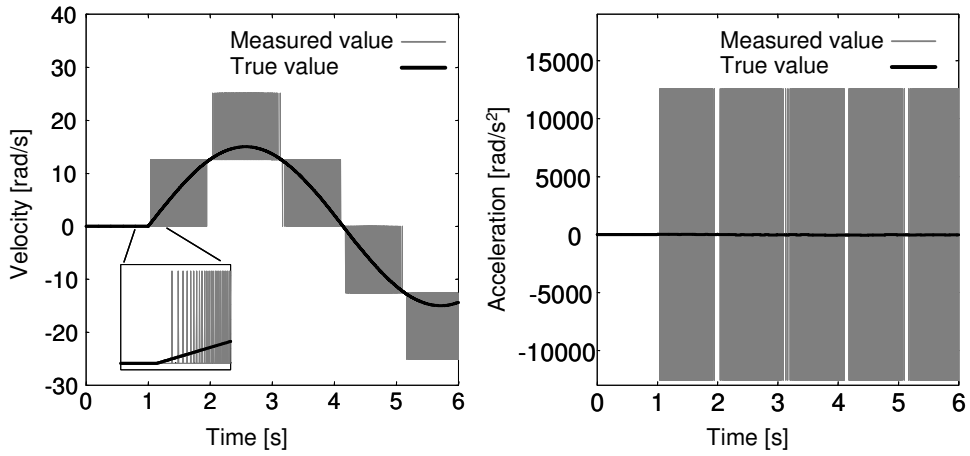


Fig.5-4: Velocity values and acceleration values derived by finite-difference derivative

5.4 Multirate sampling method

5.4.1 Multirate sampling method for acceleration control

This section describes the multirate sampling method proposed in [68] for the acceleration control system. Here, output and input periods are defined as follows:

- output sampling period: sampling period for acquisition of sensor information; and
- input sampling period: sampling period for renewal of a current input reference.

As mentioned in the previous section, it is important in acceleration control systems to acquire output information in a shorter sampling period than renewal of an actuation input. The authors, therefore, propose a new multirate sampling method shown in Fig. 5-5, in which output information is acquired several times during one input sampling period. The sampling period of an input T_u and that of the controller T_r are selected to satisfy the following equations.

$$T_u = nT_y \tag{5.7}$$

$$T_r = T_y \tag{5.8}$$

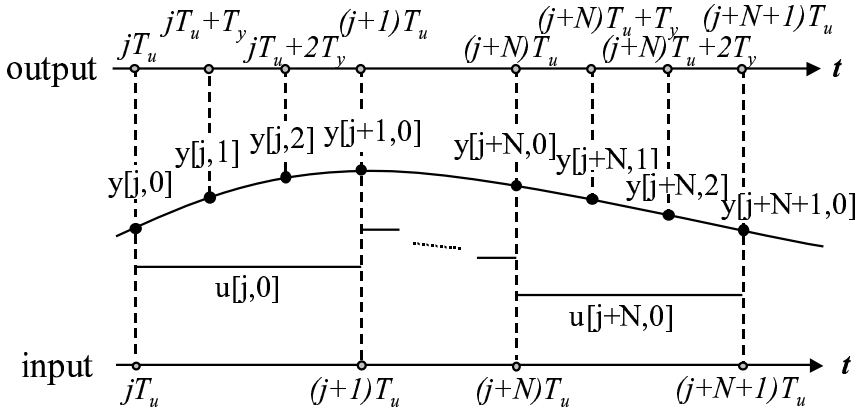
where, T_y is the sampling period of an output and n is the integer number. Control calculation is therefore performed at every output sampling period.

The limitations on sampling periods also support the adequacy of the proposal. The limitations on the input sampling period are generally more severe than those on the output. An output sampling period can be selected in proportion to the clock time of DSP or FPGA. A control sampling period is limited only by calculation time. Those sampling periods become able to set shorter and shorter, considering the development of devices such as FPGA. On the other hand, a frequency of a current input is limited strictly by performance of an amplifier or a frequency of PWM. Therefore, the output sampling period can be set shorter than the input in many cases.

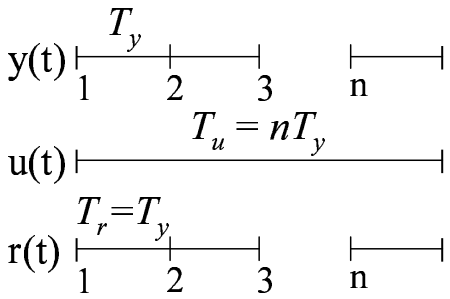
Consider a continuous-time plant represented as follows:

$$\dot{\boldsymbol{x}}(t) = \mathbf{A}\boldsymbol{x}(t) + \boldsymbol{b}u(t) \tag{5.9}$$

$$y(t) = \boldsymbol{c}\boldsymbol{x}(t). \tag{5.10}$$



(a)



(b)

Fig.5-5: Multirate sampling method for acceleration control: (a) Multirate sampling (b) Sampling periods

Assuming that the sampling periods of output and input are T and the input $u(\tau)$ remains constant from t_0 to $t_0 + T$, the discrete-time plant is represented as follows:

$$\boldsymbol{x}[i+1] = \mathbf{A}_d \boldsymbol{x}[i] + \boldsymbol{b}_d u[i] \tag{5.11}$$

$$\boldsymbol{y}[i] = \boldsymbol{c}_d \boldsymbol{x}[i] \tag{5.12}$$

where, $\boldsymbol{x}[i] = \boldsymbol{x}(iT)$. Matrix $\mathbf{A}_d$, vectors $\boldsymbol{b}_d$ and $\boldsymbol{c}_d$ are given by

$$\mathbf{A}_d = e^{\mathbf{A}T}, \quad \boldsymbol{b}_d = \int_0^T e^{\mathbf{A}\tau} d\tau \boldsymbol{b}, \quad \boldsymbol{c}_d = \boldsymbol{c}.$$

When a feedback control law is

$$u(t) = f(\boldsymbol{x}(t)), \tag{5.13}$$

then, it is rewritten into the equation below in discrete-time.

$$u[i] = f(\boldsymbol{x}[i]) \tag{5.14}$$

In the proposed multirate method, since the actuation input is updated only when $t = iT_u$ (i :integer number), the feedback control law (5.13) in the multirate system is given by the following equation.

$$u[i,k] = u[i,0] = f(\boldsymbol{x}[i,0]) \tag{5.15}$$

This equation shows that the actuation input remains constant from $t = iT_u$ to $t = (i+1)T_u$. In the proposed multirate method, the state-space equations (5.11) and (5.12) can therefore be rewritten into the equations below, considering the relation of two sampling periods, T_y and T_u.

$$\boldsymbol{x}[i,k+1] = \mathbf{A}_m\boldsymbol{x}[i,k] + \boldsymbol{b}_m u[i,0] : k \neq n-1 \tag{5.16}$$

$$\boldsymbol{x}[i+1,0] = \mathbf{A}'_m\boldsymbol{x}[i,n-1] + \boldsymbol{b}'_m u[i,0] : k = n-1 \tag{5.17}$$

$$\boldsymbol{y}[i,k] = \boldsymbol{c}_m\boldsymbol{x}[i,k] \tag{5.18}$$

where,

$$\begin{aligned} \boldsymbol{x}[i,k] &= \boldsymbol{x}((i+\frac{k}{n})T_u) = \boldsymbol{x}(iT_u + kT_y)\ (k = 0, \cdots, n-1) \\ \mathbf{A}_m &= \mathbf{A}'_m = e^{\mathbf{A}T_y},\ \boldsymbol{b}_m = \boldsymbol{b}'_m = \int_0^{T_y} e^{\mathbf{A}\tau}d\tau\boldsymbol{b}, \quad \boldsymbol{c}_m = \boldsymbol{c}. \end{aligned}$$

5.4.2 Disturbance observer in multirate system

Application of disturbance observer to the multirate system is discussed in this subsection. In the proposed multirate system, there are two values of an input, desired input value $I_m[i,k]$ and real input reference value $I_m^{real}[i,k]$. The former is calculated at an output sampling rate and the latter is a real input reference value to the robot, which is renewed at an input sampling rate. From (5.15), the following relation is obtained.

$$I_m^{real}[i,k] = I_m[i,0] \tag{5.19}$$

Application of conventional disturbance observer

Conventional disturbance observer is expanded for the multirate system in Fig. 5-6(a). Disturbance torque defined in the conventional disturbance observer is represented by the following equation in the multirate system.

$$\tau_{dis}[i,k] = \tau_l[i,k] + \Delta J\ddot{\theta}[i,k] - \Delta K_t I_m^{real}[i,k] \tag{5.20}$$

The estimated disturbance torque is acquired from the real input reference value $I_a^{real}[i,k]$ and velocity with the equation below.

$$\hat{\tau}_{dis} = \frac{G_{dis}}{s+G_{dis}}(K_{tn} I_a^{real}[i,k] - J_n s\dot{\theta}[i,k]) \tag{5.21}$$

Disturbance observer for multirate system

Equation (5.19) shows that there is a deviation between the desired input value and the real input reference value when $t \neq iT_u$. Although the deviation is not considered in the conventional disturbance observer, it may exert an influence on the system. With a focus on that, the authors define the total disturbance torque of the multirate sampling system as the equation below to include the influence of the deviation of the input values.

$$\tau_{mdis}[i,k] = \tau_l[i,k] + \Delta J\ddot{\theta}[i,k] - \Delta K_t I_m[i,k] + (K_{tn} + \Delta K_t)\Delta I_m[i,k] \tag{5.22}$$

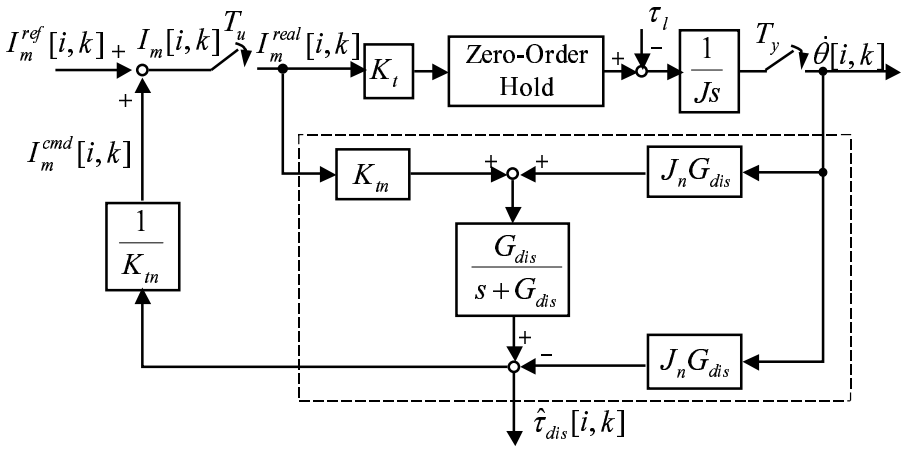
where,

$$\Delta I_m[i,k] = I_m[i,k] - I_m^{real}[i,k].$$

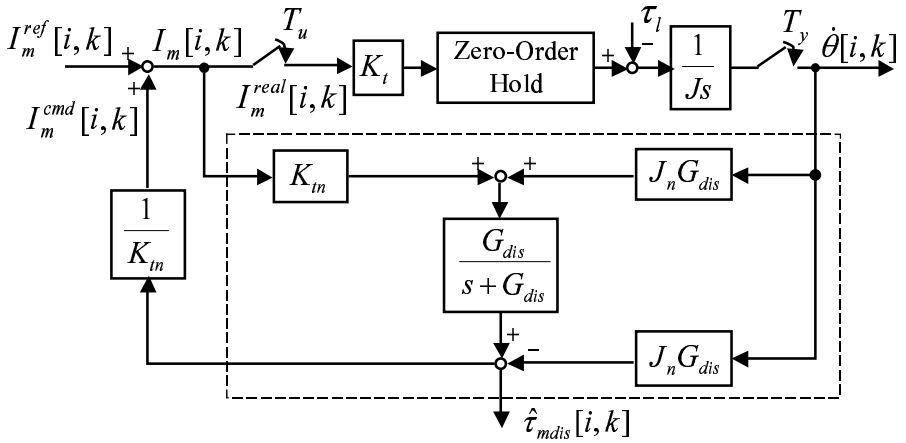
In order to estimate and compensate τ_{mdis}, disturbance observer in the multirate system is proposed as shown in Fig. 5-6(b). The total disturbance torque of the multirate sampling system is estimated by the following equation.

$$\hat{\tau}_{mdis} = \frac{G_{dis}}{s+G_{dis}}(K_{tn} I_a[i,k] - J_n s\dot{\theta}[i,k]) \tag{5.23}$$

Note that the desired input value $I_a[i,k]$ is utilized instead of $I_a^{real}[i,k]$ in the proposed observer.



(a)



(b)

Fig.5-6: Disturbance observer in multirate system: (a) Conventional disturbance observer in multirate system (b) Proposed disturbance observer for multirate system

5.4.3 Application of the proposed multirate method

Overall structure of the proposed multirate method is shown in Fig. 5-7. The following advantages are expected for the method:

- Cut-off frequency can be set higher; and

- Information of disturbance is detected in an early timing.

The first advantage is derived from its higher sampling frequency and larger number of sampled data for observer calculation. The limitation on the cut-off frequency relates to the sampling period and noise. The short sampling period reduces noise by repeated calculation. Influence of digitization is decreased by shortening the sampling period of the observer. The cut-off frequency can therefore be set higher with a shorter sampling period, as described in [**?**]. The limitation of the cut-off frequency becomes high with the multirate sampling method since both noise and influence of digitization are reduced. The second advantage is explained with reference to Fig. **??**. In the single-rate system, the influence of disturbance τ_{dis1} exerted at $t = t_1$ and τ_{dis2} at $t = t_2$ are both recognized at $t = t_0 + T_u$. On the other hand, in the multirate system, influence of τ_{dis1} and τ_{dis2} are recognized at $t_0 + T_y$ and $t_0 + T_u$, respectively. Detection of disturbance influence is generally delayed half an output sampling period in average. When the sampling periods in multirate control satisfy $T_u = nT_y = T_s$ and those in single-rate control satisfy $T_u = T_y = T_s$, the average length of delay is $\frac{T_s}{2n}$ in multirate and $\frac{T_s}{2}$ in single-rate. It shows that the average length of delay becomes n times shorter with the multirate sampling method. In other words, the detection becomes early for $\frac{n-1}{2n}T_s$ in average. As a result, quick response against disturbance is improved and bandwidth of robust acceleration control is increased. On the other hand, the absence of updating of the compensation input that occurs in the proposed multirate method may deteriorate the performance. The proposed disturbance observer enables the system to estimate the disturbance including the influence of the absence of updating. By compensating the value, high performance seems to be obtained. The performance is expected to become close to that achieved with a short sampling period for both the output and input.

5.5 Related research on velocity measurement

Although the multirate method introduced in the previous section much improves the control performance, the short sampling period exposes another issue of a quantization error on an optical encoder as shown in Section5.3. The quantization error produces large noise on velocity information. Furthermore, performance of acceleration control is limited since the error is amplified on the acceleration dimension due to the derivative calculation. Velocity measurement needs improvement to overcome this problem. This section describes related researches on

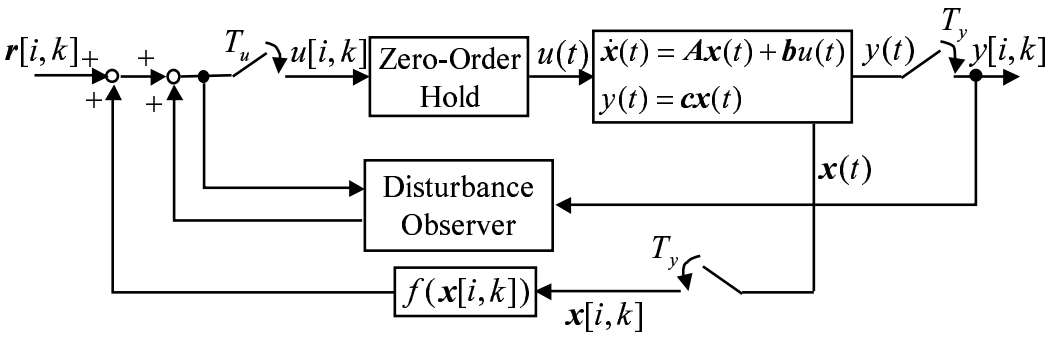


Fig.5-7: Multirate control system

velocity measurement.

5.5.1 M method

M method is the most widely-used method to measure velocity from encoder pulses. The principle is shown in Fig. 5-8. m_e is the number of pulses generated during a fixed sampling period T_s. m_e is utilized for velocity calculation by finite-difference derivative.

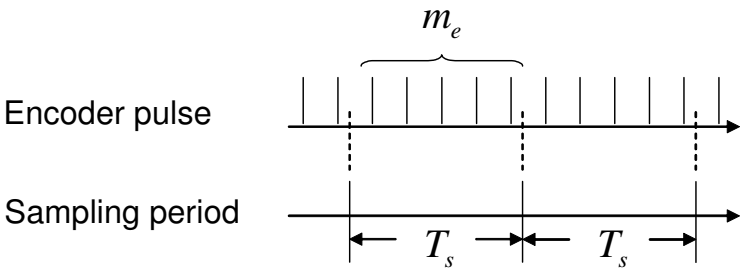


Fig.5-8: Principle of M method

Measured velocity $\bar{\omega}$ is figured out by (5.24). Following equations show velocity resolution and measurement time.

$$\bar{\omega} = \frac{2\pi m_e}{PT_s} \tag{5.24}$$

$$Q_V = \frac{2\pi}{PT_s} \tag{5.25}$$

$$\frac{Q_V}{V} = \frac{2\pi}{PT_sV} \tag{5.26}$$

$$T_m = T_s \tag{5.27}$$

where P denotes an encoder pulse number per rotation. Q_V stands for absolute velocity resolution and $\frac{Q_V}{V}$ stands for relative velocity resolution. V denotes actual angular velocity. A measurement time T_m is equal to the sampling period T_s. Here, a sampling period stands for interval time to count a pulse number periodically. On the other hand, measurement time stands for interval time to calculate velocity based on the pulse number counted in single or multiple sampling periods.

As shown in (5.25) and (5.26), velocity resolution becomes larger as the sampling period becomes shorter. However, acceleration control requires both a short sampling period and accurate velocity measurement.

The easiest way to improve accuracy is to average the velocity values. The average of n sampling periods is acquired by (5.28). The absolute and relative resolutions are shown in Eqs. (5.29) and (5.30). Measurement time is prolonged as shown in (5.31).

$$\bar{\omega}(i) = \frac{2\pi \sum_{j=0}^{n} m_e(i-j)}{nPT_s} \tag{5.28}$$

$$Q_V = \frac{2\pi}{nPT_s} \tag{5.29}$$

$$\frac{Q_V}{V} = \frac{2\pi}{nPT_sV} \tag{5.30}$$

$$T_m = nT_s \tag{5.31}$$

These equations show that averaging improves the accuracy to the nth part of the resolution while measurement time becomes n times longer.

5.5.2 T method

T method measures velocity by dividing the interpulse angle by the pulse interval time as shown in Fig. 5-9. In the figure, encoder pulses come successively with T_e intervals. Assuming that interpulse angle of the optical encoder is completely accurate, accuracy of this method depends only on the measurement of interval time T_e. T_e is measured by counting m_s, the number of sampling periods during the pulse interval. Hence, T_e is substituted by $T_m = m_sT_s$.

Although T_m is an approximate value of T_e, it contains an error less than T_s since m_s is an integer.

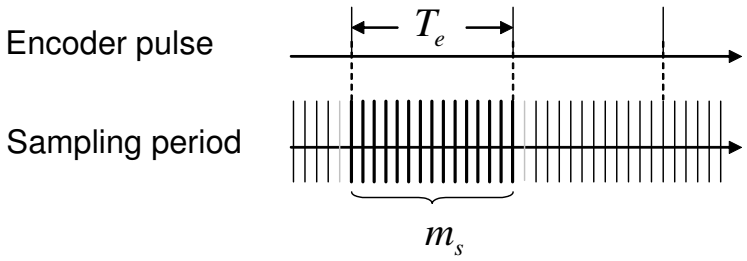


Fig.5-9: Principle of T method

Velocity is calculated by (5.32). Equations (5.33), (5.34) and (5.35) show the performance.

$$\bar{\omega} = \frac{2\pi}{m_s P T_s} \tag{5.32}$$

$$Q_V = \frac{2\pi}{m_s(m_s-1)PT_s} \tag{5.33}$$

$$\frac{Q_V}{V} = \frac{2\pi}{m_s(m_s-1)PT_sV} \tag{5.34}$$

$$T_m = m_s T_s \tag{5.35}$$

It is shown from the equations that T method reduces the maximum error inversely proportional to $m_s(m_s-1)$ while measurement time becomes m_s times longer. It is obvious that accuracy improves much more than averaging. However, there are several problems of this method. The first problem is that the pulse interval is not measurable if it is shorter than the sampling period T_s. In other words, this method is only applicable to a low-speed range. The other problem on this method is that the measurement time T_m is fluctuant and it depends on the velocity. Therefore the measurement delay may become large.

5.5.3 M/T method

M/T method is an effective measurement method that combines M method and T method. It improves the measurement accuracy in a low-speed range and furthermore, it works in all speed ranges. Its performance is in proportion to T method in the low-speed range and to M method with averaging in the high-speed range.

5.6 Proposed velocity measurement method

5.6.1 Principle of proposed velocity measurement method

Principle and procedure of the proposed velocity measurement method are shown in this subsection. The object of the method is to acquire high accuracy in all speed ranges. In order to acquire high accuracy, this subsection discusses why T method achieves high accuracy. Then the measurement method that achieves the accuracy of T method in all speed ranges is proposed.

Equation (5.32) shows that T method calculates velocity by dividing interpulse angle $\frac{2\pi}{P}$ by interval time $m_s T_s$. From other points of view, it averages the velocity synchronized with pulses.

Velocity measurement is more accurate in T method since velocity calculation is synchronized with the timing of pulse generation. Fig. 5-10(a) shows an example when averaging calculation is not synchronized. Average velocity is irregular since sum of pulse numbers in n samples fluctuates depending on the calculation timing. Fig. 5-10(b) shows the other example when averaging calculation is synchronized with pulses. Average velocity is smooth in this case since sum of pulse numbers does not fluctuate. Consequently, the synchronization in T method reduces

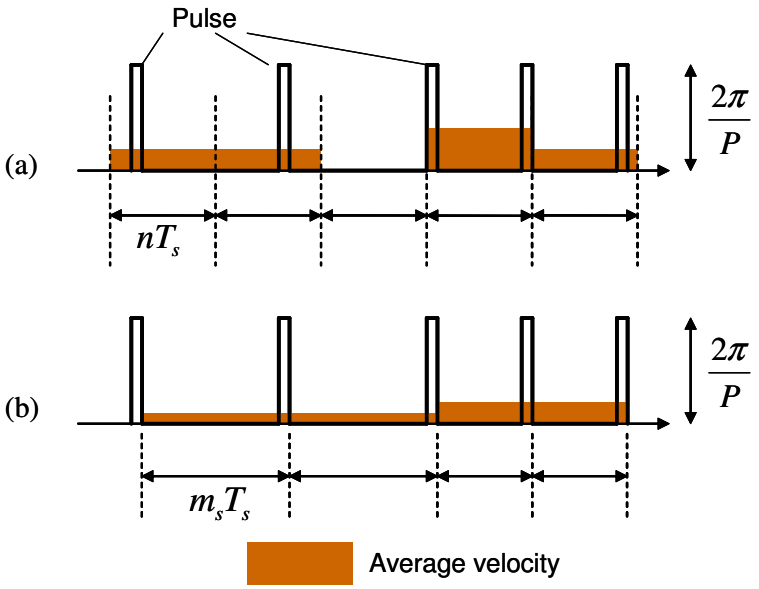


Fig.5-10: Pulse pattern in low-speed range

the error in calculated velocity. T method is limited, however, in a low-speed range because it

is impossible to measure the interval time if pulses occur in every sampling period.

Fig. 5-11 shows pulse numbers on respective sampling periods when velocity varies from low speed to high speed. In fact, it is impossible to measure the interval time between sequential pulses. However, the patterns of the waveform in a high-speed range are quite similar to that in a low-speed range. We can presume the patterns as occasional pulses with a certain amount of offset. Hence the alteration of pulse numbers shown in Fig. 5-12 should be detected so as to synchronize the velocity calculation with it. We call the alteration "pulse alteration".

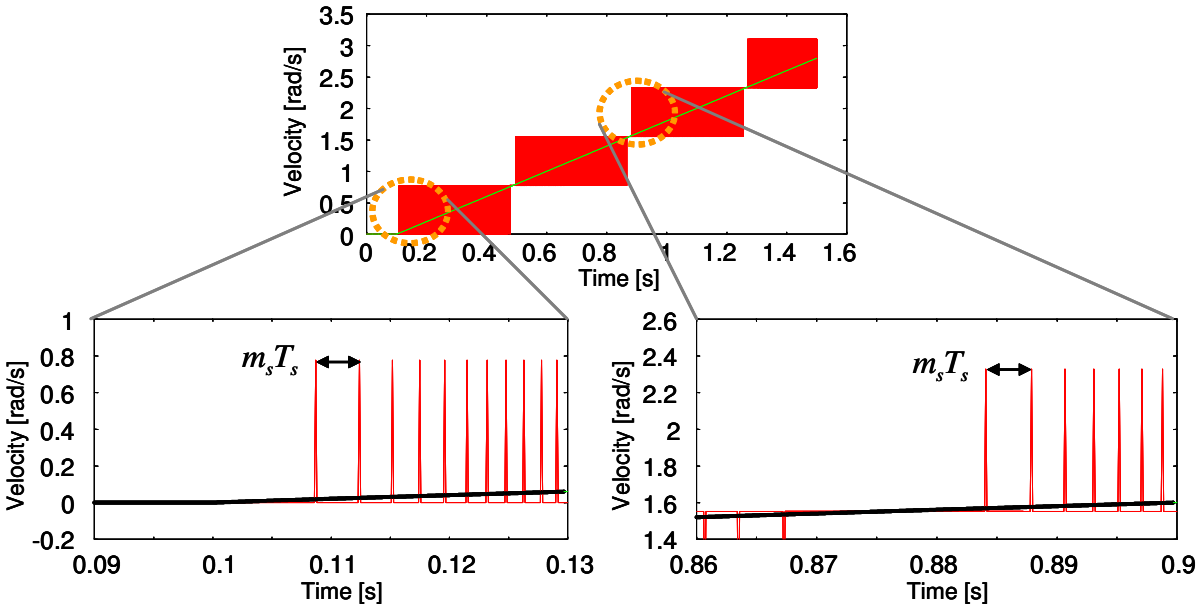


Fig.5-11: Pulse pattern in low and high-speed range

The procedures of velocity measurement are shown as follows:

1. count the pulse number $m_e(i)$ in each sampling period;
2. do not update the velocity value while the pulse numbers are constant; and
3. calculate and update the velocity value if the pulse numbers alter (i.e. if pulse alteration occurs).

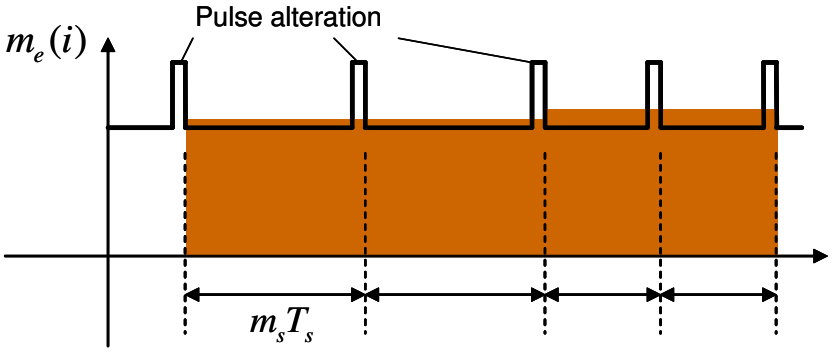


Fig.5-12: Pulse pattern in high-speed range

The velocity value is derived from (5.36).

$$\bar{\omega}(i) = \frac{2\pi \sum_{j=0}^{m_s} m_e(i-j)}{m_s P T_s} \tag{5.36}$$

Here, m_s is the sample number during interval of pulse alteration. The equation is quite similar to M method with averaging. The difference is that averaging calculation of the proposed method is synchronized with pulse alteration. Equations (5.37), (5.38) and (5.39) show that the performance is equivalent to T method. Furthermore, this method works in all speed ranges.

$$Q_V = \frac{2\pi}{m_s(m_s-1)PT_s} \tag{5.37}$$

$$\frac{Q_V}{V} = \frac{2\pi}{m_s(m_s-1)PT_sV} \tag{5.38}$$

$$T_m = m_sT_s \tag{5.39}$$

The proposed method is named "synchronous-measurement method (S method)" since its calculation is synchronous with pulse alteration. High accuracy of T method, which is synchronous not with pulse alteration but with pulses, is expanded to a high-speed range with this method.

The procedures of S method is carried out on a PC with RT-Linux in this study. One of the advantages of this method is that it is applicable to a system with a relatively long sampling period since it works in all speed ranges. Note that T method often requires an additional processor for a short sampling period to extend its measurable speed range. Furthermore, S

method is also applicable to a system with an auxiliary processor to acquire a shorter sampling period. Accuracy of the velocity measurement improves with the shorter sampling period.

5.6.2 Performance of velocity measurement methods

This subsection compares the theoretical performance of each method.

Fig. 5-13 shows the measurement time on each method. M method acquires the velocity information in every sampling period. Averaging makes the measurement time longer while it improves the measurement accuracy. T method is only applicable in a speed range lower than unit velocity, $\omega_u = 12.57\,\mathrm{rad/s}$. Here, unit velocity ω_u means the velocity on which exactly one pulse occurs during one sampling period. Measurement time depends on pulse interval in M/T method. The measurement time is long in the low-speed range while the measurement time is almost constant in the high-speed range. Measurement time is fluctuant in S method. Measurement time becomes long when pulse alteration does not often occur. In fact, the delay becomes large when the velocity is multiple of unit velocity. This deteriorates control performance when adverse effect of delay is larger than that of the quantization error. Therefore a method to modify the measurement time is introduced later in Section 5.8. In this method, velocity is calculated compulsory if measurement time exceeds a threshold.

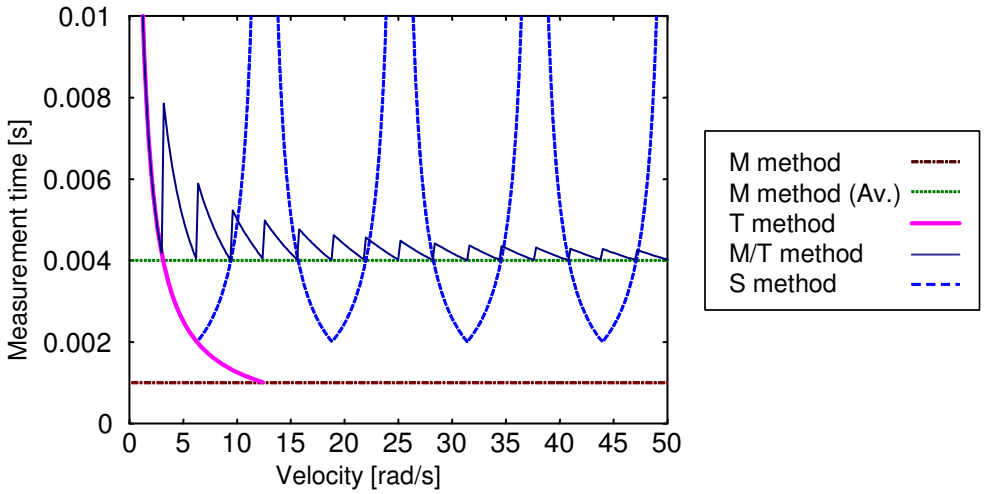


Fig.5-13: Measurement time on each method

Fig. 5-14 compares the measurement resolution on each method. M/T method is more accurate than T method in a velocity range from 3.14 rad/s to 12.57 rad/s. The reason is that its measurement time is modified to be longer than $T_c = 0.004$ while the measurement time of T method is shorter in the range. Meanwhile, M/T method has large Q_v, absolute velocity resolution, in a high-speed range. Accuracy of velocity measurement methods is often compared on Q_v/V, relative velocity resolution. Although M/T method has large Q_v in a high-speed range, Q_v/V is kept small in all speed ranges. In this point of view, M/T method is effective for many motion control systems since the relative velocity resolution, which determines accuracy of velocity feedback, is high in M/T method. However, acceleration control requires accuracy in absolute velocity resolution since the accuracy in the acceleration dimension is in proportion to Q_v, absolute velocity resolution. Hence small Q_v is indispensable for acceleration control. The result shows that S method achieves small Q_v in all speed ranges.

5.6.3 Simulation result

Simulations were executed to compare the accuracy of velocity measurement methods in the completely same conditions. Parameters were equal to those of the experiment. In each simulation, the same sequence of external force input was applied without any control input in order to compare just the accuracy of measurement.

Fig. 5-15 shows the velocity measurement results of M method, M/T method, and S method, respectively. M method was averaged and the sample number for averaging was 4. Velocity resolution was constant and large in M method. On the other hand, M/T method achieved high resolution in the low-speed range while the resolution was still low in the high-speed range. The result shows that high resolution in all speed ranges was achieved by S method.

5.7 Technical issues

5.7.1 Disordered pulse alteration

Fig. 5-16 shows velocity in an experiment when the flywheel was rotating around unit velocity. At the same time, the velocity values of M method also stand for pulse numbers in each sampling period. That is, a value of unit velocity 6.28 rad/s stands for 1 pulse and double of unit velocity 12.56 rad/s stands for 2 pulses. Zero velocity means no pulse occurred in the period.

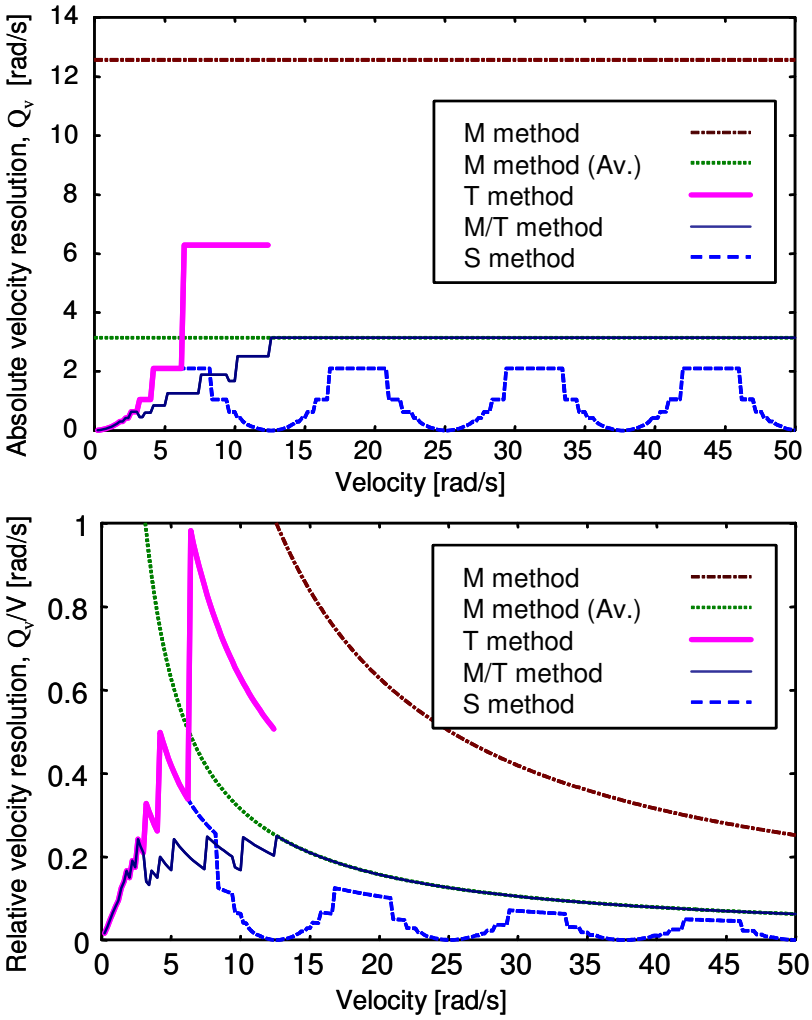


Fig.5-14: Measurement resolution on each method

Constant torque input without any feedback was given in the experiment so as to eliminate errors due to the control feedback.

From the result, disordered pulse alterations were confirmed around unit velocity. It is obvious by comparing the result with Fig. 5-4, a simulation result with an ideal model. The disordered pulse alterations also occur around other multiples of unit velocity. The measurement time of S method varies enormously with them. They enlarge the measurement error since accuracy of S method strongly depends on the accuracy of time measurement between two pulse alterations.

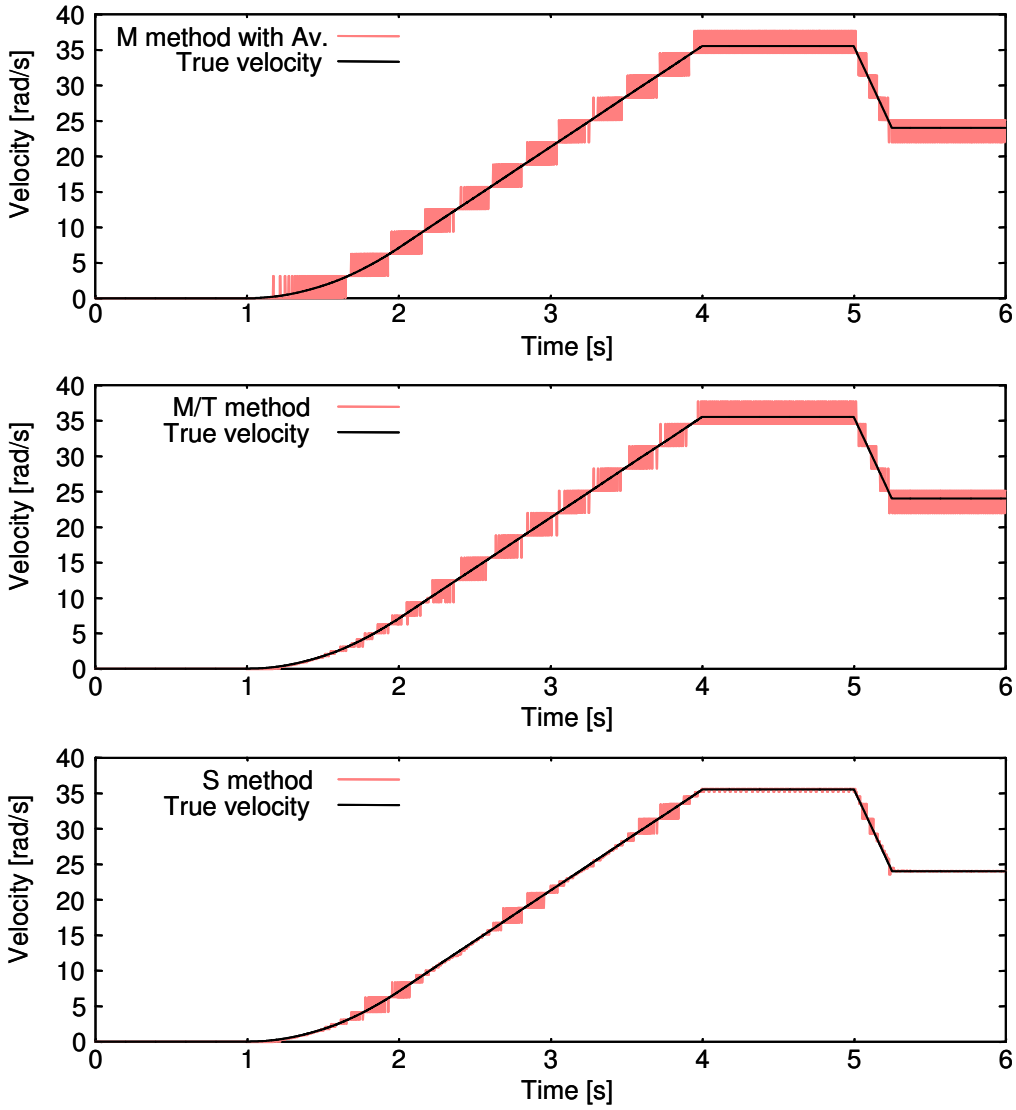


Fig.5-15: Velocity measurement on each method

A solution for this issue is indispensable for high performance control. Aspects of the disordered pulse alteration therefore were inquired to search the solution. The right side of Fig. 5-16, a close-up of the left side, implies the aspects as follows:

- positive and negative pulse alterations occur alternately;
- size of disordered pulse alterations are always 1.

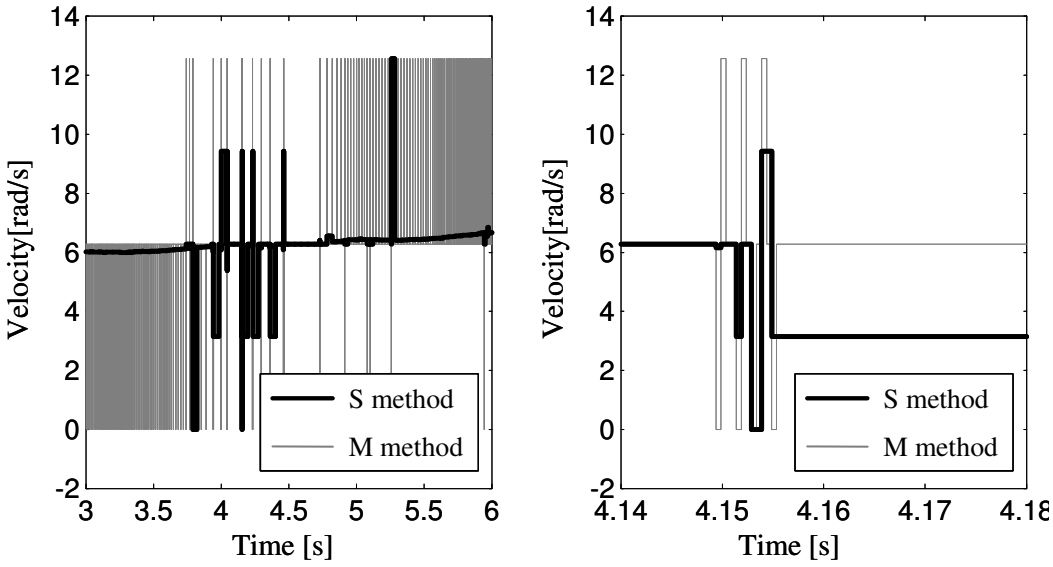


Fig.5-16: Error due to disordered pulse alteration

This kind of disordered pulse alterations are not expected from the ideal model of the optical encoder. The alterations seem to come from following terms:

- nonuniformity of scale;
- minute oscillation of shaft.

The minute oscillation of the shaft, which is mainly due to the noise in motor input current, is real while nonuniformity of the scale brings an irreal error. These two terms are indistinctive and small. Although they are small, they appear as disordered pulse alterations and cause seriously large errors. Hence the solution is to somehow ignore these terms without distinguishing them. The following discussion assumes that the disordered pulse alteration occurs because of shaft oscillation. However, the method in the discussion also holds when nonuniformity of the scale exists.

Fig. 5-17 shows how the minute oscillation raises alternate pulses when velocity is about zero. If the shaft oscillates around a threshold of position determination, measured position shifts even if the oscillation is small as shown in the left figure. Positive and negative pulse alterations occur alternately each time the shaft overstrides the threshold as shown in the right figure. In

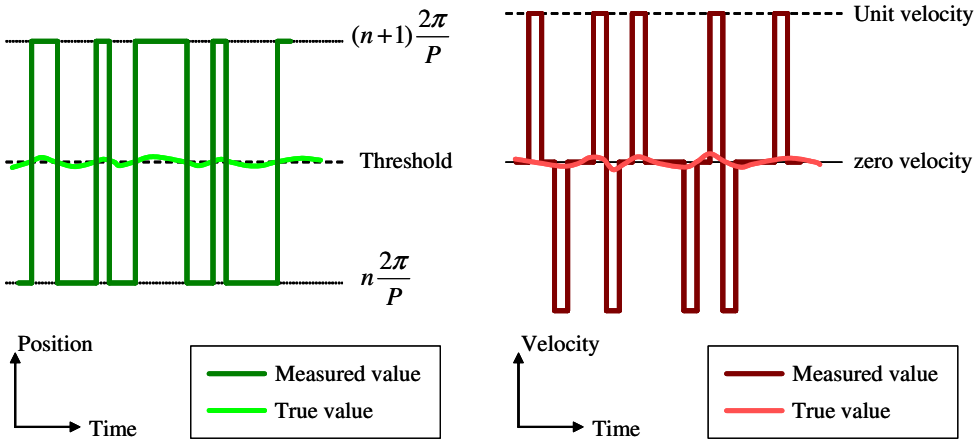


Fig.5-17: Mechanism of alternate pulse alteration

this way, disordered pulse alterations occur with minute oscillation. Lee and Song [79] have already remarked that an error on velocity measurement occurs when the sign of the velocity flips. However, they have not shown any solution for this problem.

The alternate pulses are often due to quite small oscillation while they appear as extremely large oscillation in measured values. Indeed, average velocity between alternate pulses is 0 because the detected positions of alternate pulses are physically identical as shown in [79]. The solution to solve this issue is to cancel out the pulses when positive and negative pulses occur alternately.

This method is applicable not only around zero velocity but also around multiple of unit velocity. The mechanism of alternate pulse alterations around multiple of unit velocity are similar to the mechanism around zero velocity. Alternate pulse alterations should be canceled out in the same fashion around multiple of unit velocity.

The procedures of velocity measurement are revised as follows:

1. count the pulse number $m_e(i)$ in each sampling period;
2. do not update the velocity value while the pulse numbers are constant; and
3. calculate and update the velocity value when the pulse alteration occurs,
    - if sign of alteration is same to the last one,

$$\bar{\omega}(i) = \frac{2\pi \sum_{j=0}^{m_s} m_e(i-j)}{m_s P T_s},$$

- if sign of alteration is different to the last one,
  $\bar{\omega}(i) = \frac{2\pi m_e(i-1)}{PT_s}$ (multiple of unit velocity).

Note that regular pulse alterations are always the same sign except the moment the velocity overstrides the multiple of unit velocity.

Fig. 5-18 compares the measurement results with and without the proposed modification. It is confirmed that oscillation due to the alternate pulse alterations practically disappeared with the modification.

5.7.2 LPF for unscheduled measurement

The fluctuant measurement time in S method requires some modifications on the system since the simplest control system is based on an assumption of constant sampling rate. It is a common issue for S method, T method and M/T method, which are velocity measurement methods synchronous with pulses.

Firstly, the digitized LPF for velocity measurement should be modified for fluctuant measurement time due to S method. S method calculates velocity at the moment a pulse alteration occurs with m_s samples of interval. The calculated velocity stands for an average velocity of m_s samples. Therefore the digitized LPF should be applied for m_s times. For this filtering process, the author calculated an equivalent value from the solution of an analysis of the LPF. Although it is also available to execute the filtering process by repetitive calculation of digitized LPF, it may require much calculated amount when m_s is large. Furthermore, the analytical solution provides much accurate filtering since the digitized LPF is approximation of an analogue LPF.

The analytical solution of normal LPF is given as follows:

$$y = (1 - \exp^{-GT_s})(u - y_0) + y_0. \tag{5.40}$$

Here, u stands for an input to the LPF and y stands for an output. G is a cutoff frequency of the LPF. y_0 is the initial value on each sampling period. It is assumed that a constant input u is given during the sampling period T_s.

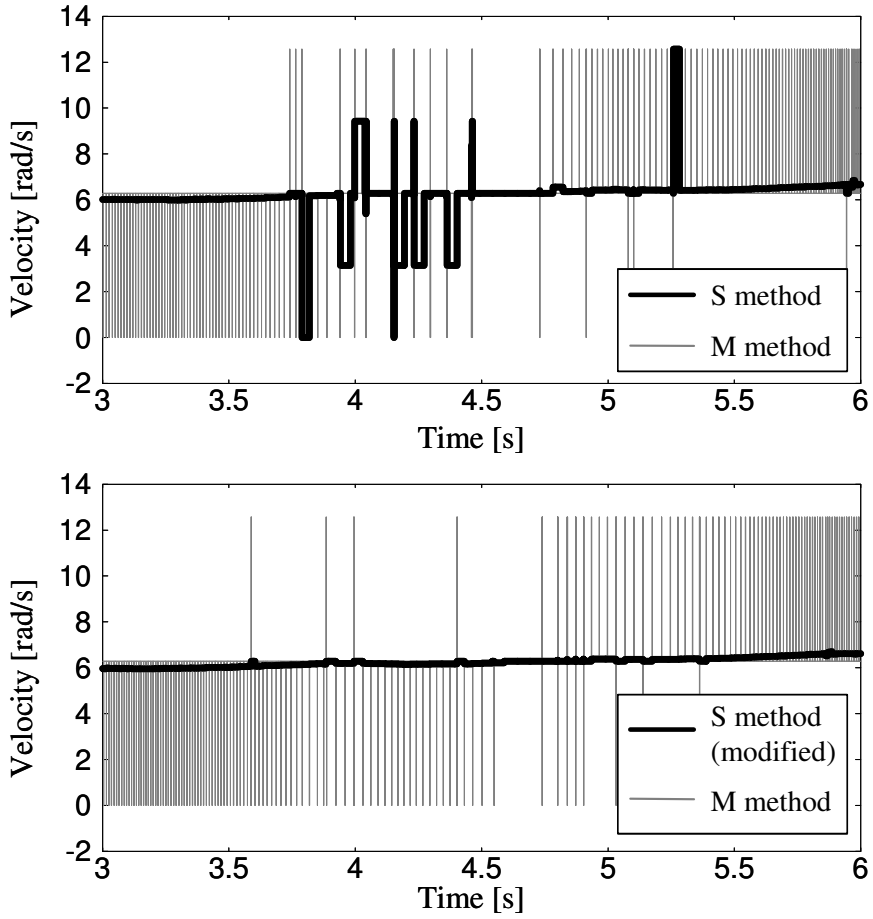


Fig.5-18: Velocity measurement with alternate pulse alteration

Equation (5.40) is expanded to (5.41), a LPF for a fluctuant measurement time.

$$y = (1 - \exp^{-Gm_s T_s})(u - y_0) + y_0. \tag{5.41}$$

Equation (5.41) is substituted to the conventional LPF for velocity measurement and that for disturbance observer. The response is equal to the disturbance observer realized in discrete time by using invariant impulse response Z-transform [80]. However, the proposed method has less calculation process and it does not have any transitional error on the initial stage of control. Furthermore, the proposed method is applicable to fluctuant measurement time in contrast to the disturbance observer with Z-transform.

Secondly, the form of disturbance observer should also be modified due to the fluctuant

measurement time. The general form of disturbance observer shown in Fig. 5-2 is for a constant measurement time. We apply and modify the original form of disturbance torque estimation as shown in Fig. 5-3, equivalent transformation of Fig. 5-2. Velocity information is calculated synchronous to a pulse alteration. Input current I_a is also given to the disturbance observer synchronous to the pulse alteration. The LPF in disturbance observer is modified in the same way described above. Finally the disturbance torque is calculated synchronous to the pulse alteration. The form of the modified disturbance observer is shown in Fig. 5-19.

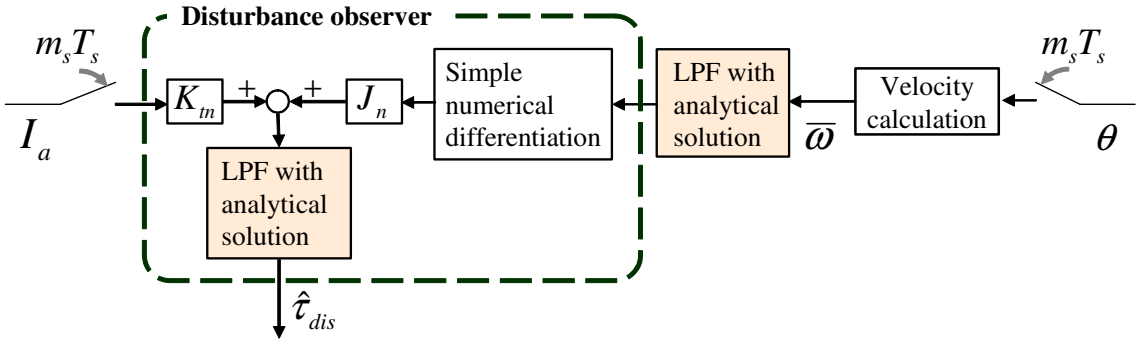


Fig.5-19: Disturbance observer modified for fluctuant measurement time

5.8 Modification of measurement time

Table 5.3 compares the performances of the velocity measurement methods. We focus on measurement time and resolution. These two factors are in trade-off relationship while either factor may deteriorate control performance. Appropriate measurement time is required for a robust control system. In fact, an excessive delay sometimes deteriorates the control performance in the T method. Periomatic [81] is a method to calculate the speed when measurement time is longer than the latest one. The constant elapsed time (CET) method [82] has a measurement time with small variation. Furthermore, this method provides a compromise between resolution and measurement time. Measurement time is determined from necessary resolution in this method.

This section discusses how to optimize the measurement time. A performance index is introduced to assess both measurement time and resolution. These two factors are assessed in the

Table 5.3: Performance of velocity measurement methods

	accuracy	length of delay	fluctuation of delay	plant model information
M method	low	short	determinable	unnecessary
T method				
in low speed range	high	long	indeterminable	unnecessary
in high speed range	n/a	n/a	n/a	n/a
M/T method				
in low speed range	high	long	indeterminable	unnecessary
in high speed range	low	short	indeterminable (small variation)	unnecessary
CET				
in low speed range	high	long	indeterminable (small variation)	unnecessary
in high speed range	low	short	indeterminable (small variation)	unnecessary
S method				
in low speed range	high	long	indeterminable	unnecessary
in high speed range	high	long	indeterminable	unnecessary
Instantaneous speed observer	high	short	determinable	necessary
Kalman filter	high	short	determinable	necessary

same dimension by figuring out the effect of a measurement delay in the velocity dimension. Based on the error assessment, a method to modify the measurement time is proposed.

5.8.1 Assessment of measurement time and quantization error

As described above, measurement time and resolution are in trade-off relationship in any method. Either a large measurement time or low resolution may deteriorate the control performance. A performance index is therefore introduced to determine an appropriate measurement

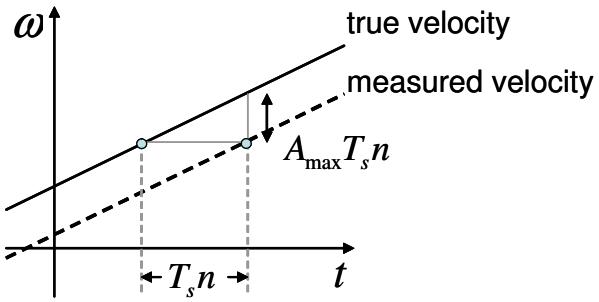


Fig.5-20: Velocity error due to measurement delay

time. It assesses the adverse effect of the measurement delay and the quantization error in the velocity dimension.

Measured velocity value $\bar{\omega}$ contains an error less than Q_V while the measurement delays for $T_m = T_s n$. In T method and M/T method, m_s shows the number of sampling periods instead of n. However, n is substituted for m_s in the following discussion.

The true velocity alters during the measurement time and this alteration could be treated as an additional error in the velocity dimension. Fig. 5-20 shows the concept. This additional error is named "delay error". Assuming maximum acceleration A_{max} occurs constantly during the measurement time, the maximum error in the velocity dimension is figured out in (5.42).

$$e_{max} = Q_V + A_{max} T_s n \tag{5.42}$$

e_{max} is sum of a quantization error and a delay error. It is named "sum error" and it directly affects the control performance. Fig. 5-21 shows relationship between e_{max} and n.

5.8.2 Optimization of measurement time in M method

Although quantization errors decrease as measurement time is enlarged, the effect of the measurement delay increases the sum error. There is a best measurement time with the smallest sum error. Measurement time in M method is a product of the sampling period T_s and the number of averaging sample n. T_s usually depends on the limitation of hardware e.g. PC or micro controller. On the contrary, n can be arbitrarily decided in M method. Therefore n could be optimized.

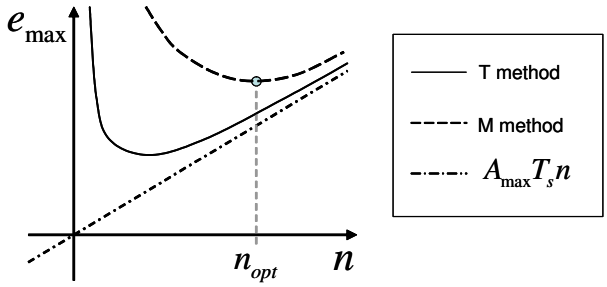


Fig.5-21: Relationship between e_{max} and n

By substituting (5.29) to (5.42), e_{max} in M method is derived as shown in (5.43)

$$e_{max} = \frac{2\pi}{PT_s}\frac{1}{n} + A_{max}T_s n \tag{5.43}$$

Equation (5.44) is derived by partially differentiating (5.43) with respect to n.

$$\frac{\partial e_{max}}{\partial n} = -\frac{2\pi}{PT_s}\frac{1}{n^2} + A_{max}T_s \tag{5.44}$$

e_{max} is the smallest when $\frac{\partial e_{max}}{\partial n} = 0$. Hence n_{opt}, the optimized number of averaging samples, is figured out as follows:

$$-\frac{2\pi}{PT_s}\frac{1}{n^2} + A_{max}T_s = 0$$

$$n_{opt} = \left[\sqrt{\frac{2\pi}{PA_{max}}}\frac{1}{T_s} + 1\right] \tag{5.45}$$

where operators $[\cdot]$ return the greatest integer less than the content.

5.8.3 Modification in T method

In contrast to M method, the measurement time in T method is indeterminable. Therefore it is a problem of T method that measurement time becomes extremely long when no pulse comes. In this case, it is better to compulsively measure the velocity before the delay error affects the control system. The velocity value is 0 in this case since detected displacement of the wheel is 0. The accuracy of compulsive measurement is equal to that of M method with averaging since it is just averaging the zero velocity in essence. Although compulsive measurement provides inaccurate velocity value, it avoids further deterioration due to the long measurement time.

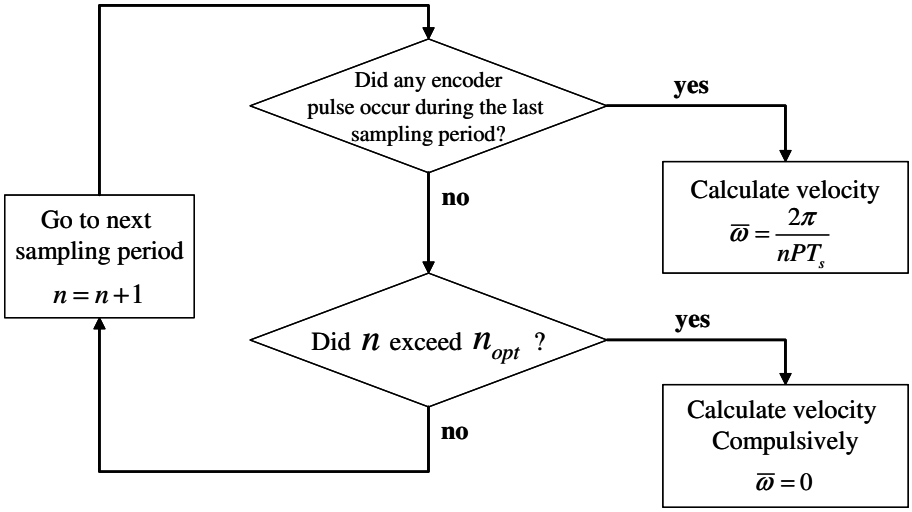


Fig.5-22: Procedure of velocity measurement

Fig. 5-22 shows the procedure of velocity measurement with measurement time modification. The description of the procedure is as follows: A pulse from the optical encoder keeps being observed on each sampling period. If a pulse is detected, velocity is measured by T method. Velocity is also compulsively measured when $n \geq n_{opt}$. n_{opt} is the threshold here since a sum error of the compulsive measurement increases with longer measurement time. If no pulse is detected and $n < n_{opt}$, the system waits for the next sampling period.

When the interval time is longer than $T_s n_{opt}$, velocity measurement is executed compulsively. A sum error on compulsive measurement is figured out as follows:

$$e_{max} = \frac{2\pi}{PT_s}\frac{1}{n_{opt}} + A_{max}T_s n_{opt}. \tag{5.46}$$

e_{max} would not exceed this value even if interval time of pulse is longer. Consequently, the compulsive measurement prevents the further deterioration due to a long interval time. This modification is effective not only for T method but also for M/T method since M/T method also has the same problem of a long pulse interval.

Fig. 5-23 shows the plot of e_{max} when measurement time is modified. Note that horizontal axis shows the interval time of pulse variation instead of n. If the interval time of pulse variation is longer than $T_s n_{opt}$, the compulsive measurement provides a velocity value with accuracy of M method at $n = n_{opt}$.

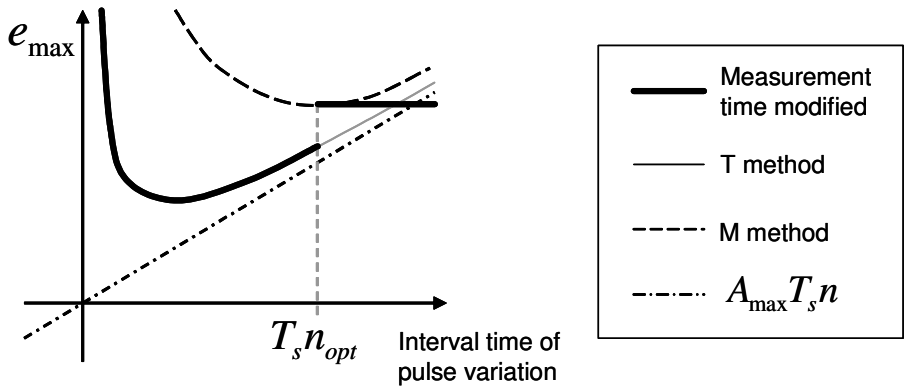


Fig.5-23: Sum error when measurement time is modified

5.8.4 Simulation

Simulation of velocity measurement was executed to assess the measurement error. Here, the error is defined as the difference between true velocity and measured velocity. The error was assessed by two indices: a mean square error (MSE) and a maximum error. The two indices were derived on different numbers of averaging sample. Table 5.4 shows the results. Mechanical parameters were the same to the experimental setup. Measurement was executed considering the quantization error of the optical encoder.

Table 5.4: Mean square error in simulation(without control)

Number of Av. sample	10	20	30	40	50	
MSE [rad^2/s^2]	17.90	5.335	2.613	1.552	1.019	
Maximum error [rad/s]	24.82	12.37	8.249	6.333	5.103	

70	100	150	200	300	500	700	1000
0.6730	0.5573	0.5970	0.7160	1.013	1.681	2.401	3.559
3.691	2.675	1.954	1.645	1.463	1.599	1.915	2.499

Rotational velocity was calculated while no feedback control was applied. The trajectory of the flywheel was given as follows:

$$\theta(t) = 10 \sin 3t \tag{5.47}$$

where $\theta(t)$ denotes the angle of the flywheel. The given acceleration is derived by differentiating (5.47) for twice as shown in (5.48):

$$\ddot{\theta}(t) = -90 \sin 3t \tag{5.48}$$

Consequently, maximum acceleration here is $A_{max} = 90$. Then, the optimal number of averaging sample is derived as follows.

$$n_{opt} = \left[\sqrt{\frac{2\pi}{PT_s^2 A_{max}}} + 1 \right] = 237 \tag{5.49}$$

The simulation results show that the maximum error was the smallest when the number of averaging sample n is about n_{opt}. On the other hand, MSE was the smallest when n is about 100 to 150. Since a quantization error and a delay error in e_{max} are the maximum values, they differ from the mean values. This seems to be the reason that the minimum point of MSE is different from n_{opt}. However, it is difficult to forecast the exact effect of a quantization error and a delay error. n_{opt} therefore should be derived as a conservative standard to design a control system.

5.9 Experiment

This section shows experimental results to verify the validity of S method. The main focus is how robustness of a motion control system improves with the method.

5.9.1 Improvement on robustness

We applied PD control with disturbance observer to the experimental system. Command velocity was kept constant after it was raised from 0 rad/s to 40 rad/s in 5 seconds. Virtual disturbance of 0.2 Nm was given for 0.5 second when 7 seconds passed from the beginning of the experiment. The disturbance was given with adding a current input to the control system. Fig. 5-24 shows the results. Here, k_v denotes a derivative gain of the PD controller. In order to show a fair comparison, all of the response values in the figure are derived from M method

with a LPF. In other words, the values in the figure are just to show and they are not used for control. G_v, G_{dis} and k_v were changed in each experiment while k_p, a proportional gain, was 20.0 all the time.

The motor made a sound noise when G_v and G_{dis} were larger than 20.0 rad/s in M method. A large oscillation was confirmed in the result when $G_v = G_{dis} = 50.0$ in M method. Acceleration control with $G_v = G_{dis} = 50.0$ was achieved without a sound noise in S method. The result shows that the performance of acceleration control improved with S method. Furthermore, k_v could be further heighten without any destabilization because of the high resolution of measured velocity. It is confirmed from the result that an error on velocity control was reduced with higher k_v.

5.9.2 Improvement on force sensing

External force observer (EFOB) [59] is an effective tool to estimate external force on robots. Its force sensing accuracy depends on the accuracy of velocity measurement. Therefore, not only control performance but also force sensing accuracy improves drastically by S method. Touching motion on 1 DOF manipulator was experimented to compare the force sensing on respective methods.

Fig. 5-25 shows the experimental result. Compliance control was applied while a position command to press a fixed aluminum plate was given. The first two positive bumps are the force response during pressing motion. The operator touched the manipulator twice after that. Two negative bumps show the human force. External force was measured in parallel in one experiment by three types of measurement. The authors intended to compare S method, M method and true value. Since it is difficult to know the true value, we substituted it by output of EFOB with an 81000 PPR encoder. We compared M method and S method with an 8100 PPR encoder, a lower resolution encoder. 8100 PPR was simulated by dividing the pulse number from the 81000 PPR encoder by 10. Large noise was confirmed on M method while the noise on S method was even smaller than that of M method with the 81000 PPR encoder.

5.9.3 Verification of measurement time modification

Two kinds of experiments were executed for the verification of measurement time modification. In the first experiment, the flywheel was controlled by M method with averaging. The mean

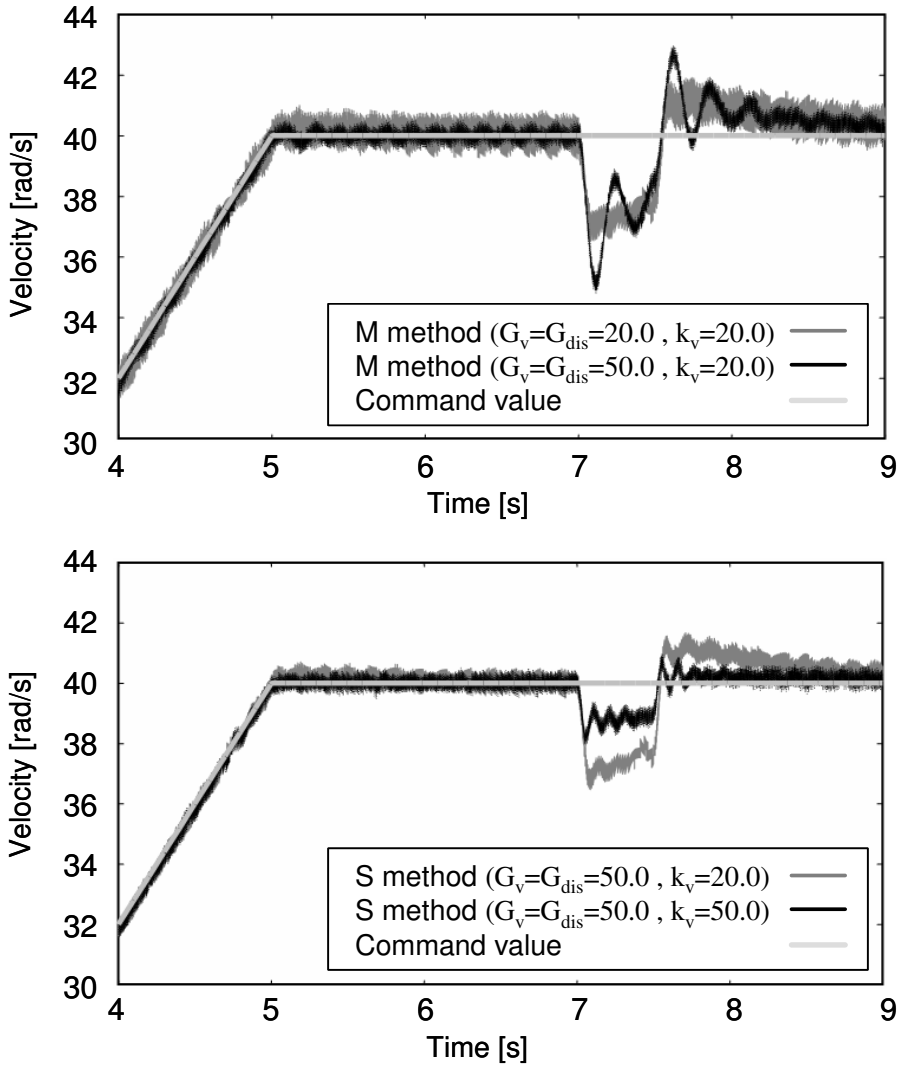


Fig.5-24: Experimental result of PD control with disturbance observer

square error and the maximum error were shown to compare control performance. In the second experiment, the flywheel was controlled by T method. Measurement time modification shown in Section 5.8 was applied. It is shown that the control performance around zero velocity was improved with the measurement time modification.

M method with averaging was applied to the PD control of a flywheel. Disturbance observer

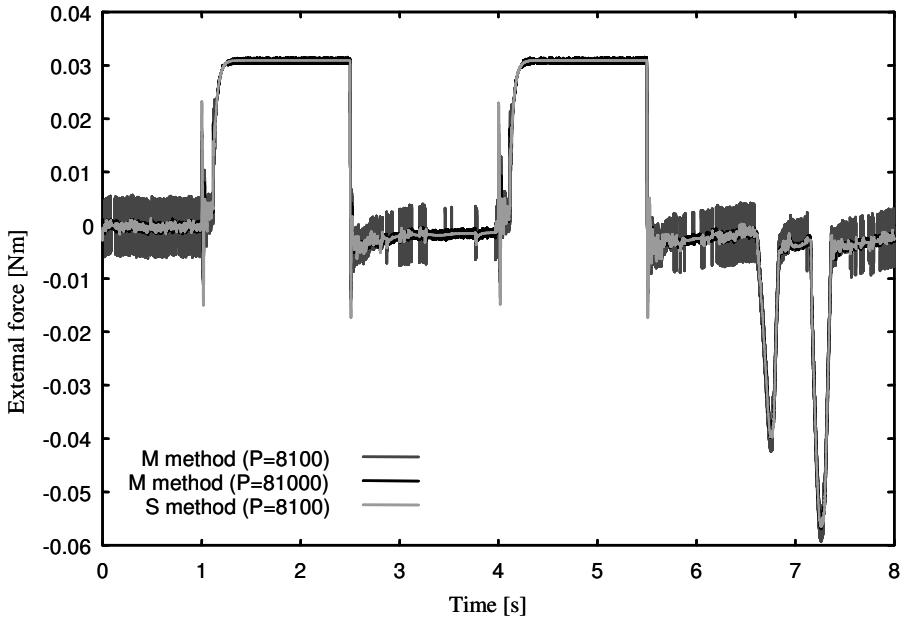


Fig.5-25: Comparison on force measurement

was also applied. The command angle was given as follows:

$$\theta_{cmd} = 3\sin 0.2t \tag{5.50}$$

Table 5.5 shows the control parameters in the experiment.

Table 5.5: Control parameters

Position gain		K_p	30.0
Velocity gain		K_v	6.0
Cutoff frequency of DOB	[rad/s]	G_{dis}	5.0
Cutoff frequency of LPF	[rad/s]	G_v	10.0

MSE and the maximum error were derived and compared on different n. Here, the error is defined as the difference between command velocity and response velocity of the flywheel so as to compare the control performance instead of measurement accuracy. In every experiment, response velocity for the error assessment was derived through LPF after calculating the finite-

difference derivative of response position. In other words, velocity measured for control input was not used for error assessment. This is for comparing errors in the same measurement condition. Fig. 5-26 compares velocity measured for control input, i.e. results in different measurement conditions. It shows that the wave form of measured velocity varies significantly depending on n.

On the other hand, Table 5.6 shows the results compared in the same condition. There is no result when $n = 200$ since the control system became unstable.

Table 5.6: Mean square error in experiment (with control)

Number of Av. sample	2	3	5	10	20	30	50	100	200
MSE [rad^2/s^2]	0.001119	0.001263	0.001229	0.001217	0.001248	0.001231	0.001354	0.001376	unstable
Maximum error [rad/s]	0.1757	0.1698	0.1638	0.1668	0.1662	0.1662	0.1621	0.1536	unstable

Although the wave form varied significantly, MSE and the maximum error rarely altered. The control performances were about the same in every experiment. This indicates that control performance mainly depends on the structure and parameters of controller. On the other hand, the performance does not directly depend on the measurement accuracy. However, the result at $n = 200$ shows that measurement accuracy determines the stability of the control system. High stability allows heightening control gains. In sum, high stability thanks to measurement time modification allows high-performance control by heightening high control gains although the modification itself does not much improve the control performance.

5.10 Summary of chapter

This chapter proposed S method, a novel method of velocity measurement for motor drives with optical encoders. In S method, the velocity measurement is synchronized with alteration

of pulse numbers. High accuracy of T method is acquired in all speed ranges due to this synchronization. Furthermore, other estimation methods are applicable in addition since the method does not require any model of a control object. Multirate sampling method with a shorter output sampling period is a good candidate to combine with S method and to complement each other. Simulation and experimental results verified the validity of the method. Although the method is a fundamental technique for all motion control systems with optical encoder, it is particularly effective for acceleration control systems.

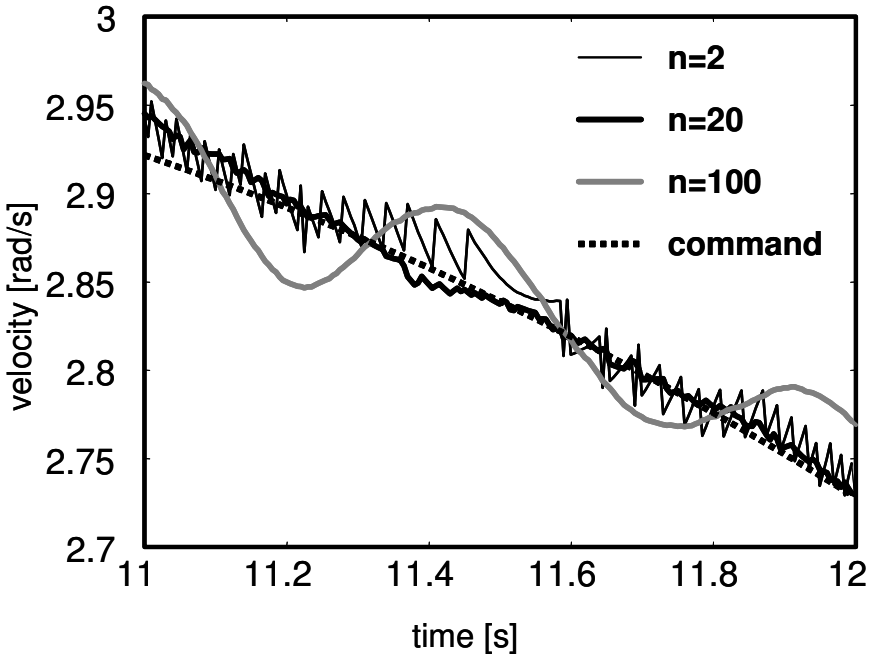


Fig.5-26: PD control applying M method with averaging

Chapter 6

Conclusions

This study aimed at improving adaptability of robots to human environment while the issue has many factors. The author categorized the factors into three parts shown as follows:

- sophisticated motion planning;
- flexible controller design; and
- robust motion control.

Firstly, some methods for a biped robot working in human environment were proposed. The studies showed importance of motion planning in known environment. On the other hand, compliance control is requisite for a robot in unknown environment while it may degrade tracking performance. It was confirmed from the study that a hybrid control system based on environmental modes acquires both environmental adaptability and tracking performance.

Secondly, a method to design such a hybrid control system was proposed. The method should have flexibility to deal with exceptions and environmental variation. The method should also be for a large-scale system since a robot in human environment needs hyper-DOF. The idea of functionality provides a design framework for a large-scale system that deals with exceptions and environmental variation in a unified manner. The study implies that the concept of "controller design as combination of detachable components" will be a key for a robot in human environment.

Finally, performance improvement on acceleration control was discussed. S method, a velocity measurement method for robust acceleration control, was proposed. Acceleration control is indispensable for a hybrid control system based on functionality since acceleration represents both position and force in a unified manner. Furthermore, robust acceleration control enhances

contact stability to human environment. Although S method is a fundamental technique for all motion control systems with optical encoder, it is particularly effective for acceleration control systems.

The main issue for establishing a robot adaptive to human environment is how to deal with the interactions between human environment and a robot. The three approaches in this study are to solve the issue. An experiment on parallel link manipulators in Section 4.8 shows an example of combining the three approaches. The result shows that adaptability to human environment was improved by the combination. Three approaches in this study are all requisites for adaptation to human environment since motion control is based on layers of these techniques. Although further development is still required for a robot in practical use, this paper proposed fundamental techniques of motion control for adaptation to human environment and showed a possibility of future robot systems working in human environment.

References

[1] J. Verne. *Le Voyage Dans La Lune(From the Earth to the Moon).* Sogen SF bunko, (in Japanese).

[2] Villiers de l'Ile Adam. *L'Eve Future.* Tokyo-sogen-sha, 1996 (in Japanese).

[3] K. Capec. *R. U. R.* Kodansha bunko, (in Japanese).

[4] M. Shelley. *Frankenstein: or The Modern Prometheus.* Kadokawa Shoten, 1994 (in Japanese).

[5] J. Turney. *Frankenstein's Footsteps: Science, Genetics and Popular Culture.* Yale University Press, 1998.

[6] Japanese robot industry in the 21 century. Japan Robot Association, http://www.jara.jp/ (in Japanese).

[7] K. Hirai, M. Hirose, Y. Haikawa, and T. Takenaka. The development of honda humanoid robot. *IEEE Int. Conf. on Robotics & Automation, pp. 1321–1326*, 1998.

[8] *J. SICE, Vol. 42, No. 6*, 2003 (in Japanese).

[9] J. Miura and Y. Shirai. Modeling obstacles and free spaces for a mobile robot using stereo vision with uncertainty. *Proc. Int. Conf. Robotics & Automation, pp. 3368–3373*, 1994.

[10] P. Moutarlier and R. Chatila. Incremental free-space modeling from uncertain data by an autonomous robot. *Proc. IEEE Int. Workshop on Intelligent Robots and Systems, pp. 1052–1058*, 1991.

[11] A. Elfes. Sonar-based real-world mapping and navigation. *Int. J. Robotics and Automation, Vol. 3, No. 3, pp. 249–265*, 1987.

[12] U. R. Dhond and J. K. Aggarwal. Structure from stereo: A review. *IEEE Trans. Systems, Man, and Cyvernet., Vol. 19, No. 6, pp. 1489–1510*, 1989.

[13] T. Murakami. Force control and environmental recognition in robust motion control. *J. JSPE, Vol. 69, No. 10, pp. 1387–1390*, 2003 (in Japanese).

[14] D. Ito, T. Kageyama, J. Suzuki, T. Tsuji, M. Morisawa, and K. Ohnishi. A design of decentralized control system in unstructured environment. *IEEJ Trans. Industry Applications, Vol. 123, No. 10, pp. 1219–1226*, 2003 (in Japanese).

[15] M. Morisawa, T. Tsuji, Y. Nishioka, K. Akuzawa, H. Takahashi, and K. Ohnishi. Contact motion in unknown environment. *Proc. IEEE Int. Conf. Industrial Electronics, Control and Instrumentation (IECON'03), pp. 992–996*, 2003.

[16] S. Katsura and K. Ohnishi. Advanced motion control for wheelchair based on environment quarrier. *IEEJ Trans. Industry Applications, Vol. 125, No. 7, pp. 698–704*, 2005 (in Japanese).

[17] H. Choset, I. Konusksven, and A. Rizzi. Sensor based planning: A control law for generating the generalized voronoi graph. *Proc. of IEEE Int. Conf. on Autonomous Robots*, 1977.

[18] R. A. Brooks. A robust layered control system for a mobile robot. *IEEE J. R & A, Vol. RA-2, No. 1, pp. 14–23*, 1986.

[19] M. Habib and S. Yuta. Map repersentation, on-line path planning and navigation for a mobile robot in a vuiling environment. *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, pp. 306–315*, 1989.

[20] J. J. Kuffner, K. Nishiwaki, S. Kagami, M. Inaba, and H. Inoue. Footstep planning among obstacles for biped robots. *Proc. IEEE Int. Conf. on Intelligent Robots & Systems, pp. 500–505*, 2001.

[21] T. Tsuji and K. Ohnishi. Global step planning of dynamic biped locomotion considering obstacles. *The 28th Int. Conf. on Industrial Electronics, Control and Instrumentation(IECON'02)*, 2002.

[22] M. Okada, K. Tatani, and Y. Nakamura. Polynomial design of the nonlinear dynamics for the brain-like information processing of whole body motion. *Proc. IEEE Int. Conf. Robotics & Automation, pp. 1410–1415*, 2002.

[23] H. Asama, M. K. Habib, I. Endo, K. Ozaki, A. Matsumoto, and Y. Ishida. Functional dis-

tribution among multiple mobile robots in an autonomous and decentralized robot system. *Proc. IEEE Int. Conf. Robotics & Automation, pp. 1921–1926*, 1991.

[24] T. Ueyama, T. Fukuda, F. Arai, Y. Katou, S. Matsumura, and T. Uesugi. A study on dynamically reconfigurable robotic systems (10th report, distributed control structure for organization using an evaluation of network energy for group structure of cebot). *Journal of JSME, Part C, Vol. 58, No. 549, pp. 132–139*, 1992 (in Japanese).

[25] Y. U. Cao, A. S. Fukunaga, and A. B. Kahng. Cooperative mobile robotics: Antecedents and directions. *Autonomous Robots, No. 4, pp.7-27*, 1997.

[26] S. Ichikawa and F. Hara. Characteristics of object-searching and object-fetching behaviors of multi-robot system using local communication. *Proc. IEEE Int. Conf. on Systems, Man, and Cybernetics(SMC'99), Vol. 4, pp. 775–781*, 1999.

[27] N. Nishikawa, Y. Fujimoto, T. Murakami, and K. Ohnishi. Variable compliance control for 3 dimensional biped robot considering environmental fluctuations. *IEEJ Trans. Industry Applications, Vol. 119-D, No. 12, pp. 1507–1514*, 1999 (in Japanese).

[28] K. Ohnishi, M. Shibata, and T. Murakami. Motion control for advanced mechatronics. *IEEE/ASME Trans. Mechatronics, Vol. 1, No. 1, pp. 56–67*, 1996.

[29] M. Doi and S. Amano, editors. *Special issue — Distributed Real-Time Networks for Supporting Human Activity.* IPSJ Magazine, Vol. 44, No. 1, pp. 3–51, 2003 (in Japanese).

[30] Honda worldwide -asimo. http://world.honda.com/ASIMO/ (in Japanese).

[31] Sony global - qrio -. http://www.sony.net/SonyInfo/QRIO/ (in Japanese).

[32] Welcome to humanoid robotics group. http://www.is.aist.go.jp/humanoid/ (in Japanese).

[33] http://www.toyota.co.jp/en/news/04/1203_1d.html.

[34] M. H. Raibert, H. B. Brown, and S. S. Murthy Jr. 3d balance using 2d algorithms. *In First Int. Symposium of Robotics Research, (MIT Press, Cambridge), pp. 279–301.*, 1984.

[35] R. R. Playter. Passive dynamics in the control of gymnastic maneuvers. *Ph.D Thesis, Dept. of Aeronautical and Astronautical Engineering, Massachusetts Institute of Technology*, 1994.

[36] J. Pratt and G. Pratt. Intuitive control of a planar bipedal walking robot. *Proc. of the IEEE Int. Conf. Robotics and Automation (ICRA '98)*, 1998.

[37] T. McGeer. Passive dynamic walking. *Int. J. Robotics Research, Vol.9, No. 2, pp. 62–82*, 1990.

[38] K. Loffler, M. Gienger, and F. Pfeiffer. Sensor and control design of a dynamically stable biped robot. *IEEE Int. Conf. Robotics & Automation, pp. 484–490*, 2003.

[39] M. Vukobratovic, B. Borovac, D. Surla, and D. Stokic. Biped locomotion: Dynamics, stability, control and application. *Scientific Fundamentals of Robotics, Vol. 7*, 1990.

[40] H. Miura and I. Shimoyama. Dynamic walk of biped. *Int. J. of Robotics Research, Vol. 3, No. 2, pp. 60–74*, 1984.

[41] J. Furusho and M. Masubuchi. Control of a dynamical biped locomotion system for steady walking. *Trans. ASME, Journal of Dynamic Systems, Measurement and Control, Vol. 108, No. 2, pp. 111–118*, 1986.

[42] S. Kajita, T. Yamaura, and A. Kobayashi. Dynamic walking control of a biped robot along a potential energy conserving orbit. *IEEE Trans. Robotics & Automation, Vol.8, No.4, pp.431–438*, 1992.

[43] K. Mitobe, K. Yajima, and Y. Nasu. Control of walking robots by manipulating the zero moment point. *J. RSJ, Vol. 18, No. 3, pp. 359–365*, 2000 (in Japanese).

[44] H. Minakata and Y. Hori. A study on biped bike—analysis and control of sagittal plane motion—. *IEEJ Trans. Industry Applications, Vol. 117-D, No. 9, pp. 1057–1062*, 1997 (in Japanese).

[45] T. Sugihara and Y. Nakamura. Whole-body cooperative balancing of humanoid robot using cog jacobian. *Proc. of the 2002 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2002.

[46] T. Tsuji and K. Ohnishi. A control of biped robot which applies inverted pendulum mode with virtual supporting point. *Proc. 7th IEEE Int. Workshop on Advanced Motion Control, pp. 478-483*, 2002.

[47] M. Yagi and V. Lumelsky. Biped robot locomotion in scenes with unknown obstacles. *Proc. IEEE Int. Conf. Robotics & Automation, pp. 375–380*, 1999.

[48] E. R. Dunn and R. D. Howe. Foot placement and velocity control in smooth bipedal walking. *Proc. IEEE Int. Conf.Robotics & Automation, pp. 578–583*, 1996.

[49] N. Nishikawa et al. An approach to stable motion control of biped robot with unknown load by torque estimator. *Proc. 5th Int. Workshop on Advanced Motion Control, pp. 129–134*, 1998.

[50] K. Yoshida. The spacedyn:a matlab toolbox for space and mobile robots. *J. SICE, Vol. 38, No. 2, pp. 138–143*, 1999 (in Japanese).

[51] M. Morisawa and K. Ohnishi. Motion control taking environmental information into account. *European Power Electronics and Drives Journal, Vol. 12, No. 4, pp. 37–41*, 2002.

[52] D. A. Lawrence. Stability and transparency in bilateral teleoperation. *IEEE Trans. Robotics and Automation, Vol. 9, pp. 624–637*, 1993.

[53] Y. Matsumoto, S. Katsura, and K. Ohnishi. An analysis and design of bilateral control based on disturbance observer. *Proc. IEEE Int. Conf. Industrial Technology, ICIT'03-MARIBOR, pp. 802–807*, 2003.

[54] D. Lee and P.Y. Li. Passive bilateral feedforward control of linear dynamically similar teleoperated manipulators. *IEEE Trans. Robotics and Automation, Vol. 19, No. 3, pp. 443–456*, 2003.

[55] Y. Yokokohji and T. Yoshikawa. Bilateral control of master-slave manipulators for ideal kinesthetic coupling. *Proc. IEEE Conf. R & A, pp. 849–858*, 1992.

[56] B. Hannaford. A design framework for teleoperators with kinesthetic feedback. *IEEE Trans. Robotics & Automation, Vol. 5, No. 4, pp. 426–434*, 1989.

[57] W. Iida and K. Ohnishi. Reproducibility and operationality in bilateral teleoperation. *Proc. 8th IEEE Int. Workshop on Advanced Motion Control, pp. 217–222*, 2004.

[58] G. J. Raju, G. C. Verghese, and T. B. Sheridan. Design issuse in 2-port networ1k models of bilateral remote manipulation. *Proc. IEEE Int. Conf. R & A, pp. 1316–1321*, 1989.

[59] T. Murakami, F. Yu, and K. Ohnishi. Torque sensorless control in multidegree-of-freedom manipulator. *IEEE Trans. Industrial Electronics, Vol. 40, No. 2, pp. 259–265*, 1993.

[60] T. Tsuji and K. Ohnishi. Oscillation control of suspended load with flywheels. *IEEJ Trans. Industry Applications, Vol. 125-D, No. 6, pp. 548–553*, 2005 (in Japanese).

[61] T. Miyazaki and S. Hagihara. Parallel control method for a bilateral master-slave manipulator. *J. RSJ, Vol. 7, No. 5, pp. 46–52*, 1989 (in Japanese).

[62] M. C. L. Sabatucci and A. Chella. A possible approach to the development of robotic multi-agent systems. *Proc. IEEE/WIC Int. Conf. Intelligent Agent Technology, pp. 539–544*, 2003.

[63] Y. Fujimoto and T. Sekiguchi. Fault-tolerant configuration of distributed discrete controllers. *IEEE Trans. Industrial Electronics, Vol. 50, No. 1, pp. 86–93*, 2003.

[64] S. Arimoto, K. Tahara, M. Yamaguchi, P. T. A. Nguyen, and H. Y. Han. Principles of superposition for controlling pinch motions by means of robot fingers with soft tips. *Robotica, Vol. 19, pp. 21 – 28*, 2001.

[65] N. Hayashida, T. Yakoh, T. Murakami, and K. Ohnishi. A friction compensation in twin drive system. *Proc. 6th IEEE Int. Workshop on Advanced Motion Control, pp. 187–192*, 2000.

[66] T. Murakami and K. Ohnishi. A study of stability and workspace decoupling control based on robust control in multi-degrees-of-freedom robot. *IEEJ Trans. Industry Applications, Vol. 113-D, No. 5, pp. 639–646*, 1993 (in Japanese).

[67] T. Kageyama and K. Ohnishi. An architecture of decentralized control for multi-degrees of freedom parallel manipulator. *Proc. 7th Int. Workshop on Advanced Motion Control, pp. 74–79*, 2002.

[68] M. Mizuochi, T. Tsuji, and K. Ohnishi. Multirate sampling method for acceleration control system. *IEEE Intr. Symp. on Industrial Electronics*, 2005.

[69] M. Mizuochi, T. Tsuji, and K. Ohnishi. Force sensing and force control using multirate sampling method. *Proc. IEEE Int. Conf. Industrial Electronics, Control and Instrumentation (IECON'05)*, 2005.

[70] T. Hara and M. Tomizuka. Performance enhancement of multi-rate controller for hard disk drives. *IEEE Trans. on Magnetics, Vol. 35, No. 2, pp. 898–903*, 1999.

[71] H. Fujimoto and Y. Hori. Visual servoing based on intersample disturbance rejection by multirate sampling control. *Proc. IEEE Conf. Decision and Control, pp. 334–339*, 2001.

[72] H. Fujimoto, Y. Hori, and A. Kawamura. Perfect tracking control based on multirate feedforward control with generalized sampling periods. *IEEE Trans. on Industrial Electronics, Vol. 48, No. 3, pp. 636–644*, 2001.

[73] T. Ohmae, T. Matsuda, K. Kamiyama, and M. Tachikawa. A microprocessor-controlled high-accuracy wide-range speed regulator for motor drives. *IEEE Trans. Industrial Electronics, Vol. 29, No. 3, pp. 207–211*, 1982.

[74] Richard C. Kavanagh. Improved digital tachometer with reduced sensitivity to sensor nonideality. *IEEE Trans. Industrial Electronics, Vol. 47, No. 4, pp. 890–897*, 2000.

[75] P. R. Belanger. Estimation of angular velocity and acceleration from shaft encoder measurements. *Proc. IEEE Int. Conf. Robotics & Automation, pp. 585–592*, 1992.

[76] Y. Okamura, Y. Chun, and Y. Hori. Robust and adaptive control of servomotor with low resolution shaft encoder by average speed type instantaneous speed observer. *Proc. of Int. Power Electronics Conf. (IPEC-Yokohama'95), pp. 705–711*, 1995.

[77] R. H. Brown, S. C. Schneider, and M. G. Mulligan. Analysis of algorithms for velocity estimation from dicrete position versus time data. *IEEE Trans. Industrial Electronics, Vol. 39, No. 1, pp. 11–19*, 1992.

[78] S. Komada, N. Machii, and T. Hori. Control of redundant manipulators considering order of disturbance observer. *IEEE Trans. Industrial Electronics, Vol. 47, No. 2, pp. 413–420*, 2000.

[79] S. H. Lee and J. B. Song. Acceleration estimator for low-velocity and low-acceleration regions based on encoder position data. *IEEE/ASME Trans. Mechatronics, Vol. 6, No. 1, pp. 58–64*, 2001.

[80] I. Godler, H. Honda, and K. Ohnishi. Design guidlines for disturbance observer's filter in discrete time. *Proc. 7th IEEE Int. Workshop on Advanced Motion Control, pp. 390–395*, 2002.

[81] Cocoresearch Inc. http://www.cocores.co.jp/english/index.html.

[82] R. Bonert. Design of a high performance digital tachometer with a microcontroller. *IEEE Trans. Instrumentation and Measurement, Vol. 38. No. 6, pp. 1104–1108*, 1989.

[83] T. Tsuji, K. Natori, and K. Ohnishi. A controller design method of bilateral control system. *EPE-PEMC'04, Vol. 4, pp. 123–128*, 2005.

[84] T. Tsuji, M. Mizuochi, H. Nishi, and K. Ohnishi. A velocity measurement method for acceleration control. *The 31st Int. Conf. on Industrial Electronics, Control and Instrumentation(IECON'05)*, 2005.

[85] T. Tsuji and K. Ohnishi. A controller design method of decentralized control system. *Proc. 5th Intr. Power Electronics Conference (IPEC-Niigata 2005)*, 2005.

[86] T. Tsuji, M. Mizuochi, and K. Ohnishi. Modification of measurement time for digital tachometers. *Proc. on Japan Industry Applications Society Conf.*, 2005.

[87] T. Yakoh, H. Sato, and T. Aoyama. Fine real-time processing in distributed systems. *Proc. of IEEE Int. Workshop on Factory Communication Systems, pp. 135–142*, 2000.

[88] Y. Yokokohji and T. Yoshikawa. Bilateral control of master-slave manipulators for ideal kinesthetic coupling-formulation and experiment. *IEEE Trans. Robotics and Automation, Vol. 10, No. 5, pp. 605–620*, 1994.

[89] K. Yoshida. The spacedyn : a matlab toolbox for space and mobile robots. *http://kepler.mech.tohoku.ac.jp/spacedyn/index-e.html*, 2000.

[90] T. Yoshikawa. Foundations of robotics: Analysis and control. *Cambridge, MA: MIT Press*, 1990.

[91] T. Yoshikawa and A. Sudou. Dynamic hybrid position/force control of robot manipulators—on-line estimation of unknown constraint. *IEEE Trans. Robotics and Automation, Vol. 9, No. 2, pp. 220–226*, 1993.

Acknowledgements

I have given myself over to research since 1999, the year I entered Ohnishi laboratory. While conducting my research, many people gave me good advices and prompts. After writing up my summarization of 6 years work, I realized that my paper, my study and I myself are based on relationships between them. Therefore, I want to express my acknowledgements to them here.

First of all, I greatly appreciate Professor Dr. Kouhei Ohnishi in Keio University for his tolerant tutorship. In retrospect, I was a surprisingly clumsy student when I entered this laboratory. Even so, he educated me by capitalizing on my strength, which is only possible since he knew my character. Furthermore, I got many good chances from him to widen my view. I believe that I have to be a high-quality researcher out of gratitude to his tutorship.

Associate Professor Dr. Toshiyuki Murakami in Keio University was so kind to give me comments in detail to complete my study. I deeply thank Professor Dr. Koichiro Sawa, Associate Professor Dr. Takahiro Yakoh and Lecturer Dr. Hiroaki Nishi in Keio University, for giving me good advices during meetings and every other occasion. I am grateful to Professor Dr. Tojiro Aoyama and Associate Professor Dr. Hideo Saito in Keio University with the valuable discussions to polish up this doctoral thesis.

In year 2003, I studied under Professor Dr. Karel Jezernik in Maribor University. The most important attitude I learned from him is that experiments show every truth. Dr. Ales Hace in Maribor University was my mentor and also my best friend at the same time. Dr. Miran Rodic in Maribor University kindly taught me not only skills but also Slovenian life. I greatly appreciate them for the priceless time in Maribor.

I was lucky enough to have an opportunity to discuss a research topic with Professor Dr. Asif Sabanovic in Sabanci University . The idea of functionality was further developed with his mathematically accurate tuition. Professor Dr. Ohishi in Nagaoka University of Technology

always gave me valuable comments at conferences. I am thankful to Professor Dr. Yoichi Hori in University of Tokyo, Associate Professor Dr. Yasutaka Fujimoto in Yokohama National University, Associate Professor Dr. Hiroshi Fujimoto in Yokohama National University, Associate Professor Dr. Satoshi Komada in Mie University, Associate Professor Dr. Masaaki Shibata in Seikei University, and Assistant Professor Dr. Naoki Oda in Chitose Institute of Science and Technology with their advices to my study.

I appreciate teachers and staffs of The 21st Century Center of Excellence(COE) program for Optical and Electronics Device Technology for Access Network which supported my research.

Dr. Mitsuharu Morisawa, now in Advanced Industrial Science and Technology, was my best senior in the laboratory who gave me shrewd comments about my study. I still honor him as an archetype of an engineer. I also feel lucky that I had good colleagues to share my time. 6 years in the laboratory was so precious for me. I often learned from them as well as they learned from me. I especially express my thanks to Ms. Mariko Mizuochi, who was my best research partner.

At the last, I am grateful to my parents: my father who passed away when I was 15 years old; and my mother who brought me up by her lonesome after that. They were both researchers. I know my passion for research is inherited from them. All I know now is that I would never lose the passion.

I have the fortune to have these people around me. Moreover, there are a lot of people I couldn't mention here although I wanted to. I would like to express my great thanks again to all the people who supported my research life in Ohnishi laboratory.

February 2006

Toshiaki Tsuji

List of Achievements

Journals

[1] Toshiaki Tsuji and Kouhei Ohnishi: "A Trajectory Planning of Biped Robot Taking Environment into Account," *IEE Japan Trans. Industry Applications*, Vol. 122–D, No. 11, pp. 1076–1081, 2002. (in Japanese)

[2] Toshiaki Tsuji and Kouhei Ohnishi: "A Global Step Planning Method for Biped Robot Considering Obstacles," *IEE Japan Trans. Industry Applications*, Vol. 124–D, No. 6, pp. 549–555, 2004. (in Japanese)

[3] Toshiaki Tsuji and Kouhei Ohnishi: "Oscillation Control of Suspended Load with Flywheels," *IEE Japan Trans. Industry Applications*, Vol. 125–D, No. 6, pp. 548–553, 2005. (in Japanese)

[4] Toshiaki Tsuji, Kenji Natori, Hiroaki Nishi and Kouhei Ohnishi: "Controller Design Method of Bilateral Control System," *European Power Electronics and Drives Journal*, Vol. 16, 2006. (Accepted)

Journals (as co–author)

[1] Daiki Ito, Takashi Kageyama, Jun Suzuki, Toshiaki Tsuji, Mitsuharu Morisawa and Kouhei Ohnishi: "A Design of Decentralized Control System in Unstructured Environment," *IEE Japan Trans. Industry Applications*, Vol. 123–D, No. 10, pp. 1219–1225, 2003. (in Japanese)

[2] Kenji Natori, Toshiaki Tsuji, Takahiro Yakoh and Kouhei Ohnishi: "Bilateral Teleoperation through Networks," *IEE Japan Trans. Industry Applications*, Vol. 126–D, No. 2, pp. 161–167, 2006. (in Japanese)

[3] Mariko Mizuochi, Toshiaki Tsuji, Hiroaki Nishi and Kouhei Ohnishi: "Realization of Acceleration Control Using Multirate Sampling Method," *IEE Japan Trans. Industry Applications*, Vol. 126–D, No. 3, 2006. (Accepted) (in Japanese)

[4] Eijiro Ohashi, Takahiro Aiko, Toshiaki Tsuji and Kouhei Ohnishi: "Collision Avoidance Method of Humanoid Robot with Arm Force," *IEEE Trans. on Industrial Electronics.* (Accepted)

International Conference

[1] Toshiaki Tsuji and Kouhei Ohnishi: "A Control of Biped Robot which Applies Inverted Pendulum Mode with Virtual Supporting Point," *The 7th Int. Workshop on Advanced Motion Control, AMC'02–Maribor*, pp. 478–483, July, 2002.

[2] Toshiaki Tsuji and Kouhei Ohnishi: "Global Step Planning of Dynamic Biped Locomotion Considering Obstacles," *The 28th Int. Conf. on Industrial Electronics, Control and Instrumentation, IECON'02*, pp. 2445–2450, November, 2002.

[3] Toshiaki Tsuji and Kouhei Ohnishi: "Mechanism of Attitude Control Device for Floating Object," *The 10th IEEE Int. Conf. on Industrial Technology ICIT'03–MARIBOR*, pp. 250–255, December, 2003.

[4] Toshiaki Tsuji, Atsushi Kato, Kouhei Ohnishi, Ales Hace and Karel Jezernik: "Safety Control of Teleoperation System under Time Varying Communication," *The 8th Int. Workshop on Advanced Motion Control, AMC'04–Kawasaki*, pp. 463–468, March, 2004

[5] Toshiaki Tsuji, Kenji Natori and Kouhei Ohnishi: "Controller Design Method of Bilateral Control System," *The 11th Int. Power Electronics and Motion Control Conf., EPE–PEMC '04–RIGA*, Vol. 4, pp. 123–128, September, 2004.

[6] Toshiaki Tsuji and Kouhei Ohnishi: "Position/Force Scaling of Function-based Bilateral Control System," *IEEE Int. Conf. on Industrial Technology, ICIT ' 04–Tunisia*, pp. 96–101, December, 2004

[7] Toshiaki Tsuji and Kouhei Ohnishi: "A Controller Design Method of Decentralized Control System," *IEEJ Int. Power Electronics Conf., IPEC–NIIGATA 2005*, pp. 1378–1385, April, 2005.

[8] Toshiaki Tsuji and Kouhei Ohnishi: "A Design Method of Decentralized Control System Applying System Connection," *IEEE Int. Symp. on Industrial Electronics, ISIE 2005*, pp. 1769–1774, June, 2005.

[9] Toshiaki Tsuji, Mariko Mizuochi, Hiroaki Nishi and Kouhei Ohnishi: "A Velocity Measurement Method for Acceleration Control," *The 31st Int. Conf. on Industrial Electronics, Control and Instrumentation, IECON'05*, pp. 1943–1948, November, 2005.

[10] Muis Abdul, Atsushi Kato, Toshiaki Tsuji and Kouhei Ohnishi: "RTLinux based Real-time Control Cooperation through Internet Protocol," *Mechatronics & Robotics 2004, MechRob 2004*, pp. 117–122, September, 2004.

[11] Kenji Natori, Toshiaki Tsuji, Kouhei Ohnishi, Ales Hace and Karel Jezernik: "Robust Bilateral Control with Internet Communication," *The 30th Annual Conf. of the IEEE Industrial Electronics Society, IECON '04–BUSAN*, November, 2004.

[12] Eijiro Ohashi, Takahiro Aiko, Toshiaki Tsuji and Kouhei Ohnishi: "Collision Avoidance Method of Humanoid Robot with Arm Force," *IEEE Int. Conf. on Industrial Technology, ICIT ' 04–Tunisia*, pp. 1057–1062, December, 2004

[13] Motomi Ikebe, Toshiaki Tsuji, Yoshiharu Sato, Koichiro Tsuji and Kouhei Ohnishi: "Pushing Motion of Humanoid Robot on Dynamic Locomotion," *IEEE Int. Conf. on Industrial Technology, ICIT ' 04–Tunisia*, pp. 90–95, December, 2004

[14] Mariko Mizuochi, Toshiaki Tsuji and Kouhei Ohnishi: "Multirate Sampling Method for Acceleration Control System," *IEEE Int. Symp. on Industrial Electronics, ISIE 2005*, pp. 1629–1634, June, 2005.

[15] Kenji Natori, Seiichiro Katsura, Toshiaki Tsuji and Kouhei Ohnishi: "Analysis of Bilateral Systems with Time Delay," *IEEE Int. Symp. on Industrial Electronics, ISIE 2005*, pp. 1511–1516, June, 2005.

[16] Tomoyuki Suzuki, Toshiaki Tsuji and Kouhei Ohnishi: "Trajectory Planning of Biped Robot for Running Motion," *The 31st Int. Conf. on Industrial Electronics, Control and Instrumentation, IECON'05*, pp. 1815–1820, November, 2005.

[17] Mariko Mizuochi, Toshiaki Tsuji and Kouhei Ohnishi: "Force Sensing and Force Control Using Multirate Sampling Method," *The 31st Int. Conf. on Industrial Electronics, Control and Instrumentation, IECON'05* pp. 1919–1924, November, 2005.

Domestic Conference

[1] Toshiaki Tsuji, Kouhei Ohnishi and Toshiyuki Murakami: "A Hybrid Control of Biped Robot with Virtual Passive Joint," *IEE Japan Industry Applications Society Conference 2001, JIASC 2001.* Vol. 1, pp. 587–592, 2001.

[2] Toshiaki Tsuji and Kouhei Ohnishi: "A Control of Biped Robot with Linear Inverted Pendulum Mode which Applies Virtual Supporting Point," *Technical Meeting on Industrial Instrumentation and Control, IEE Japan*, IIC–02–18, pp. 19–24, 2002.

[3] Toshiaki Tsuji, Atsushi Kato, Kouhei Ohnishi, Ales Hace and Karel Jezernik: "Construction of Bilateral Tele-manipulation System Considering Time Varying Communication Delay," *Technical Meeting on Industrial Instrumentation and Control, IEE Japan*, IIC–04–43, 2004.

[4] Toshiaki Tsuji and Kouhei Ohnishi: "Oscillation Control of Suspended Load with Flywheels," *IEE Japan Industry Applications Society Conference 2004, JIASC 2004.* pp. 465–470, 2004.

[5] Toshiaki Tsuji, Koichi Nishimura and Kouhei Ohnishi: "Performance Assessment of Attitude Control Device with Flywheels," *Technical Meeting on Industrial Instrumentation and Control, IEE Japan*, IIC–05–77, 2005.

[6] Toshiaki Tsuji, Mariko Mizuochi and Kouhei Ohnishi: "Modification of Measurement Time for Digital Tachometers," *IEE Japan Industry Applications Society Conference 2005, JIASC 2005.*

[7] Kenji Natori, Toshiaki Tsuji, Takahiro Yakoh and Kouhei Ohnishi: "Bilateral Teleoperation through Networks, " *Technical Meeting on Industrial Instrumentation and Control, IEE Japan*, IIC–05–23, 2005.

[8] Mariko Mizuochi, Toshiaki Tsuji and Kouhei Ohnishi: "Realization of Acceleration Control Using Multirate Sampling Method, " *Technical Meeting on Industrial Instrumentation and Control, IEE Japan*, IIC–05–9, 2005.

[8] Koichi Nishimura, Toshiaki Tsuji and Kouhei Ohnishi: "Sensorless Oscillation Control of a Suspended Load with Flywheels," *IEE Japan Industry Applications Society Conference 2005, JIASC 2005.*

Others

[1] Toshiaki Tsuji and Kouhei Ohnishi: "Structurization of decentralized control system based on environmental information," Interim RAs Research Progress Report, Vol. 2, pp. 73–76, 2004.

Awards

[1] Gratitude & Appreciation Award, *General Co-chairs of 8th IEEE International Workshop on Advanced Motion Control*, March 26, 2004.

[2] Excellent Presentation Award, *Committee of Industrial Instrumentation and Control, IEE Japan*, December 14, 2004.

[3] Promotion Award for International Conference Publication, *Keio University Graduate School of Science and Technology*, March 17, 2005.

[4] Outstanding Contribution to the AMC 2004, *IEEE Industrial Electronics Society*, April 1, 2005.

[5] Student Forum Paper First Prize, *The IEEE International Symposium on Industrial Electronics (ISIE 2005)*, June 22, 2005.

[6] Excellent Paper Award of IEEJ Industry Application Society, *IEE Japan Industry Application Society*, September 9, 2005.

[7] Excellent Presentation Award, *Committee of Industrial Instrumentation and Control, IEE Japan*, December 21, 2005.

Patent Applications

[1] Toshiaki Tsuji and Kouhei Ohnishi:
"Flywheel Equipment for Attitude Control"
Serial number: 2004–151284, Publication number: 2005-329857
Application date: May 21, 2004
Date of publication of unexamined patent application: December 2, 2005
Applicant: Keio University

[2] Kenji Natori, Toshiaki Tsuji and Kouhei Ohnishi
"Remote-control System Transmitting Signal via Network with Time Delay"
Serial number: 2004–315391
Applicant: Keio University

Career

[1] From April, 2003
COE Research Assistant

Keio University Graduate School The 21st Century COE Program Information, Electrical Engineering, and Electronics, Optical and Electronic Device Technology for Access Network