Master Thesis

Sequential Jumping-Stepping Motion on Bipedal Musculoskeletal Robot for Agile Locomotion

(ジャンプとステップの連続によるヒト型筋骨格ロボットの素早い移動方法に関する研究)

Submitted on July 6th, 2018

Supervisor: Asst. Prof. Ryuma Niiyama

Department of Mechano-Informatics
Graduate School of Information Science and Technology
The University of Tokyo

48-166546 Dwindra Sulistyoutomo

Abstract

Combination of various dynamic motions may lead to better performance of dynamic motion task by utilizing energy transfer between the motions. By doing the motion sequentially, energy transfer may be beneficial for stronger or faster movement on the last motions. Such ability may be a good solution for various task in robotics. However, realization of combination of dynamic motions is hard to implement on robot, due to limitation of actuator and control method.

We have developed a robot capable of doing sequential jumping-stepping motion for more agile performance. The robot was a musculoskeletal humanoid robot which mimics human skeletal system and muscular system with pneumatic articial muscle as the actuator. Pneumatic artificial muscle was chosen as it was able to perform explosive movement such as jumping. For realizing the motion, we proposed control strategy for sequential jumping-stepping motion with muscle control command on a musculoskeletal humanoid robot. Landing command is important for the sequence, as post-motion strategy for the jumping and preparation for stepping.

We have tried our control method on the fabricated robot, and shown that the robot was capable of realizing the jumping-stepping motion in sequence. The results show that performing pre-jumping before a stepping give faster stepping action rather than stepping without pre-jumping. We also found that proper timing of landing command and stepping command became crucial for the motion performance.

Contents

Chapter 1 Introduction 1
1.1 Background 1
1.2 Research Objectives and Methods 2
1.3 Thesis Overview 2

Chapter 2 Agile Locomotion with Musculoskeletal Robot 5
2.1 Artificial Muscle for Humanoid Robot 5
2.2 Pneumatic Artificial Muscle Model 8
2.2.1 McKibben Muscle Tension 9
2.2.2 McKibben Muscle Compliance 10
2.3 Dynamic Motion and Control on Musculoskeletal Robot 10

Chapter 3 Combination of Dynamic Motions with Bio-inspired Muscle Mechanism 13
3.1 Dynamic Motions Sequence 14
3.2 Postural Balance Control with Joint Angle Controller 15
3.3 Dynamic Properties of Musculoskeletal System 16
3.3.1 Force Profile 17
3.3.2 Muscle Stiffness Analysis 17
3.4 Muscle Control Command for Dynamic Motions 19
3.4.1 Jumping Motion 19
3.4.2 Landing Motion 20
3.4.3 Stepping Motion 21
3.5 Sequential Jumping-Stepping Motion Strategy 22

Chapter 4 Robot Design and Fabrication for Realizing Human-like Motion 25
4.1 Robot Overall Specification 25
4.2 Skeletal System 26
4.3 Muscle Configuration 27
4.3.1 Inspiration from Human Muscle 27
4.3.2 McKibben-type Pneumatic Artificial Muscle as Actuator . 31
4.3.3 Muscle Actuator Length Calculation 32
4.3.4 Force Profile Verification 33
4.4 Electrical System 35

4.4.1 Data Flow for The System 35
4.4.2 Embedded Sensor 37
4.5 Pneumatic System 38

Chapter 5 Sequential Dynamic Motions Experiment 41
5.1 Standing Upright with Postural Balance Control 41
5.2 Jumping Performance 47
5.3 Single Stepping Experiment 51
5.4 Sequential Jumping-Stepping 54
5.5 Discussion 60
5.5.1 Effect of Flexion Timing Parameter 60
5.5.2 Stepping without and with Pre-jumping Comparison . . . 62
5.5.3 Faster Thrusting on Jumping-Stepping motion 62
5.5.4 Reproducibility 64

Chapter 6 Conclusion and Future Work 65
6.1 Conclusion 65
6.2 Future Work 66

Acknowledgement 67

References 69

List of Figures

2.1 Musculoskeletal robots implementing different types of artificial muscle actuator 6
2.2 Series Elastic Actuators (SEA) on robot 7
2.3 Nonlinear spring tensioner for tendon-driven muscle 7
2.4 Pneumatic Artificial Muscle(PAM) 7
2.5 Electromagnetic linear actuator (EMLA) 8
2.6 Different types of pneumatic artificial muscle actuator 8
2.7 McKibben muscle geometry parameters for initial and pressurized muscle 9
2.8 Athlete Robot running based on EMG patterns 10
2.9 Muscle activation pettern based on EMG data for running 11
2.10 Walking with muscle synergy hypothesis 11

3.1 Proposed sequence for combination of dynamic motions. Orange circle shows the joints of the robot. The bullet-started red arrow on the robot body shows desired motion direction of the robot. Dashed line show top position of the robot in order to show change of robot position in vertical axis. 14
3.2 Antagonistic muscle configuration on a joint 15
3.3 Block diagram for joint angle controller 16
3.4 Maximum output force (MOF) profile on different muscle configuration 17
3.5 Muscle configuration and human muscle model for stiffness analysis . 18
3.6 Muscle control command for jumping divided into two sets of action in sequence 20
3.7 Landing muscle control command as jumping post-motion strategy and preparation for next dynamic motion 20
3.8 Stepping motion divided into three gait phases for swing leg and two phases for support leg 21
3.9 Proposed strategy for sequential jumping-stepping motion 22

4.1 Acrylic box for valve placement and main processor unit 26
4.2 Robot design in CAD software 27
4.3 Final fabricated bipedal robot 27
4.4 Active muscle for hip flexion motion 28
4.5 Active muscle for hip abduction movement 29

4.6 Active muscle for knee flexion movement 29
4.7 Active muscle for ankle dorsiflexion and eversion movements 30
4.8 Muscle configuration and degrees of freedom of designed robot 30
4.9 McKibben-type Pneumatic Artificial Muscle 31
4.10 Parts for fabricating McKibben muscle 31
4.11 McKibben muscle parts length in detail (in milimeter) 32
4.12 Joint angle definition on the fabricated robot 33
4.13 Joint angle range of the robot 34
4.14 MOF profile of designed robot on squatting position for jumping . . . 34
4.15 Data flow diagram for the robot 35
4.16 ADC Board system diagram 36
4.17 DAC Board diagram 36
4.18 Sensor embedded in robot system 38
4.19 Pneumatic system diagram for muscle actuator 38
4.20 Proportional pressure regulator 39
4.21 Quick exhaust valve 39

5.1 Deadband to prevent oscillation on joint angle controller 42
5.2 Standing upright experiment using proposed postural balance controller 43
5.3 Initial standing upright posture achieved from postural balance controller from front and side view 44
5.4 Joint angle for initial postural balance 45
5.5 IMU data for standing upright task 45
5.6 Muscle pressure on postural balance 46
5.7 SOL force vector (red arrow) on jumping motion for 0.8 MPa (left) and 0.6 MPa (right) 47
5.8 Muscle activation command for jumping motion 48
5.9 Muscle pressure for jumping experiment 48
5.10 IMU data for jumping experiment 49
5.11 Joint angles data for jumping experiment 49
5.12 Jumping experiment recorded with 59.94 frames per second video . . 50
5.13 Muscle activation command for stepping motion 51
5.14 Muscle pressure for stepping experiment 52
5.15 Joint angles for stepping experiment 53
5.16 IMU data for stepping experiment 53
5.17 Stepping experiment recorded with 59.94 frames per second video from side and front view 54
5.18 Muscle activation command for sequential jumping-stepping motion . 55
5.19 Muscle pressure for sequential jumping-stepping experiment 56
5.20 IMU data for sequential jumping-stepping experiment 56
5.21 Joint angle for sequential jumping-stepping experiment 57
5.22 Sequential jumping-stepping experiment recorded with 59.94 frames per second video from side and front view 59
5.23 Different timing for flexion and thrusting 60

5.24 Stepping performance for different muscle activation timing 61
5.25 Jumping time data from 24 jumping experiments 62
5.26 Comparison between stepping without pre-jumping and with pre-jumping) 63
5.27 Stepping performance from five stepping motion experiments and six jumping-stepping motion experiments 63

List of Tables

4.1 Robot Overall Specification 26
4.2 Muscle Length 33
4.3 Joint angle range comparison between robot (geometry calculation) and human 33

5.1 Initial pressure for each muscles 43

Chapter 1

Introduction

1.1 Background

Agile locomotion has always been a big challenge in the robotics field. Even though humanoid robots are designed to mimic how human moves, it is still a difficult task to realize the same human dynamics for a humanoid robot. Differences between human and robot body configuration are one of the issues for this difficulty. Limitation from the robot actuator affects robot performance on doing the dynamic motion. Numbers of studies tried to explore biomechanics in order to get inspiration for tackling this issue in order to give robot similar dynamic motion ability with a human.

One important key from biomechanical studies is that body structure takes an important role in human and animal to perform various dynamic motions. The musculoskeletal system integrates skeletal structure and muscular system to give human this ability. Muscle has fundamental properties of viscoelastic behavior and force-length-velocity relationship. This viscoelastic property gives compliance on muscles so that human can receive impacts from external perturbation and transfer the energy. The force-length-velocity relationship shows capability of muscle on producing instantaneous force. These properties are important to give the ability to perform the dynamic motion, especially agile locomotion [1].

On common electric motor-driven humanoid robots, the actuation system of the robot has differences with the human. Differ from human muscle mechanism, motor actuators directly drive the joint of the robot. Using this actuator, motor-driven humanoid robot is capable of doing joint trajectory planning for stabilization and walking motion [2] [3]. Electric motor-driven robot is also capable of doing dynamic motion based on Zero Moment Point (ZMP) control method. However, it is difficult to perform agile locomotion which involves large impact.

On the other hand, musculoskeletal robot is one specific type of robot which tries to mimic muscle mechanism into its actuation system. Musculoskeletal humanoid robot uses artificial muscle as the actuator with similar property with human muscle. One type of this muscle actuator is Pneumatic Artificial Muscle (PAM) actuator. PAMs benefit compliance property from air compressability. PAMs also have a high power-to-weight ratio and able to produce large force in a short amount of time. Taking advantage of these

properties, musculoskeletal robots are capable of doing dynamic motion like walking [4], running [5], and jumping [6].

So far, the realization of dynamic motion using a musculoskeletal humanoid robot is applicable only for single dynamic motion. Development of musculoskeletal robot capable of doing multi-modal locomotion was done [7], but the locomotions were done as individual locomotion mode. Therefore, there is still space for improvement in musculoskeletal humanoid robot locomotion study.

Human doing a various combination of different modes of dynamic motion, for example running then jumping, especially in doing sports. Many movements and techniques in sports involve a combination of dynamic motions. One example of techniques that involves combination is split step in tennis. Split step is done by doing a small jump or hop then stepping. This action is done as a preparatory motion to increase stepping speed on receiving ball [8] [9]. From the results, it was shown that the combination of dynamic motions results in more agile locomotion.

1.2 Research Objectives and Methods

The objective of this research is to realize agile locomotion on a musculoskeletal humanoid robot by combining two dynamic motions continuously. The two dynamic motions are jumping and stepping, inspired by the split-step technique in tennis. The realization is done in the real environment using a real fabricated musculoskeletal humanoid robot. Chosen dynamic motions are motions for a legged-system robot so that the fabricated robot is limited to an only active lower-body humanoid robot. Upper-body only acts as a passive component with most of the weight of the robot.

For this goal, strategy for realizing the sequence of dynamic motion is proposed. The control method for each motion on the sequence is independent at first. Each control method is based on muscle mechanism inspired by human motion. Proposed strategy aims to combine the motion by concerning in transition between the motions. For a jumping-stepping combination, landing strategy is proposed as a post-motion strategy for jumping motion and preparation for stepping motion.

1.3 Thesis Overview

The thesis is divided into six chapters. The first chapter introduces the background, objective, and method of the research. Overview of this thesis is also explained in this chapter.

The second chapter reviews musculoskeletal robot and how we can make the robot do agile locomotion. We will discuss artificial muscle used as an actuator and have a comparison between various types of artificial muscle actuator. Model and control of the actuator for realizing agile locomotion will also be explained in this chapter.

The third chapter explains our method for achieving our research objective. The method will be divided into several sections, based on our strategy for realizing the combination of dynamic motions. Each dynamic motion will be explained in different

sections.

The fourth chapter explains how we designed and fabricated real bipedal musculoskeletal robot to realize the dynamic motions on real environment. In this chapter, we describe each system of the robot including the mechanical system, electrical system, and pneumatic system. We also explain the muscle actuator configured for the robot in detail.

The fifth chapter discusses our experiments and results for the combination of dynamic motions. We will discuss the experiments consecutively from first motion to the next motion until the last dynamic motions in our sequence of dynamic motions. The experiments in this chapter are conducted based on the method that explained previously in the third chapter.

Finally, in the sixth chapter, we state our conclusion for the research and discuss the future direction of the research.

Chapter 2

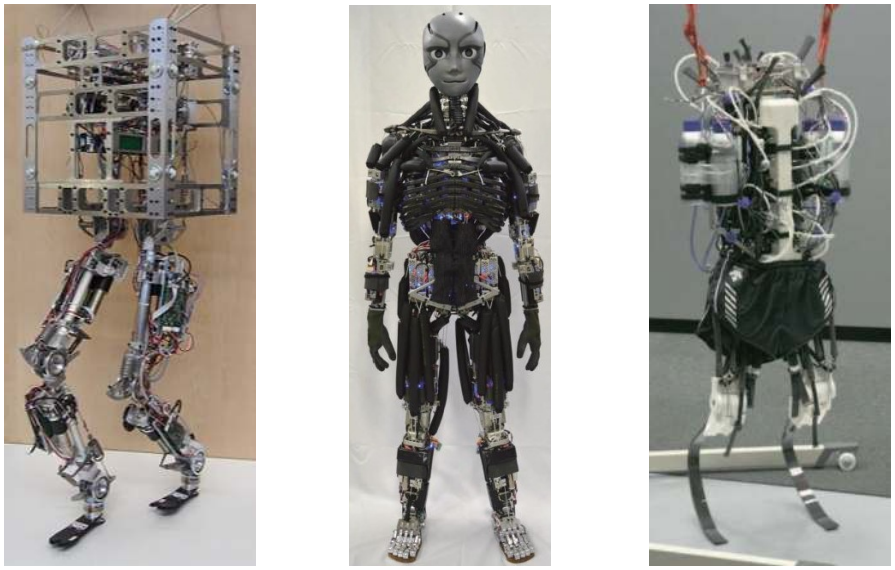
Agile Locomotion with Musculoskeletal Robot

Musculoskeletal robot adopts the skeletal system of human as robot structure and muscular system as robot actuation system. The main difference with other types of robot is muscle actuation system which tries to mimic the characteristics of human muscle. There are two different objectives for implementing this musculoskeletal system on a humanoid robot: a better understanding of human physiology using an anatomical humanoid robot [10] and improving robot performance by exploiting musculoskeletal feature [11]. In order to achieve these objectives, different types of actuator were developed as muscle actuator. Characteristics of each actuator differ from the others, resulting in different application purpose when implemented on robots. Some of the muscle actuators give the ability for the robot to perform a dynamic motion, even an agile locomotion. While the other one gives feasibility to build a more detailed anatomical robot.

In this chapter, we will discuss various types of muscle actuator on a musculoskeletal robot in the first section. The difference between each muscle actuator will be discussed, including examples of robots which implemented the actuator. Actuators are not sorted by first invented order. From the comparison, we will specifically elaborate more about pneumatic artificial muscle actuator. Lastly, we will discuss dynamic motions that have been done by the musculoskeletal humanoid robot.

2.1 Artificial Muscle for Humanoid Robot

For adopting a musculoskeletal system into a robot, actuator becomes the main issue for the robot design. The main feature of human muscle which plays important roles in realizing various dynamic motion is the muscle compliance. Several methods were done in order to mimic this muscle compliance on the robot as artificial muscle actuator.



(a) BioBiped3 with Series Elastic Actuators (SEA) [12]

(b) Tendon-driven robot Kengoro [10]

(c) Athlete Robot with Pneumatic Artifical Muscle (PAM) [5]

Fig.2.1: Musculoskeletal robots implementing different types of artificial muscle actuator

The first type of actuator designed for having muscle compliance is Series Elastic Actuators (SEA) [13]. Structure of SEA is simple and easy to implement, where the configuration consists of only motor and spring. With this structure, SEA has compliance and energy storage possibility from the spring. One disadvantage of SEA is the compliance is not changeable due to fixed spring constant. Robots in BioBiped project, shown in Fig. 2.1(a) implemented this type of actuator as muscle actuator [11] [12]. The SEA configuration of BioBiped3 is shown in Fig. 2.2. BioBiped1 had shown the advantage of using SEA from its compliance on passive rebound behavior in terms of energy restoration from receiving impact [14]. However, one drawback from SEA is that the compliance is not adjustable due to the usage of fixed spring. Produced torque also depends on the motor actuator which difficult to large power in a limited size.

Another type of actuator for artificial muscle is motor-driven muscle actuator for tendon-driven robots [10] [15] [16] [17] [18]. This actuator employs similar configuration with SEA but with a different arrangement for the spring. Motor-driven muscle uses nonlinear spring tension mechanism [19] for its compliance property. The robots implemented with tendon-driven muscle had shown the capability of performing using redundant muscles. Since the actuator is small, it is possible to use this actuator to realize anatomical musculoskeletal robot with a lot of muscles. Using motor-driven muscle actuators, Kengoro was shown able to do push up, roll over, and stepping motion [10]. However, motor-driven muscle actuator does not have much power to do agile locomotion which needs instantaneous force because of limitation of the motor. Fig. 2.1(b) shows the complicated Kengoro with 116 muscles on a whole-body musculoskeletal robot.

Aside from using the electric motor, Pneumatic Actuator Muscle (PAM) uses air pres-

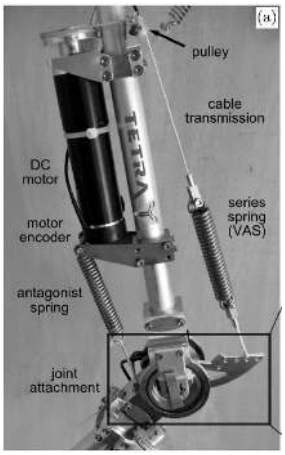


Fig.2.2: Series Elastic Actuators (SEA) on robot [11]

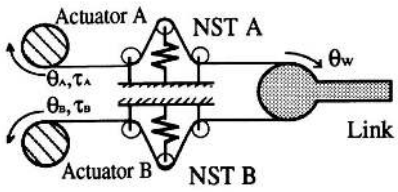


Fig.2.3: Nonlinear spring tensioner for tendon-driven muscle [19]

sured tube as a mechanism to mimic how human muscle contracting and relaxing [20]. PAM has features of high power-to-weight ratio and passive compliance behavior from its air compressibility [21]. With this, PAM is good to mimic muscle compliance characteristics. Having this features, PAM based musculoskeletal robots were shown capabale of doing agile locomotion, such as walking [4] [22], bouncing [23], jumping [6], and running [5]. With simplified locomotion dimension only to two dimensions, even all three locomotion was able to be done independently [7]. Even so, PAM has a drawback of slow response caused by air compressibility, resulting in PAM has weak precise and fast controllability. Fig. 2.4 shows one example of PAM configuration and its application.

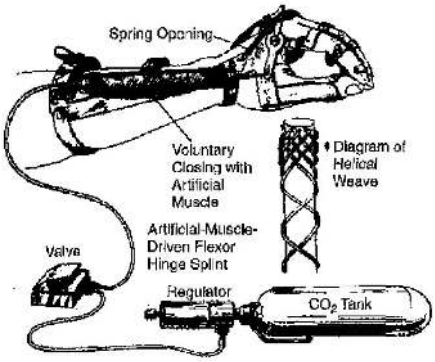


Fig.2.4: Pneumatic Artificial Muscle(PAM) [20]

Electromagnetic linear actuator (EMLA) which uses permanent magnet linear actuator was also developed to have compliance like muscle [24]. The configuration of EMLA is shown in Fig. 2.5. EMLA tries to resolve the drawback of PAM for its slow response for changing stiffness during robot motion. Using EMLA, monopedal robot was able to do 11 hopping steps by changing the stiffness dynamically [25]. However, EMLA cannot be used for maintaining posture since the coils will continuously be excited and produce heat. Therefore EMLA is not suitable for a humanoid robot.

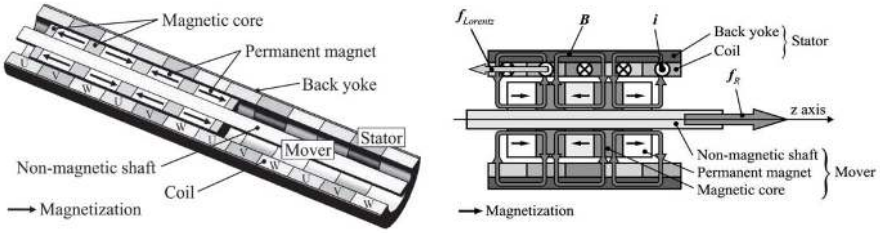
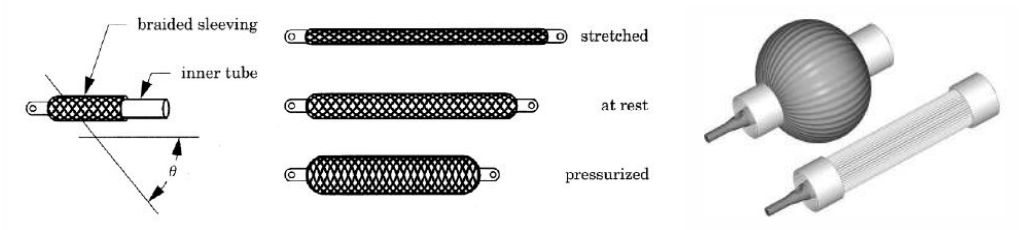


Fig.2.5: Electromagnetic linear actuator (EMLA) [24]

From the comparison, we can infer that SEA and motor-driven muscle have good compliance and controllability, but lack of ability in producing large force. PAM is capable of producing large force and has good compliance, but lack of precise controllability. EMLA is also capable of producing large force, but can not be used for static posture for its drawback. Therefore, in order to realize agile locomotion ability on a humanoid musculoskeletal robot, PAM actuator is the suitable choice.

2.2 Pneumatic Artificial Muscle Model

Pneumatic artificial muscle (PAM) uses air pressure to act as actuator. Therefore, lightweight becomes one of major attraction from this actuator. Exploiting its material structure and feature, the basic mechanism for the PAM is by contracting and reducing its length giving change in linearly by default. However, PAM has passive compliant behavior due to air compressibility. This leads to another attraction of this actuator, where interaction with environment become soft and safe from its compliant behavior.



(a) Braided muscle, McKibben muscle **(b)** Pleated Muscle

Fig.2.6: Different types of pneumatic artificial muscle actuator [26]

There are various types of developed PAMs. Two most known types are braided muscle and pleated muscle shown in Fig. 2.6 [26]. Braided muscle, mostly known as McKibben muscle, is composed of an inner tube covered with a braided sleeve. Braided fibers limit the increased volume of the inner tube and change the direction of air expansion inside to fit braided sleeve shape. Braided fibers run helically along the muscle with an angle θ. Braided angle is changed as muscle length changed when pressurized and stretch from the initial angle at rest position. Dry friction occurs at the operation, making braided muscle has hysteresis on its operation and resulting in difficult to achieve accurate position control.

Pleated muscle is composed of a membrane which has numbers of pleats in the axial direction. When pressurized the pleat is unfolded, making the shape bulky. No friction happens in this process, so that pleated muscle has no hysteresis on its operation. However, bulky form when it is pressurized may become a limitation for installation.

2.2.1 McKibben Muscle Tension

A model that relates the actuator parameters into produced force is important for better usage of the actuator. McKibben changes its shape as air pressure and tension is produced from this mechanism. Geometry parameters of McKibben is shown in Fig. 2.7.

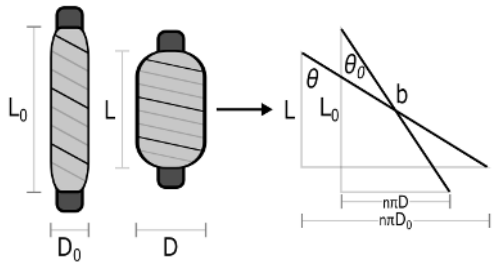


Fig.2.7: McKibben muscle geometry parameters for initial and pressurized muscle

The relationship between output force and geometry parameters is calculated as below [27].

$$F = \frac{\pi D_0^2 P}{4 \sin^2 \theta_0}(3 \cos^2 \theta - 1) \tag{2.1}$$

where

F	= output force[N]	P	= pressure [Pa]
D_0	= initial McKibben diameter [m]		
θ	= braided thread angle	θ_0	= initial braided thread angle
L	= McKibben length	L_0	= initial McKibben length

The equation is then simplified for assumption that for diameter D_0, the angle is 90°.

$$F = \frac{\pi D_0^2}{4} P(\alpha(1 - \epsilon)^2 - \beta) \tag{2.2}$$

with $\alpha = 3/\tan^2\theta_0$, $\beta = 1/\sin^2\theta_0$, and $\epsilon = (L_0 - L)/L_0$

2.2.2 McKibben Muscle Compliance

Another important property of PAM actuator is its compliance that affects the joint compliance of the structure. Muscle stiffness of McKibben muscle can be determined as the derivative of produced force about muscle contraction [26]. Then, muscle compliance can be determined as the inverse of muscle stiffness. From Eq. 2.2, it can be derived that

$$C^{-1} = K = \frac{\delta F}{\delta(\Delta L)} = \frac{6\pi}{\cot^2\theta_0}\frac{D_0^2}{L_0^2}PL \tag{2.3}$$

where C = muscle compliance and K = muscle stiffness.

2.3 Dynamic Motion and Control on Musculoskeletal Robot

The musculoskeletal system gives a different approach on robot control method for performing dynamic tasks. As the actuators mimic human muscular system, most of the control method on musculoskeletal also get inspiration on how human control its muscle. From a biomechanical approach, artificial muscle actuator can be actuated to perform as muscle activity pattern of a human. Using electromyography (EMG) data, human muscle activity pattern on a specific task or motion can be known. By referring to this data, a musculoskeletal robot can be controlled to perform a dynamic motion similar to human, even for an agile locomotion.

Athlete Robot used EMG data from human performing running task to mimic the muscle activation from human muscle to robot actuator [28]. Recorder EMG data was divided into three synergy phases: recovery swing phase; descent foot phase; and thrust phase. Based on these phasic synergies, pressure command and duration was adjusted by exploration in simulation on each phase.

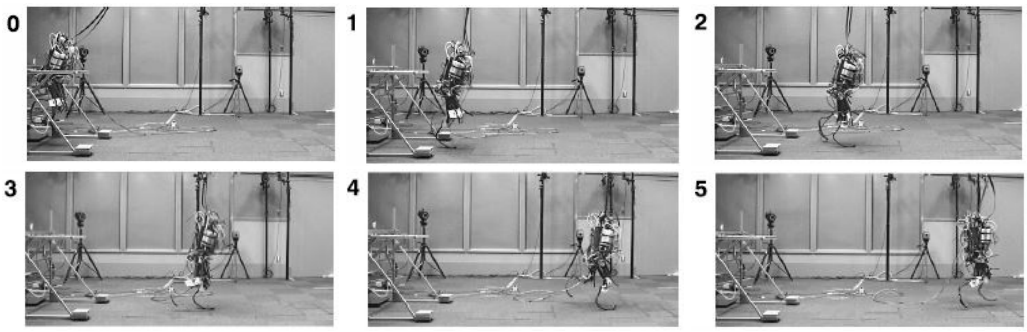
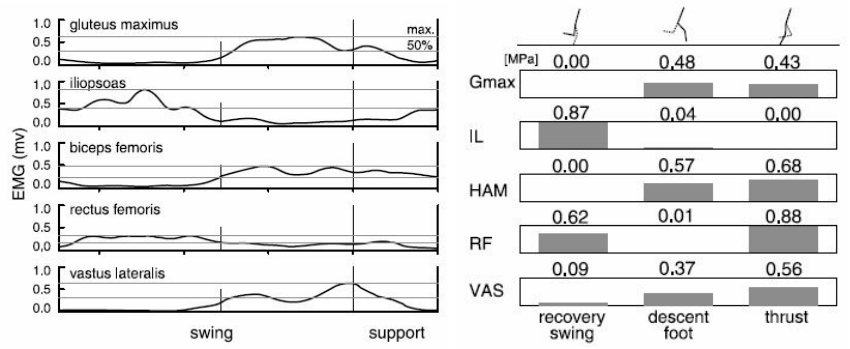


Fig.2.8: Athlete Robot running based on EMG patterns [28]



(a) EMG data for running task on human **(b)** Muscle activation pattern on robot

Fig.2.9: Muscle activation pettern based on EMG data for running [28]

Rather than a direct translation from each human muscle EMG, another control approach involving muscle synergies concept was also proposed. Muscle synergy is coordination on a group of muscles that work for a specific task in synchrony with weighted activation pattern [29] [30] [31]. Muscle synergies for a task are extracted based on experimental data based on human EMG data. These synergies can be used to reconstruct muscle activation pattern for each muscle. Muscle synergies method was shown to be working on musculoskeletal robots which imitate human lower limb completely. The task were walking [32], running [33], and pedalling [34]. However, the experiments were done with robot fixed on a stable frame, since the objectives were only to reproduce similar motions with the human.

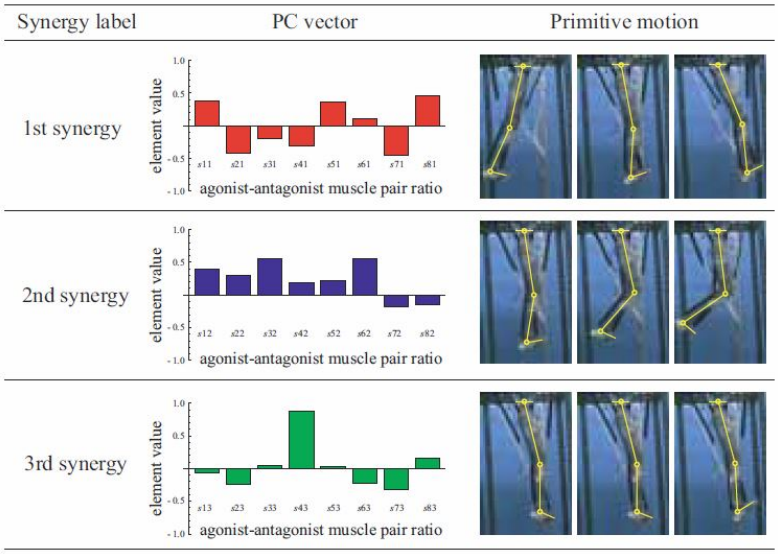


Fig.2.10: Walking with muscle synergy hypothesis [32]

Chapter 3

Combination of Dynamic Motions with Bio-inspired Muscle Mechanism

The musculoskeletal humanoid robot was already shown to have the capability of realizing dynamic motions. Using a pneumatic artificial muscle actuator, agile locomotion was done by exploiting its high power feature. However, there are still problems for this dynamic motion on the musculoskeletal humanoid robot. One of the problems is the robot was still hard to maintain its stability after the dynamic motion. One attempt to handle this problem was using postural control based on PID controller on each joint of the robot [21]. But the performance still needs to be improved for better stability. A whole-body musculoskeletal robot was also developed with standing control strategy based on ankle strategy from human standing strategy [4]. The standing control was shown to be robust on standing task from disturbance on the sagittal plane. However, this strategy still had not answered how to get stability after dynamic motion. There is also a challenge for dynamic motion on a musculoskeletal humanoid robot as a result of the problem stated before. Continuation of the dynamic motion, either by stopping into statically stable position or changing into different dynamic motion, becomes one big challenge to solve.

In this chapter, we will explain about our proposed method for realizing the combination of dynamic motions. The sequence will be a combination of two dynamic motions: jumping and stepping. We also proposed postural balance control to get initial posture for the robot before performing the dynamic motions. For the dynamic motions, the control method was based on human muscle activity pattern, since the robot has a similar musculoskeletal feature with a human. Each method will be discussed in separate sections within this chapter.

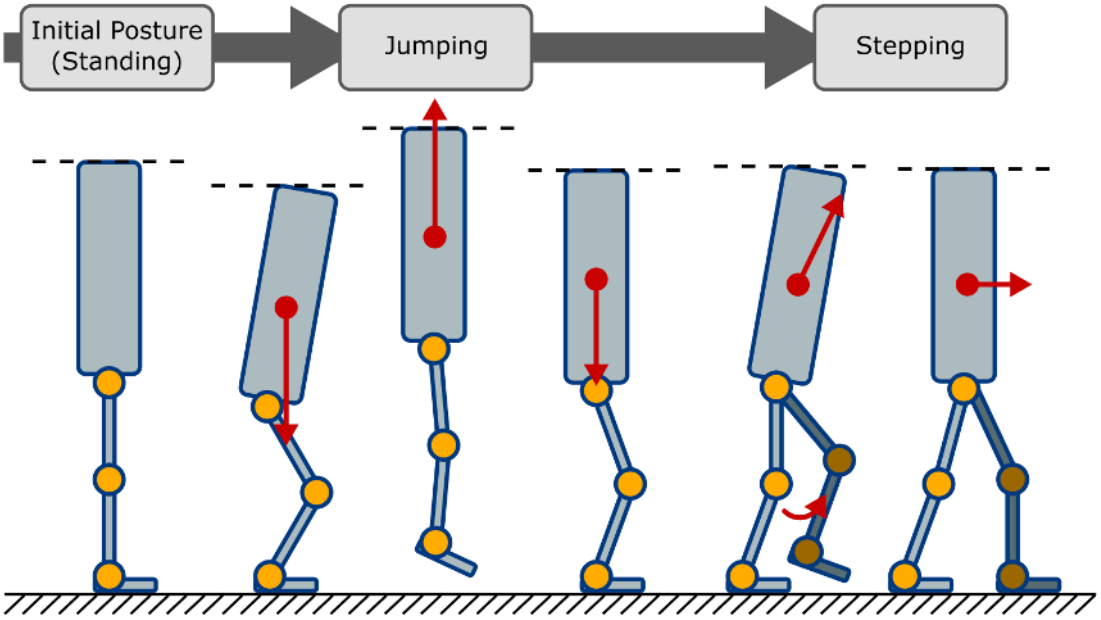


Fig.3.1: Proposed sequence for combination of dynamic motions. Orange circle shows the joints of the robot. The bullet-started red arrow on the robot body shows desired motion direction of the robot. Dashed line show top position of the robot in order to show change of robot position in vertical axis.

3.1 Dynamic Motions Sequence

In this research, we aimed for realizing robot ability to perform additional dynamic motion as a continuation of one dynamic motion. Therefore, the proposed combination of dynamic motions was intended to put two basic dynamic motions together sequentially. We chose jumping motion and stepping motion as the basic dynamic motions that we wanted to combine. This combination was inspired by a similar motion done by human in several sports, like split-step on tennis in which the player hop a little and then step before receiving. The sequence for the combination of dynamic motions is shown in Fig. 3.1.

Before performing the dynamic motions, we decided to make the robot start from a stable position first. We started with standing upright posture as initial posture. For this standing task, the robot performed postural control using joint angle control on every joint. We set the angle set point as zero degree for all joints to achieve upright posture.

From the initial posture, the robot performed the dynamic motion starting with a jumping motion. The sequence for the jumping motion was initial standing, squatting, and jumping. We made the robot moved from initial standing position to a squatting position in order to gain energy. Then, accumulated energy from squatting was transferred to the jumping motion.

Next, we tried to make the robot performing stepping motion right after the landing of jumping. The sequence for stepping after jumping was landing, swinging leg, and step-

ping. During landing, the robot exploited the compliance feature of its body to transfer energy from falling after jumping energy into stepping. Pre-swing preparation also done during landing position. Then, swinging leg action was done and ended with stepping into a new position for one leg.

For limitation, we focused only on continuing one dynamic motion with another dynamic motion. So we stopped the sequence of the dynamic motions with stepping motion, without concerning the next state after the step.

3.2 Postural Balance Control with Joint Angle Controller

Postural balance control was proposed based on the joint angle controller. Joint angle controller was implemented to move the actuators in order to get desired joint angle set points. This method is widely used in the most robotic area as the primitive controller for inverse kinematics problems.

As we planned to use a musculoskeletal humanoid robot, the joints of the robot were not driven directly by a motor as in common motor-driven robots. Instead, musculoskeletal robot adopted antagonistic muscle configuration similar to muscle configuration on most joints of a human. The joint in the human body is moved by two muscles work as agonist and antagonistic muscle opposing each other. Contraction difference between these muscles changes the joint angle.

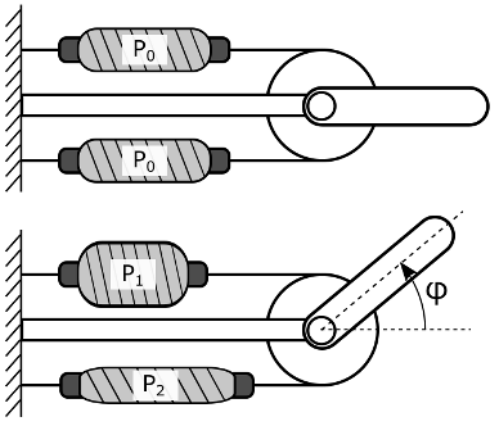


Fig.3.2: Antagonistic muscle configuration on a joint

In the musculoskeletal robot, we used pressured muscle actuators instead of contracting muscles in antagonistic muscle configuration [20]. The configuration is shown in Fig. 3.2. Joint was modeled as a pulley, connected into two antagonistic muscle actuators. Ideally, with the same pressure on both muscles, the joint angle would be zero degrees. If pressure between the two muscles were differentiated, the joint angle would be changed. The joint would be moved toward muscle with higher pressure.

In order to make the joint capable of moving in both clockwise and counterclockwise direction, initial pressure $P_0 \neq 0$ was used. Then, pressure on one muscle can be in-

creased or decreased from this initial pressure as if muscle contracting and relaxing. We can set the same muscle pressure change on both muscles so that

$$\begin{aligned} P_1 &= P_0 + \Delta P \\ P_2 &= P_0 - \Delta P \end{aligned} \tag{3.1}$$

This equation was a simple pressure command for giving better controllability to the system.

Joint angle controller for muscle actuator-based system was mostly based on this antagonistic muscle configuration. Fig. 3.3 shows a generalized block diagram for a joint angle controller based on antagonistic muscle configuration. Controller system consisted of a controller, muscles, and joint dynamics. Muscle block represented a pneumatic muscle actuator, which had an input of pressure and output force produced by the muscle. Joint dynamics block represented the dynamics of a joint which was controlled by force from the muscles and other dynamics component like the mass of the robot and resulted in the actual joint angle. The system was a feedback control system, where the feedback signal was directly fed from the actual joint angle. The error between the actual and set point joint angle was fed into the controller block which produced pressure difference ΔP as the control signal. This control signal was used for pressure input for each muscle based on Eq.3.1. Before the pressure input was fed into muscle actuator, saturation for this signal was needed to consider limitation from air pressure supply. For this generalized block diagram, controller block can be replaced with any kind of controller type: bang-bang controller, PID controller, or other types of controller. Some musculoskeletal humanoid robots were developed using the same control scheme for their postural balance control, using P controller and PI controller [4] [22].

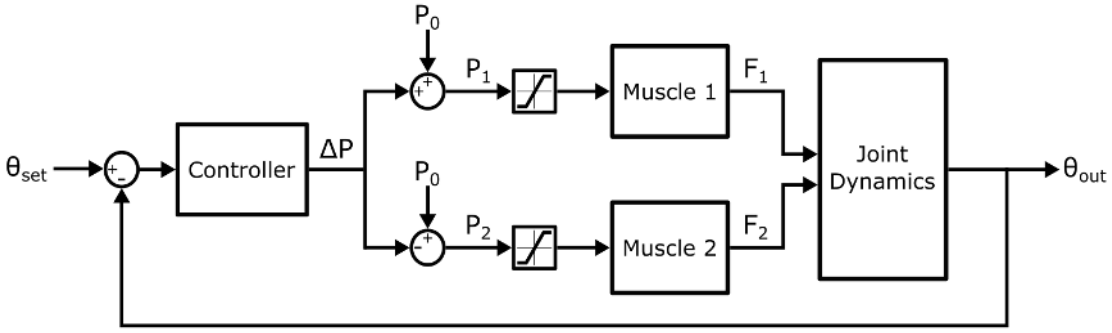


Fig.3.3: Block diagram for joint angle controller

3.3 Dynamic Properties of Musculoskeletal System

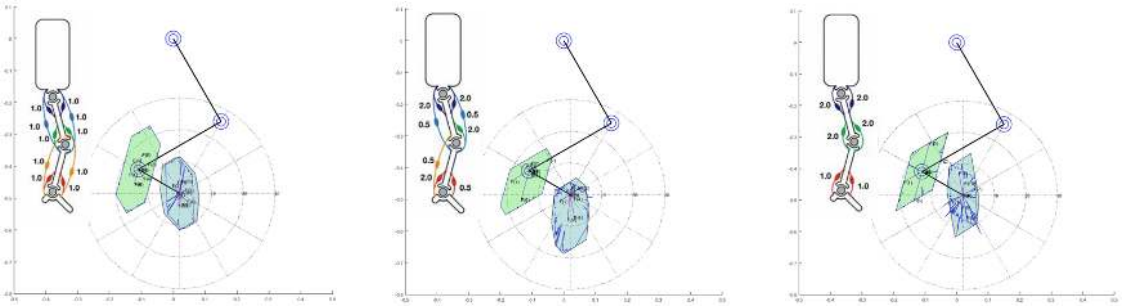
Since musculoskeletal robot had different properties with the common motor-driven robot, further analysis for its dynamic properties was needed. Muscle force and stiffness analysis will be discussed in this section. We tried to model output force produced

by the muscle configuration in order to see the performance on the system. Muscle stiffness was also modeled in order to give the capability of exploiting the compliance of the musculoskeletal system.

3.3.1 Force Profile

Force profile for the musculoskeletal system was done to analyze the output force from each muscle in the configuration and the total force combined on the system. Output force distribution on the musculoskeletal system was first analyzed based on human limb muscles analysis [35]. The human limb model was only consisted of two-joint link arm configuration, with monoarticular muscles and biarticular muscles configuration. This model showed the contribution of each muscle to the output force distribution convex.

Based on this model, a more general method for multiple-joint link arm configuration was proposed as a design framework for musculoskeletal robot [36]. Maximum output force (MOF) profile represented the distribution of the maximum output force as muscle force vector resultant. Using this framework, the force profile for a specific muscle configuration on the musculoskeletal leg can be known as shown in Fig. 3.4. The force profile was used to show force distribution on heel and toe as end-effector of the structure.



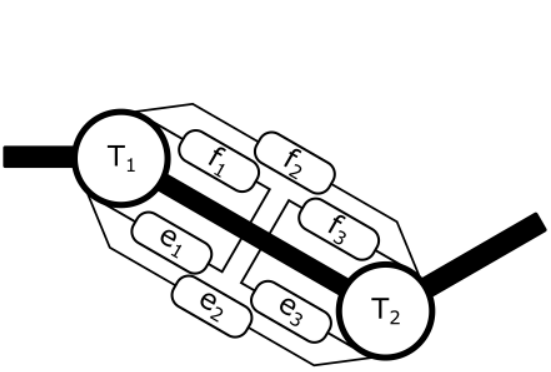
(a) Uniform configuration **(b)** Anthropomorphic configuration **(c)** Symmetric mono-articular configuration

Fig.3.4: Maximum output force (MOF) profile on different muscle configuration [36]

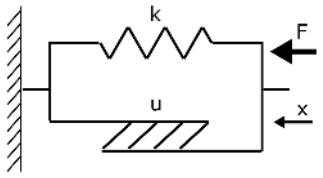
3.3.2 Muscle Stiffness Analysis

Similar to force profile analysis, muscle stiffness analysis done based on human muscle model [37]. The model used two-joint link arm model with antagonistic muscle configuration for both monoarticular and biarticular muscle pairs as shown in Fig 3.5(a). The antagonistic muscle pair consisted of joint flexor and extensor. We denoted muscle pairs starting from the first joint: monoarticular pair f_1 and e_1, biarticular pair f_2 and e_2, and monoarticular f_3 and e_3.

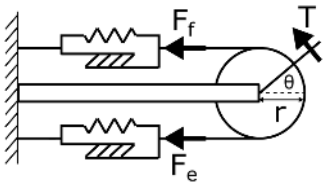
The human muscle was modeled as visco-elastic muscle model in isometric condition with zero viscosity coefficient for simplicity. Fig 3.5(b) shows muscle model with elastic



(a) Two-joint link arm model



(b) Human muscle model



(c) Antagonistic configuration using human muscle model

Fig.3.5: Muscle configuration and human muscle model for stiffness analysis

coefficient k and contraction force u. Model for the human muscle was

$$F = u - kux \tag{3.2}$$

Using the antagonistic muscle configuration, torque worked on a joint was

$$\begin{aligned} T &= (F_f - F_e)r \\ T &= (u_f - u_e)r - (u_f + u_e)kr^2\theta \end{aligned} \tag{3.3}$$

From the torque working on a joint, we could derive joint compliance C or joint stiffness K caused by a muscle pair q.

$$K_q = \frac{\delta T}{\delta \theta} = -(u_f + u_e)kr^2 \tag{3.4}$$

$$C_q = K_q^{-1} = \frac{-1}{(u_f + u_e)kr^2} \tag{3.5}$$

Stiffness matrix can be derived using Jacobian for configuration of only monoarticular and with biarticular muscle pair.

$$K_{q_{mono}} = \begin{bmatrix} K_{q_1} & 0 \\ 0 & K_{q_3} \end{bmatrix} \tag{3.6}$$

$$K_{q_{bi}} = \begin{bmatrix} K_{q_1} + K_{q_2} & K_{q_2} \\ K_{q_2} & K_{q_2} + K_{q_3} \end{bmatrix} \tag{3.7}$$

Stiffness matrix shown above consisted of stiffness on pairs of muscle. This stiffness matrix cannot describe stiffness property on configuration with missing muscle on one

pair of muscle, for example, only flexor or extensor exist. For general muscle configuration, it was important to derive the stiffness matrix into a form of each muscle stiffness. For muscle configuration as shown in Fig 3.5(a), joint stiffness matrix is

$$K_q = \begin{bmatrix} K_{f_1} + K_{e_1} + K_{f_2} + K_{e_2} & K_{f_2} + K_{e_2} \\ K_{f_2} + K_{e_2} & K_{f_2} + K_{e_2} + K_{f_3} + K_{e_3} \end{bmatrix} \tag{3.8}$$

With this generalized joint stiffness matrix, we could determine the joint stiffness based on each muscle stiffness. For pneumatic muscle actuator, muscle stiffness can be determined using Eq. 2.3.

3.4 Muscle Control Command for Dynamic Motions

For dynamic motion, we implemented muscle control command based on human muscle activation pattern. Since the musculoskeletal humanoid robot was inspired by human muscle configuration for its actuator system, we could also get some inspiration on how human control its body using muscular system. For jumping and stepping motion, we tried to implement muscle command based on muscle activation from human doing this respective motion.

3.4.1 Jumping Motion

Muscle control command for jumping was based on optimal human jumping model [38]. The model was limited to sagittal plane only. Jumping was done with squatting as initial posture, involving extensor muscles on hip, knee, and ankle joints. Control method was done only based on a bang-bang controller, with control values of zero and maximum muscle activation. For optimal performance, the activation of the extensors needs to be sequenced from hip extensors, knee extensors, to ankle plantar flexors. An important component of the control method was timing for the sequence of hip, knee, and ankle extensors.

Similar control command was implemented previously on jumping musculoskeletal robot [6]. The control command was also based on the bang-bang controller, using shut-off valves to get McKibben actuator activated with only zero or maximum pressure. Jumping motion was generated with two phases: hip and knee extension; and an addition of ankle plantar flexion. The time duration for the first phase was varied during the experiments. The motion was started with squatting by making the robot fall without any muscle pressure. Jumping motion was successfully done with this muscle control command from squatting position.

We tried to implement a similar muscle control command to realize jumping motion. The motion was divided into two sets of sequence action. The first action was activation of hip extension and knee extension. Duration of this Active 1 was specified as t_{act1}. Second, we continued to activate the hip and knee extension, with an addition of activation of ankle plantar flexion. Active 2 was done during duration of t_{act2}. Fig. 3.6 shows the muscle control command designed for jumping motion.

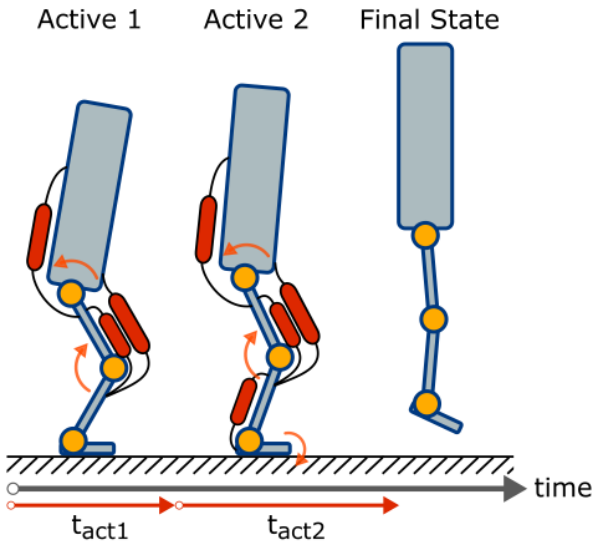


Fig.3.6: Muscle control command for jumping divided into two sets of action in sequence

3.4.2 Landing Motion

Jumping control command previously discussed was concerning only jumping, without concerning about after jumping. In order to realize sequential dynamic motion after jumping, a post-motion strategy was needed as preparation for next dynamic motions. Landing command was needed as the post-motion strategy for jumping motion. The final objective of this strategy was to make the robot land with foot and leg posture capable of doing the next motion.

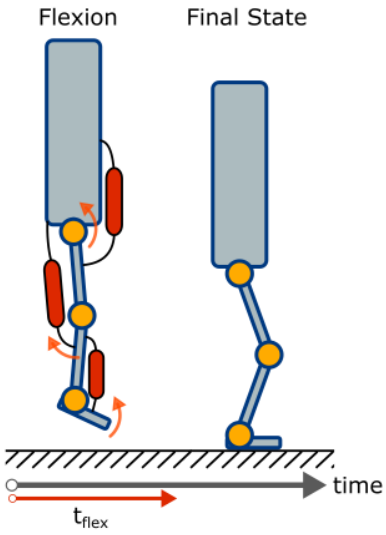


Fig.3.7: Landing muscle control command as jumping post-motion strategy and preparation for next dynamic motion

Another human jumping model was optimized for jumping motion which included landing on its task [39]. Opposite from jumping motion, landing involved the flexor muscles of leg joints. Final state from jumping was extended legs, which was stiff and not good for receiving and transferring the energy from jumping. Therefore, landing control command was hip flexion, knee flexion, and ankle dorsiflexion. The objective of the landing command was reducing leg stiffness by flexed knee and positioning feet for landing. The muscle control command for landing is shown in Fig. 3.7

3.4.3 Stepping Motion

Stepping motion was a part of walking motion, limited as only a half of walking gait cycle. Stepping was defined as one swing of one leg from touching the ground until touching the ground again, while the other leg served as support leg giving thrust for the body. In order to investigate muscle control command for stepping motion, neuro-muscular model-based approaches were done [40] [41]. The models hypothesized motor control needed to realize human walking, limited only to the sagittal plane. We proposed control command for a single stepping based on muscle activity in the models.

Stepping motion differentiated muscle command for right leg and left leg. One leg should act as swing leg and the other as support leg. We divided the stepping motion based on swing leg movement. Stepping motion was divided into three gait phases: Gait 1 as initial swing, Gait 2 as mid-swing, and Gait 3 as terminal swing. Support leg only had 2 phases which depended on the swing leg phase. Gait 1 was initial swing, where the swing leg was first lifted from the ground to give foot clearance. Hip flexion was taken place on swing leg for lifting. Knee flexion also happened due to gravity when lifting the leg. As the Gait 1 start, support leg got into Prethrust phase, where support leg bends the knee caused by knee flexion and lifted swing leg. After Gait 1 and Prethrust was

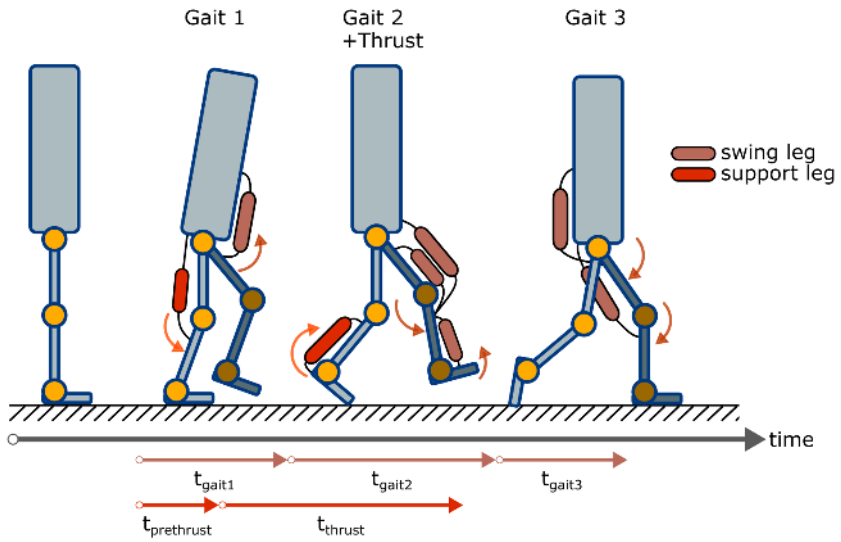


Fig.3.8: Stepping motion divided into three gait phases for swing leg and two phases for support leg

done, the robot body would be inclined forward due to the shift of the center of mass. Before swing leg change into Gait 2, support leg got into Thrust phase to move the body forward. Thrusting was done by ankle plantar flexor. Then Gait 2 extended the knee and dorsiflexed the ankle while swing leg was in the air. Last, Gait 3 activated knee flexor to decelerate and hip extensor to swing leg and put the leg into the ground. Each gait was done during t_{gait1}, t_{gait2}, and t_{gait3}.

3.5 Sequential Jumping-Stepping Motion Strategy

In previous subsections, muscle control commands for each independent motion were already discussed. However, in order to achieve the objective of this study, integrated muscle control command for realizing sequential jumping-stepping method was needed. We proposed the muscle control command for sequential jumping-stepping which was based on the independent motion control command.

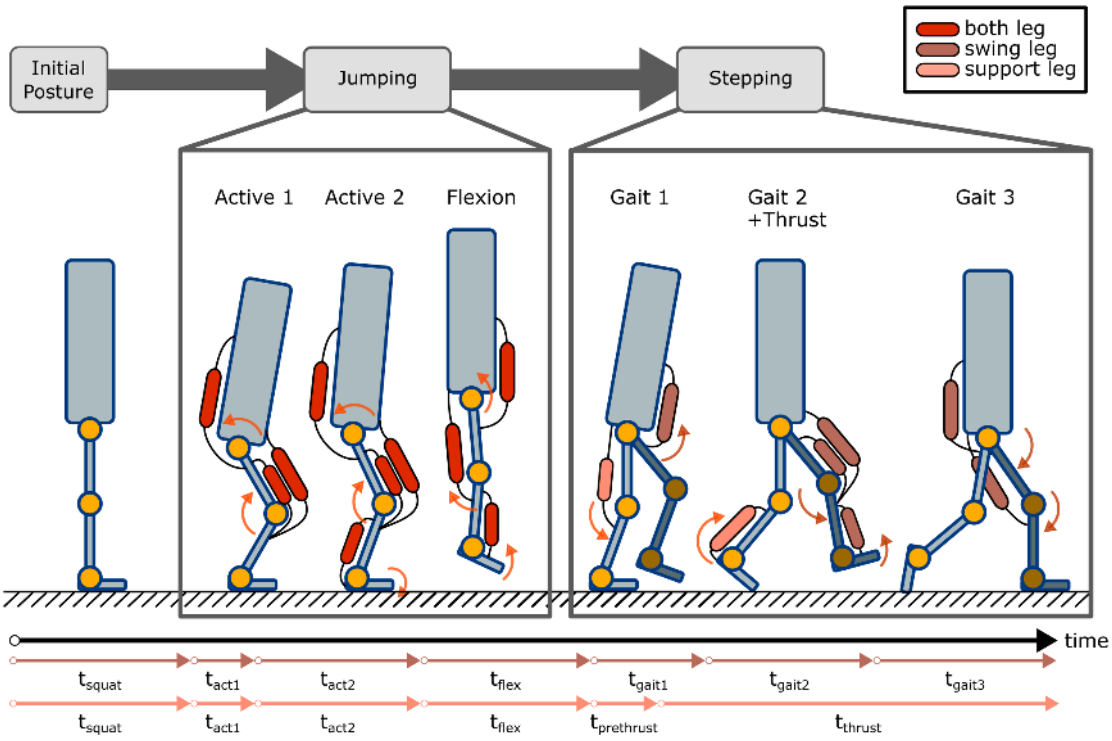


Fig.3.9: Proposed strategy for sequential jumping-stepping motion

The sequence started with an initial posture of standing upright position, achieved using postural balance control described in section 3.2. From initial posture, the robot did squatting by releasing pressure on muscles for t_{squat} time, until desired squatting posture achieved. The squatting posture was determined by knee joint angles. After squatting posture achieved, the jumping motion was triggered. We combined jumping control command and landing control command which described in subsection 3.4.1 and

subsection 3.4.2 as one jumping motion with post-motion strategy as preparation for next motion. This landing phase was very important in order to enable the robot having needed posture for the next motion and transfer the energy from jumping into stepping motion. In Fig. 3.9, the landing phase was shown as Flexion phase as the last phase of Jumping motion.

As post-motion command was done from jumping motion, stepping motion was initiated. The timing for the starting time of the first Gait 1, was controlled by feedforward control. The timing for Gait 1 was based on flexion time for landing t_{flex}. This feedforward parameter might be known based on preliminary experiments for jumping-only motion. Stepping motion was implemented as explained in subsection 3.4.3. Timing parameters for each gait phase were also feedforward controlled.

Chapter 4

Robot Design and Fabrication for Realizing Human-like Motion

In the previous chapter, we discussed a bio-inspired method for realizing dynamic motion from human muscle activation pattern. In order to implement this method, we made an actual robot in the real environment. This chapter explains in detail about the design and fabrication for the real robot which is inspired by the human musculoskeletal system.

We will start with the overall specification of the robot, followed by a further explanation for each system. The design of this robot was based on previous work of Nishikawa's jumping robot [6], with modification mostly on the muscle configuration and length to fit with our method. We concerned about muscle choice for the configuration to realize sufficient degree of freedom of the robot for posture balance and dynamic motion. Muscle length is also concerned as it is affecting the range of movement in each degree of freedom. We will also discuss how we made the pneumatic artificial muscle as the actuator of the robot, different types of muscle configuration, and attachment methods for the muscles. Electrical system and data processing flow will also be explained in the next section, including how we configured the hardware for interfacing with the processing unit. The last section will describe the pneumatic system used for the actuator.

4.1 Robot Overall Specification

We made the robot in size of a human kid, with height around 1.2 meters. We implemented muscle actuators only as lower extremity muscles for the robot and put almost all of the passive elements on the trunk, such as valves, boards, and tanks. By doing this, we concerned only with active lower extremity with passive upper extremity, so that the system might be similar to lower extremity dynamics of the human body.

The robot has 10 degrees of freedom which cover both the sagittal plane and frontal plane movement, based on human hip, knee, and ankle joints. We used potentiometers as sensors for detecting each joint angles. We also implemented an inertial measurement unit (IMU) to detect the orientation of the robot based on the angular rate and linear acceleration for detecting the orientation of the robot. We used pressure sensors attached

to each muscle actuators for reading the output air pressure from the pressure valve. All of the sensors implemented on the robot are integrated into one central processing unit which communicates directly to the server PC for user commands. Beaglebone Black was chosen as the main processing unit, as it has easy connectivity with server PC and built with interfaces for analog hardware like sensors. Complete specification of the robot is shown in Table 4.1.

Table4.1: Robot Overall Specification

Robot Specification	
Height	128 cm
Weight	10 kg
Degree of Freedom	5 DOF x 2 legs
Muscles	11 types of muscles
Sensors	10 Potentiometer
	22 Air Pressure Sensor
	1 IMU
CPU	Beaglebone Black
Air Pressure Supply	0.8 MPa
Power Supply	24 V DC (Valve)
	5 V DC (CPU)

4.2 Skeletal System

We designed the robot using PTC Creo Parametric as the CAD program. Parts for joints and other complex structures were designed to be 3d printed using nylon material. Other parts that worked as long bone that connect joints were fabricated using carbon fiber reinforced polymer (CFRP). Both nylon and CFRP material were chosen for their strong and light characteristics. We also used aluminium frames on some parts that need curve like placement for the air pressure buffer tank.

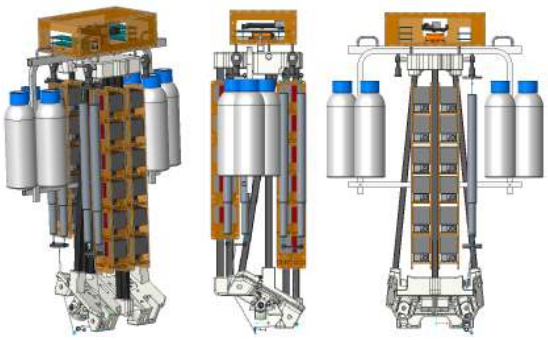


Fig.4.1: Acrylic box for valve placement and main processor unit

There was a limitation of space for valve placement so that we placed all valves on

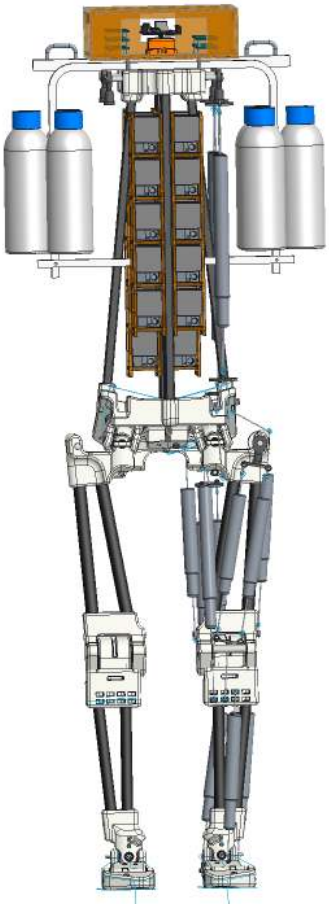


Fig.4.2: Robot design in CAD software

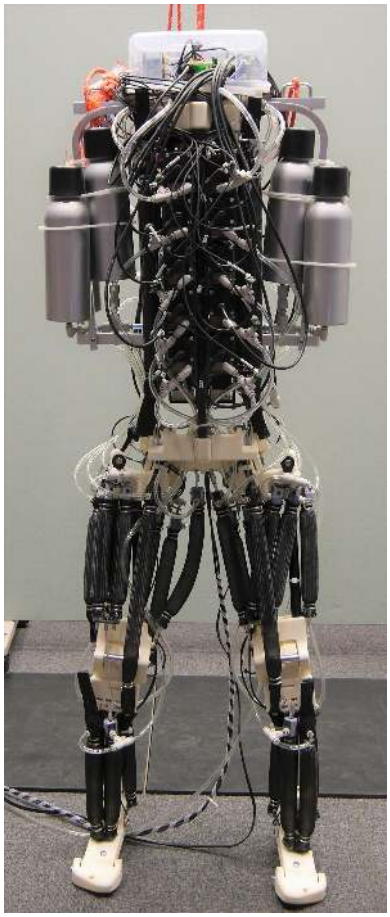


Fig.4.3: Final fabricated bipedal robot

the trunk. This was also done in purpose to make the weight of the robot focused on the trunk. We designed acrylic boxes for valve placement as shown in Fig 4.1. Since the valve placed far from the actuating muscles, we used elastic tubes for the muscles so that it is not limiting the movement of the robot. The CAD design for the robot and final fabricated robot are shown in Fig 4.2 and Fig 4.3.

4.3 Muscle Configuration

4.3.1 Inspiration from Human Muscle

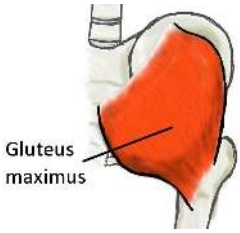
We chose the muscle based on the most important muscles acting on human lower extremity joints. First, we chose the degree of freedom of the robot based on human joints by simplifying the most important joint movements to move in the sagittal and frontal plane. We chose five degrees of freedom on each leg: hip flexion, hip abduction, knee flexion, ankle dorsiflexion, and ankle eversion. Next, we investigated which muscle play important roles in respective joint movements based on human anatomy [42].

The first degree of freedom is hip flexion which moves the robot in the sagittal plane. Most important muscles that work for this joint movement are Ilipsoas (IL) muscle and

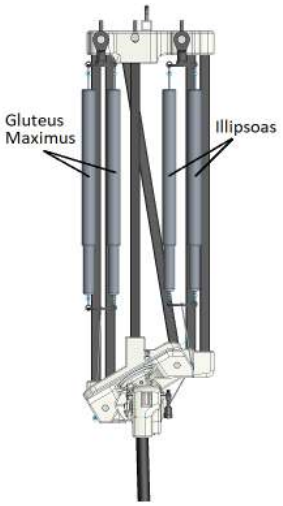
Gluteus maximus (GMAX) muscle, shown in Fig 4.4(a) and Fig 4.4(b). We designed the actuators for these muscles along the trunk since these muscles need a bigger force to move the trunk forward and backward. The configuration is shown in Fig 4.4(c). We also combined two muscle actuators as one type of muscle in order to generate bigger force. Both IL and GMAX worked as monoarticular muscles which move only in one joint movement.



(a) Ilipsoas (IL) as hip flexor



(b) Gluteus maximus (GMAX) as hip extensor



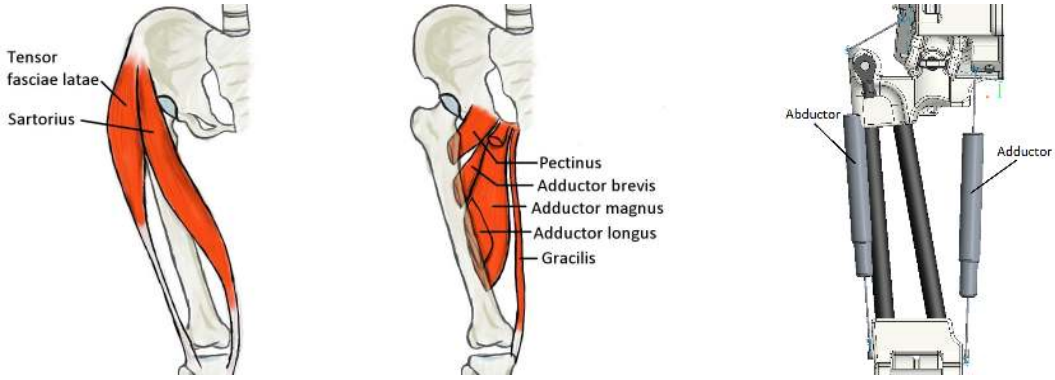
(c) GMAX and IL on CAD design

Fig.4.4: Active muscle for hip flexion motion

The second degree of freedom is hip abduction which moves the robot in the frontal plane. There are two groups of muscles that work in this joint movement: abductor (ABD) muscle and adductor (ADD) muscle. Both of these muscles work as monoarticular muscles. We designed the placement of the muscle actuators with the configuration as the muscles in human. The human muscle and robot muscle configuration are shown in Fig 4.5.

Next, we designed degree of freedom for knee flexion. There are monoarticular and biarticular muscles working on knee flexion. Vastus (VAS) muscle works as monoarticular muscle only for knee extensor, and rectus femoris (RF) muscle works as biarticular muscle that has role as a knee extensor and hip flexor. Hamstring (HAM) muscle works as a biarticular muscle that has a role as knee flexor and hip extensor. The position of these knee flexion muscles in human are shown in Fig 4.6(a) and Fig 4.6(b). We designed the muscle actuators based on VAS, RF, and HAM muscles. Since these groups of muscles consist of several muscles, we designed the actuators as a group of two muscle actuators for VAS and HAM to generate bigger force. We placed VAS, RF, and HAM as shown in Fig 4.6(c) and Fig 4.6(d).

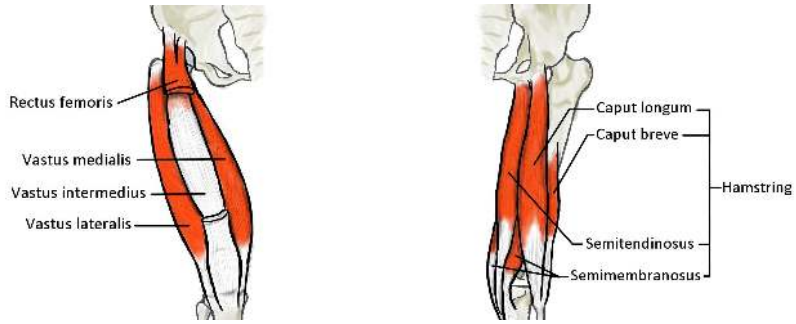
For movements on ankle and foot, we chose ankle dorsiflexion and ankle eversion as the degrees of freedom. Muscles that work for ankle dorsiflexion are shown in Fig 4.7(a) and Fig 4.7(b). Tibialis anterior (TA) as dorsiflexor and soleus (SOL) as plantar flexor are



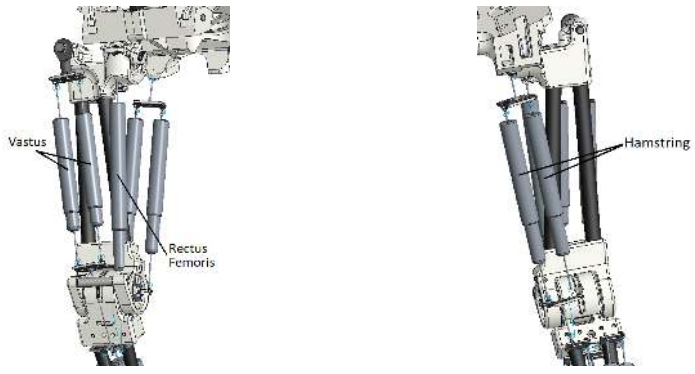
(a) Abductor group muscles (ABD) **(b)** Adductor group muscles (ADD) **(c)** ABD and ADD on CAD design

Fig.4.5: Active muscle for hip abduction movement

monoarticular muscles. There is also gastrocnemius (GAS) muscle which is a biarticular muscle that works as ankle plantarflexor and knee flexor. Muscles for ankle eversion, shown in Fig 4.7(c) and Fig 4.7(d), are fibularis (FB) as evertor and tibialis posterior (TP) as invertor. Fibularis and tibialis posterior also work as ankle dorsiflexion since their position places are in the posterior part of the leg. We chose only monoarticular muscle for ankle dorsiflexion and eversion to simplify the movement on ankle joint. The



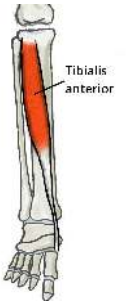
(a) Vastus (VAS) and Rectus Femoris (RF) as knee extensor **(b)** Hamstring (HAM) as knee flexor



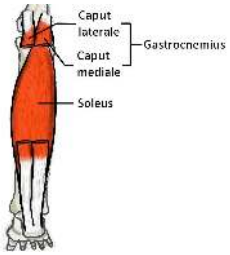
(c) VAS and RF on CAD design **(d)** HAM on CAD design

Fig.4.6: Active muscle for knee flexion movement

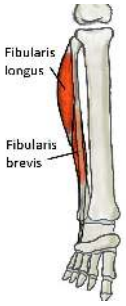
muscle actuators designed for the robot is shown in Fig 4.7(e) and Fig 4.7(f).



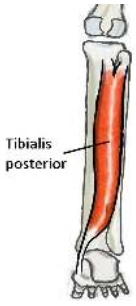
(a) Tibialis anterior (TA) as ankle dorsiflexor



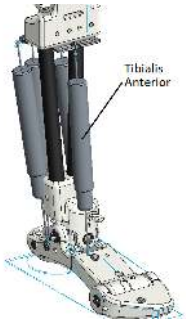
(b) Soleus (SOL) as ankle plantarflexor



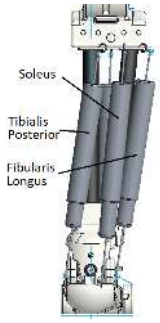
(c) Fibularis (FB) as ankle evertor



(d) Tibialis posterior (TP) as ankle invertor



(e) TA on CAD design



(f) SOL, FB, and TP on CAD design

Fig.4.7: Active muscle for ankle dorsiflexion and eversion movements

To summarize, all degree of freedom and muscle configuration of the designed robot is shown in Fig 4.8. We designed the degree of freedom to be limited only to pitch (sagittal plane) and roll (frontal plane), without considering yaw rotation.

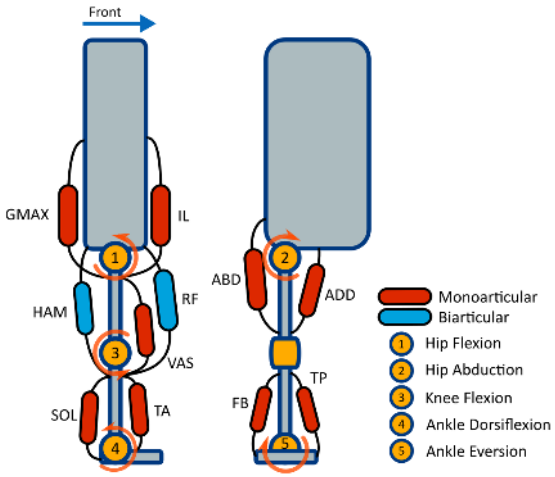


Fig.4.8: Muscle configuration and degrees of freedom of designed robot

4.3.2 McKibben-type Pneumatic Artificial Muscle as Actuator

As discussed in section 2.2, pneumatic artificial muscle (PAM) has similarity with characteristics of human muscle. For this feature, we chose to use McKibben-type PAM as our main actuator for the robot. There were two types of McKibben configuration: single-type and double-type McKibben. The type was chosen based on needed number of muscle to generate force on one muscle.



(a) Single type **(b)** Double type

Fig.4.9: McKibben-type Pneumatic Artificial Muscle

McKibben PAM has an advantage of its easy to fabricate feature. We made all the muscle from parts shown in Fig 4.10. Urethane tube was used as the muscle rubber inner tube. We used 3d-printed parts as the connector where air comes into the muscle, and plug as the end of the muscle. For attaching and adjusting the position of inner muscle tube into the connector and plug, silicon tubes were used below and above urethane tube. For preventing air leakage, we also put O-ring inside the PAM connector. Last, we tightened the rubber tube with stainless wire at the same position with the silicon tube.

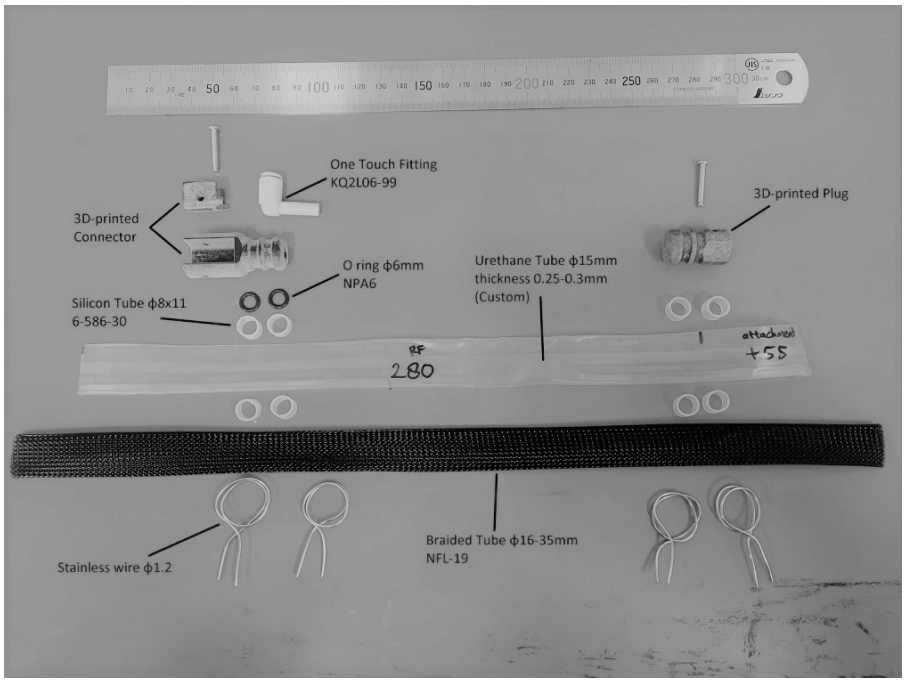


Fig.4.10: Parts for fabricating McKibben muscle

4.3.3 Muscle Actuator Length Calculation

In order to realize a robot with the capability of performing a human-like motion, joint angle range is important. Similiar to how we chose muscle configuration, we tried to design the joint angle range of the robot to be the same as the human. Therefore, muscle length calculation was done in order to satisfy desired joint angle range. We considered not only the effective muscle length but also other components of the McKibben muscle as the total length of the muscle. We made an approximate length calculation for the muscle with the total length of the muscles consist of muscle length in addition to connectors and wires. Using the 3d-printed connector and plug, we had a better calculation for desired muscle length in the design process. The muscle length and other parts length on one McKibben muscle in detail is shown in Fig 4.11.

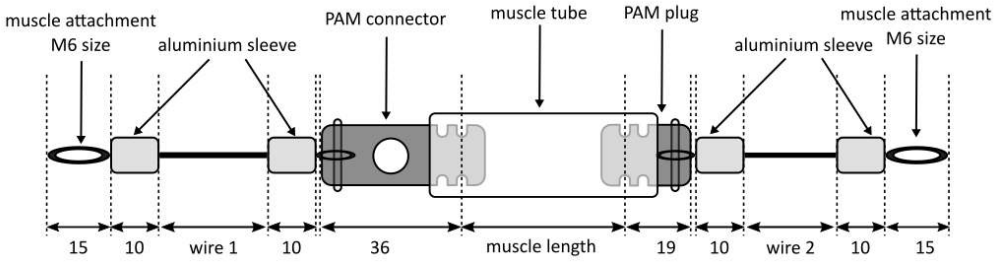


Fig.4.11: McKibben muscle parts length in detail (in milimeter)

For each degree of freedom, we did calculation on muscle length calculation based on the geometry of the muscle from their attachment points on the robot body. The maximum length change on maximum pressure was assumed to be 25% of initial muscle length. By initially set desired muscle length, we were able to get the wire length on each muscle and the total length of one muscle. All muscle length which designed for the fabricated robot based on the calculation is shown in Table 4.2. From the chosen muscle length, we got the joint angle range by geometry calculation with joint angle definition as shown in Fig. 4.12. For comparison, Table 4.3 shows both joint angle range in human and in the designed robot.

We confirmed the maximum joint angle range of the robot passively by moving the joints by hand without using the muscle actuator. The result is shown in Fig 4.13 for all joint angles of the robot. We checked the range on one joint angle by moving both right and left legs in the same direction at the same time, periodically from the maximum angle on one side to another side. On hip abduction, we were not able to do the same process since both legs will hit each other when the direction was inside the body (adduction). That is why in Fig 4.13() the graph does not show the same direction for right and left leg. The result of this passive joint angle range analysis shows that the muscle length calculation results in enough joint angle range to do a human-like motion.

Table4.2: Muscle Length

Muscle	Muscle name	Num	Muscle (mm)	Connector (mm)	Wire (mm)	Total Length (mm)
IL	Ilipsoas	2	300	97	174	571
GMAX	Gluteus maximus	2	300	97	252	649
VAS	Vastus	2	180	91	125	396
HAM	Hamstring	2	200	81	63	344
RF	Rectus femoris	1	280	105	60	445
TA	Tibialis anterior	2	180	75	2	257
SOL	Soleus	1	180	75	147	402
ADD	Adductor	1	180	105	8	293
ABD	Abductor	1	180	115	74	369
TP	Tibialis posterior	1	150	105	0	255
FB	Fibularis	1	150	105	0	255

Table4.3: Joint angle range comparison between robot (geometry calculation) and human

Joint	Human	Robot
Hip Flexion	-10° - 130°	-10° - 85°
Hip Abduction	-30° - 40°	-30° - 40°
Knee Flexion	-140° - 5°	-87° - 5°
Ankle DorsiFlexion	-50° - 30°	-50° - 30°
Ankle Eversion	-20° - 35°	-20° - 35°

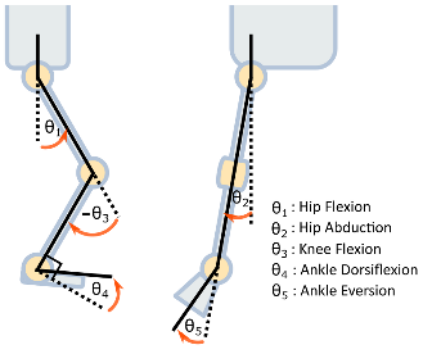


Fig.4.12: Joint angle definition on the fabricated robot

4.3.4 Force Profile Verification

Using MOF profile discussed in subsection 3.3.1, we verified the force profile of the robot. We checked the MOF profile of the robot on squatting position for jumping motion. The profile showed that the robot should be capable of doing vertical jumping motion analytically. MOF profile of the robot is shown in Fig. 4.15.

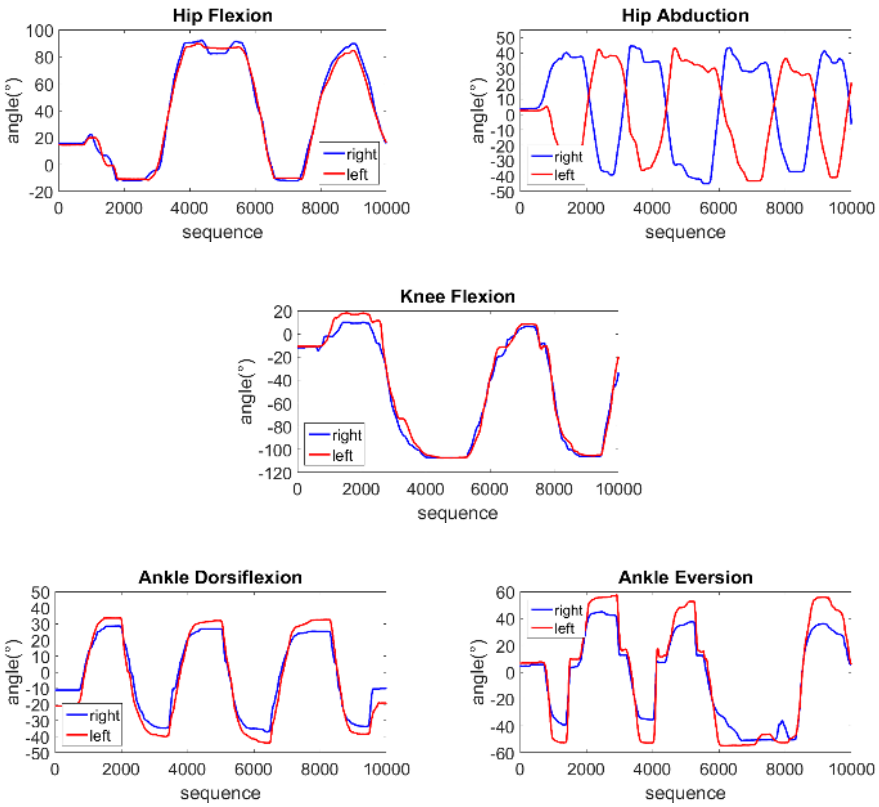


Fig.4.13: Joint angle range of the robot

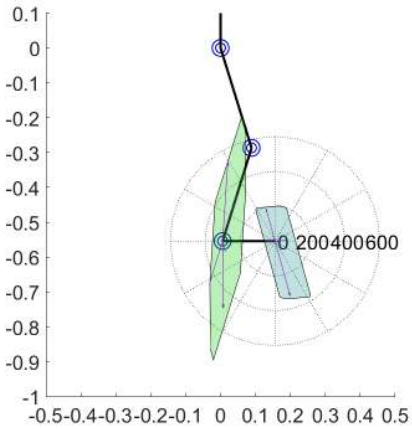


Fig.4.14: MOF profile of designed robot on squatting position for jumping

4.4 Electrical System

4.4.1 Data Flow for The System

In section 4.1, we already mentioned that the main processor of the robot and sensors embedded on the robot. In this section, we will describe more detail about the electrical system and data flow of the system. The sensor that we used on the robot will also be explained in this section.

The data flow of the system in the fabricated robot is shown in Fig 4.15. The user can communicate with the robot from server PC to the main processor of the robot which was connected in the same local area network through SSH. Beaglebone Black acted as the main processor unit, handled all the information within the robot: processed the command from server PC; gave commands to pressure valves; read and stored the sensor data. After all the process within the robot system was done, the user can access the data log from server PC for data processing.

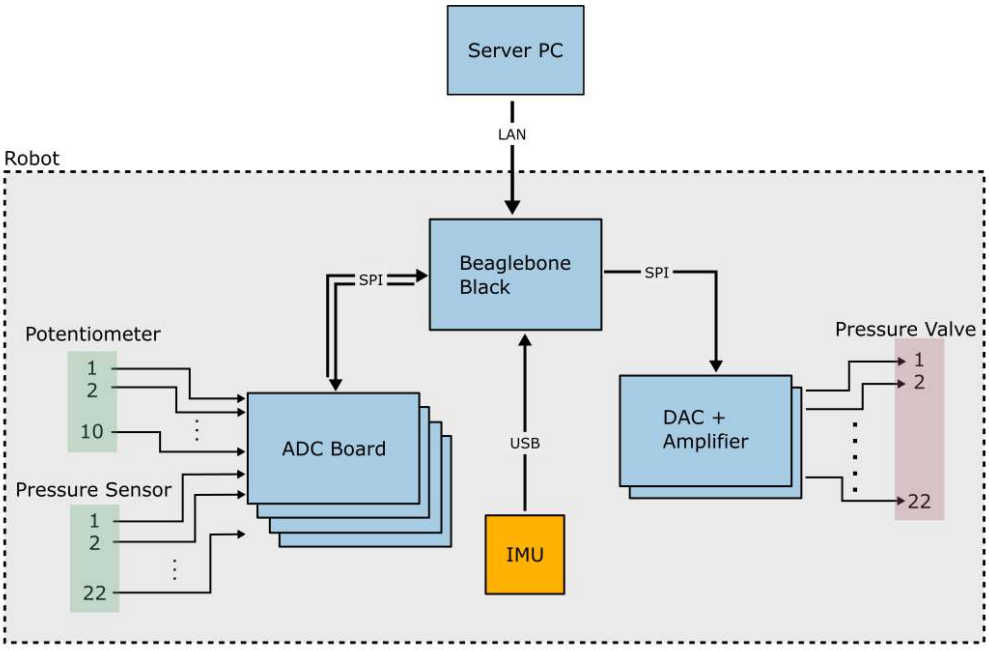


Fig.4.15: Data flow diagram for the robot

Analog data from potentiometers and pressure sensors were read by analog-to-digital converter (ADC) board and sent into Beaglebone Black via Serial Peripheral Interface (SPI). Unlike potentiometers and pressure sensors, orientation data from the inertial measurement unit (IMU) was sent directly via USB interface. The commands for muscle actuators were sent to digital-to-analog (DAC) Board via SPI and then sent into pressure

valve after being amplified for appropriate signals.

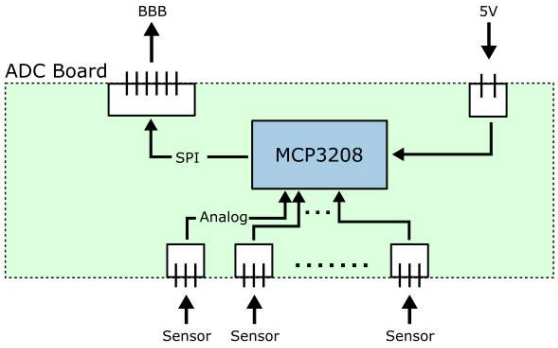


Fig.4.16: ADC Board system diagram

In the robot system, ADC Board acted as the main bridge between the analog sensors and Beaglebone Black. MCP3208 microchip was used to convert analog signal to 12-bit dimension digital signal with serial interface for communication with other devices. One ADC board handled eight analog sensors by the limitation of the ADC microchip. Therefore, we used four ADC boards to communicate with ten potentiometers and twenty two pressure sensors. The board used 5V power source so that we used the same power signal with power signal to Beaglebone Black.

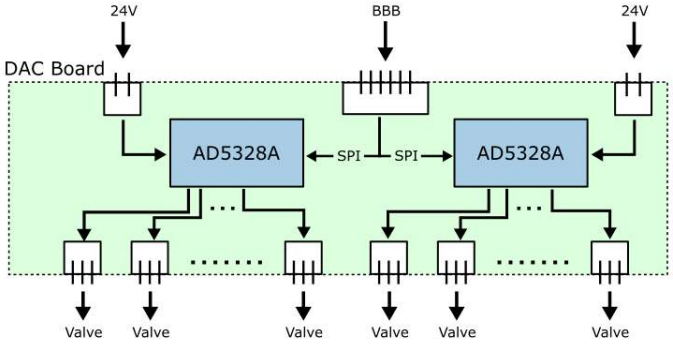


Fig.4.17: DAC Board diagram

The commands for muscles from Beaglebone Black was handled by DAC board. AD5328A microchip was used to convert digital command sent via SPI into an analog signal. Then converted analog signal was amplified to fit required 12V range signal for controlling the pressure valve. The DAC board also acted as the power source for the pressure valve which need 24V DC power source. Therefore we had to supply a different power source from the Beaglebone for the DAC Board.

4.4.2 Embedded Sensor

In order to give the robot sensing capability, we embedded sensors inside the robot system. Potentiometers, pressure sensors, and an inertial measurement unit (IMU) were embedded to provide information about the robot state into the processing unit. We used 10kΩ rotary potentiometer to sense the joint angle directly by putting the potentiometer boards on respective joints. In order to use this potentiometer as an angle sensor, we had to manually calibrate and choose values for zero degrees on each joint angle. We could get the angle by

$$\theta = \frac{D_{value} - D_{zero}}{4096} 360^{\circ} \tag{4.1}$$

where D_{zero} is potentiometer digital signal value for zero degrees. We calibrated the potentiometers by setting the robot into the desired position without any command for the muscles.

For sensing the muscle pressure, AG206-001MG pressure sensors were used and installed right after the output port of the pressure valve. The AG'2 pressure sensor that we used was able to read input pressure up to 1 MPa. Pressure value was derived from the output voltage from AG2 pressure sensor with

$$V_{out} = V_{DD}(P\alpha + \beta) \pm V_{DD} P_{err} T_{err} \alpha \tag{4.2}$$

$$P = \frac{1}{\alpha}(\frac{V_{out}}{V_{DD}} - \beta \mp P_{err} T_{err} \alpha) \tag{4.3}$$

where:

V_{out} = output voltage from AG2 pressure sensor[V]
V_{DD} = suply voltage 5V
P = pressure value [kPa]
P_{err} = pressure error, 15 kPa
T_{err} = temperature error multiplier, 1 for room temperature (0° - 85° C)
α = 0.0009
β = bias, 0.04

As for the IMU, we used MTi-20 IMU unit from Xsens which has features of embedded Kalman filter and USB interface. The IMU can only provide orientation, linear velocity, and angular rate data not position data, so that we just used the orientation data given by this IMU. The sensors embedded in the robot is shown in Fig 4.18



(a) Potentiometer

(b) Pressure Sensor



(c) MTi-20 Inertial Measurement Unit

Fig.4.18: Sensor embedded in robot system

4.5 Pneumatic System

We used air pressure supply from external air compressor with a big tank as the accumulator. The system was dependent on external air supply since the McKibben muscles were dynamically controlled by releasing and supplying air pressure into the muscles. Six air buffer tank was installed on the robot, placed on robot trunk just like a lung for a human. We made the buffer tank by modifying Eco Jet multifunction spray can. Pneumatic system diagram for the system is shown in Fig 4.19.

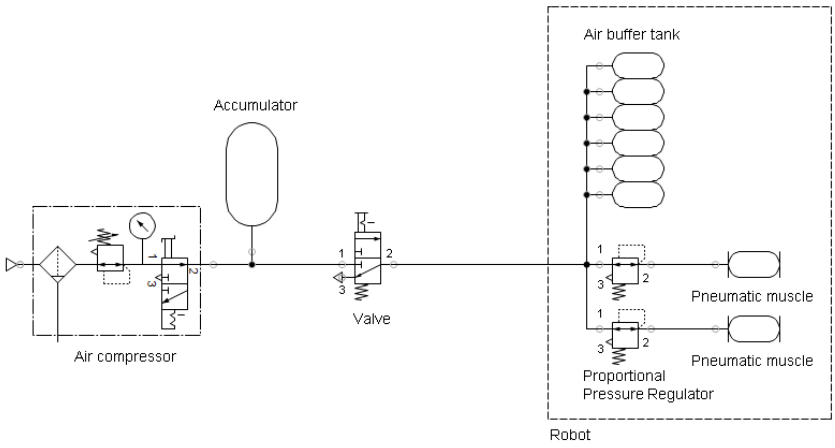


Fig.4.19: Pneumatic system diagram for muscle actuator

Proportional pressure regulators were utilized to control the McKibben muscle. We used Tecno basic PRE-U PS120006-080 from Hoerbiger as the pressure regulator. The regulator controlled output pressure based on command signal in form of exact desired output pressure. Maximum output pressure for this regulator was 0.8MPa. Command with bigger value will be saturated into maximum output pressure value. In order to increase time response for exhausting pressure from the muscles, we also used quick exhaust valves with one-touch fitting AQ340F from SMC. Pressure regulator and quick exhaust valve that were implemented are shown in Fig. 4.20 and Fig. 4.21.



Fig.4.20: Proportional pressure regulator



Fig.4.21: Quick exhaust valve

Chapter 5

Sequential Dynamic Motions Experiment

We implemented the method discussed in chapter 3 on the fabricated robot and did experiments to realize the combination of dynamic motions on real environment. We did the experiment consecutively based on the proposed sequence of dynamic motions as in Fig. 3.1. First, we tried postural balance control to get the robot into standing upright position as an initial posture for the dynamic motions. Next, we did experiment on various dynamic motions. Since the objective of the study was to realize sequential jumping-stepping motion, we tried a series of experiments on dynamic motions starting with independent motion experiments until the combination of those motions. The first dynamic motion was jumping motion only starting from an initial posture, without any concern about the after state of the robot. Next, we tried stepping motion, starting also from initial posture. Stepping was also done without concerning the state of the robot after the motion. Finally, we tried the experiment on sequential jumping-stepping motion. We tried to integrate the motion control command from each independent motion. However, jumping motion was different as we had to do landing control as preparation for stepping motion.

5.1 Standing Upright with Postural Balance Control

For postural balance control, we implemented the joint angle controller from Fig. 3.3. We got the type for the controller by doing trial and error with several types of controllers at first, from a bang-bang controller, P controller, and PI controller. The final type for the optimal controller was PI controller. Pressure change from the controller as control signal was

$$\Delta P(t) = K_p \theta_{error}(t) + K_i \int \theta_{error}(t) dt \tag{5.1}$$

Implementation of the PI controller needed conversion from continuous to discrete representation [43]. Therefore, we changed the representation from Eq. 5.1 using z-

transform into

$$\Delta P[n] = \Delta P[n-1] + K_p(\theta_{error}[n] - \theta_{error}[n-1]) + K_i T_s \theta_{error}[n] \tag{5.2}$$

with

$\Delta P[n]$	= current pressure change	$\Delta P[n-1]$	= previous pressure change
$\theta_{error}[n]$	= current joint angle error	$\theta_{error}[n-1]$	= previous joint angle error
K_p	= proportional gain	K_i	= integral gain
T_s	= sampling time		

To simplify, the equation can be changed into

$$\Delta P[n] = \Delta P[n-1] + P_{gain}(\theta_{error}[n] - \theta_{error}[n-1]) + I_{gain}\theta_{error}[n] \tag{5.3}$$

with $P_{gain} = K_p$ and $I_{gain} = K_i T_s$.

We did trial and error in order to find an optimal control parameter for the robot. We used $P_{gain} = 0.003$ and $I_{gain} = 0.002$ as the control parameter for optimal performance of the joint angle controller.

Pneumatic muscle actuator had limitation for its controllability because of its slow response and hysteresis. With this problem, oscillation might occur on smaller joint angle error. In order to prevent this, we used deadband region for the controller. We chose deadband of 4° (deadband between −2° and 2°) for joint angle error. There was no control signal produced while joint angle error inside this deadband. Control signal had a saturation limit, as input pressure for the robot was limited by 0.8 MPa.

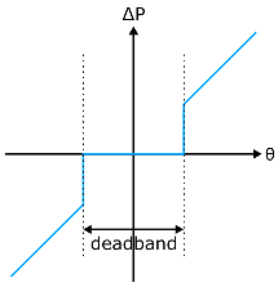


Fig.5.1: Deadband to prevent oscillation on joint angle controller

Initial pressure P_0 for each joint was set differently. All muscles on the sagittal plane were set with the same initial pressure of 0.3 MPa. For ADD and ABD muscles, we increased the initial pressure into 0.4 MPa as the moment arm for these muscles were shorter. For TP and FB, the placement was posterior to the leg so that they also affected ankle dorsiflexion. Therefore, we decreased the initial pressure to 0.22 MPa as the range of movement for this joint was relatively small. The initial pressure of all muscles is shown in Table 5.1. The saturation limit for a muscle was set based on initial pressure for each muscle, to be two times higher than initial pressure for the muscle.

The experiment for this postural balance control needed a proper setup for a successful attempt. The objective of this experiment was to get standing upright posture for initial

Table5.1: Initial pressure for each muscles

Muscle	Initial Pressure
IL	0.3 MPa
GMAX	0.3 MPa
VAS	0.3 MPa
HAM	0.3 MPa
RF	0.3 MPa
TA	0.3 MPa
SOL	0.3 MPa

Muscle	Initial Pressure
ADD	0.4 MPa
ABD	0.4 MPa
TP	0.22 MPa
FB	0.22 MPa

posture for dynamic motion. The postural balance controller was still not able to get the robot into standing posture from an arbitrary position. Therefore, we started the posture control while the robot hanged with no load on the leg.

Since the controller set the muscle pressure when there was no load, when the user put the robot on the ground, the pressure might be not sufficient for holding the load from its body. At this point, user help was needed to adjust the position to help joint angle controller to get needed pressure for maintaining the posture with load. The postural balance controller then controlled the muscle pressure in response to the posture adjustment set by the user. Fig. 5.2(a) shows how user set desired posture and muscle pressure.

After getting the right desired position and muscle pressure which worked for maintaining the posture, the postural balance controller was able to keep this posture. Deadband that was implemented in the controller played an important role in this task. Due



(a) Initial posture adjustment set by user



(b) Initial posture was maintained by joint angle controller. The robot fell by external disturbance after 40 minutes standing

Fig.5.2: Standing upright experiment using proposed postural balance controller

to the limit of memory on the main processor of the robot, the program was stopped and the command was not sent any more to the muscles. However, even without command from the processor, muscle pressure set by the postural balance controller was able to keep robot standing for about 40 minutes since the first time the desired posture was set. At minutes 40, an external disturbance occurred on the ground which made the robot fell without any controller work anymore. Fig. 5.2b shows how the experiment was done.

From this postural balance controller experiment, we got initial standing upright posture for our next dynamic motion. The initial standing posture for the robot is shown in Fig. 5.3. From the front view, we could see the posture of the robot mimic how human stand upright with the distance between feet was as wide as its shoulder in order to maintain balance in the frontal plane. From the side view, the robot had forward inclination on the trunk to compensate its center of mass and structure of the leg.

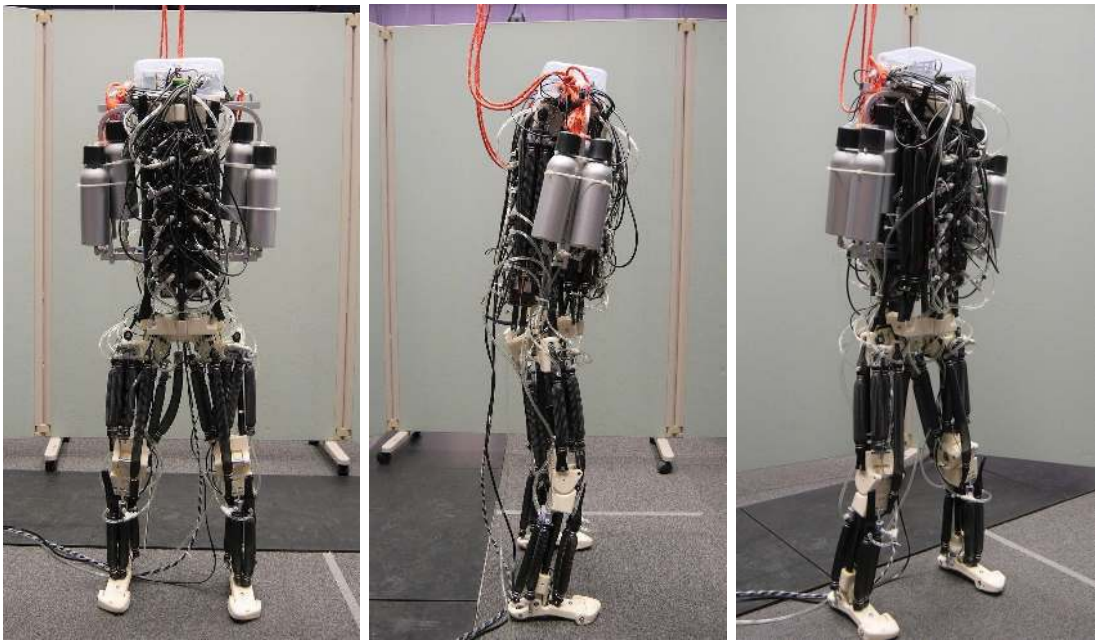


Fig.5.3: Initial standing upright posture achieved from postural balance controller from front and side view

For detailed information about the state of the robot, we collected the data from sensors on the robot during this experiment, including the posture adjustment by the user at the beginning. Each data was sampled not with fixed time sampling. From the log data, 1 data sequence was around 10 milliseconds. Joint angle data for all joints is shown in Fig. 5.4. It is shown that the adjustment by user was done for the first 6000 data sequences. The joint angle position for Fig. 5.3 is shown after those sequences. From the mechanical error, we could not get exact distributed mass on the robot and exact same length of the leg. With this error and as we used manually calibrated angle sensors by hand tuning, zero degrees did not mean to be a stable position for one angle. This is shown in right hip abduction and right ankle eversion. Orientation data read by the IMU sensor is shown in Fig. 5.5. It is shown that the pitch of the robot trunk was rotated as

the final standing posture had a forward inclined trunk.

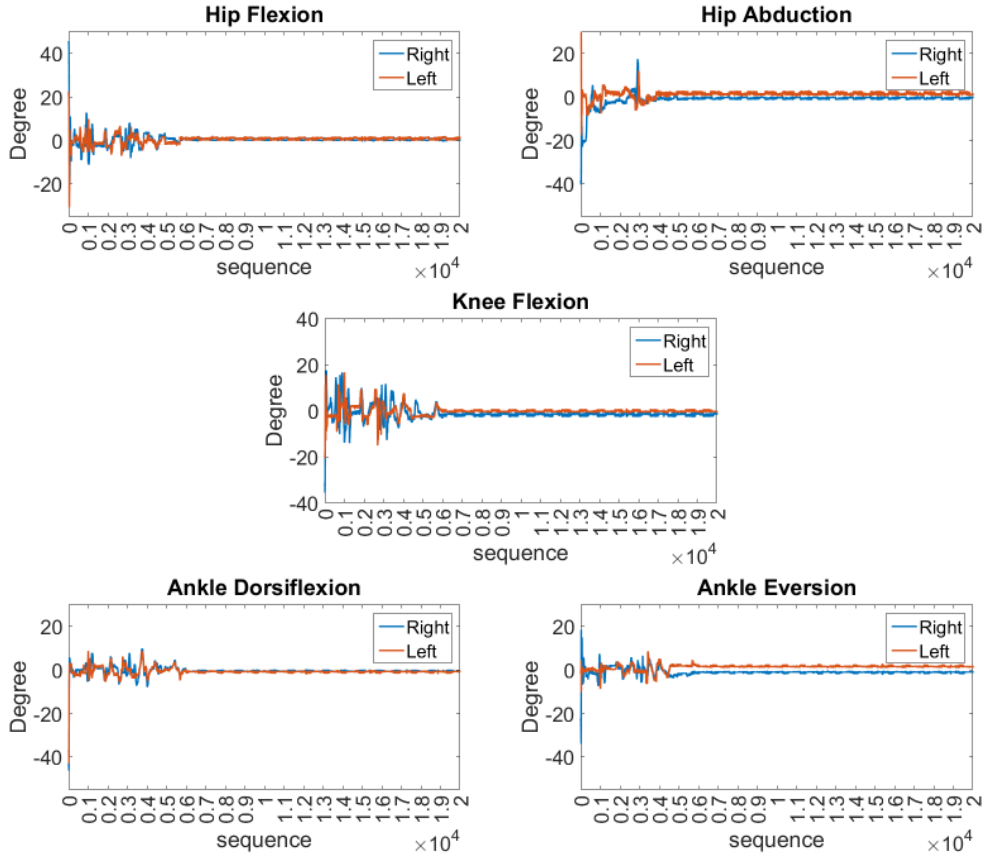


Fig.5.4: Joint angle for initial postural balance

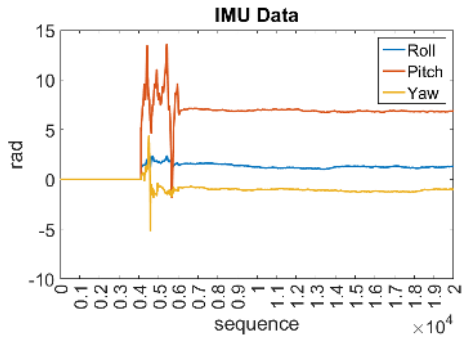


Fig.5.5: IMU data for standing upright task

Muscle pressure data for all muscles were also recorded as shown in Fig. 5.6. Required muscle pressure worked for each joint angles in standing upright posture is shown. The data shows how joint angle controller set the muscle pressure for muscles working in pair counterbalanced each other with the same pressure change from their initial pressure.

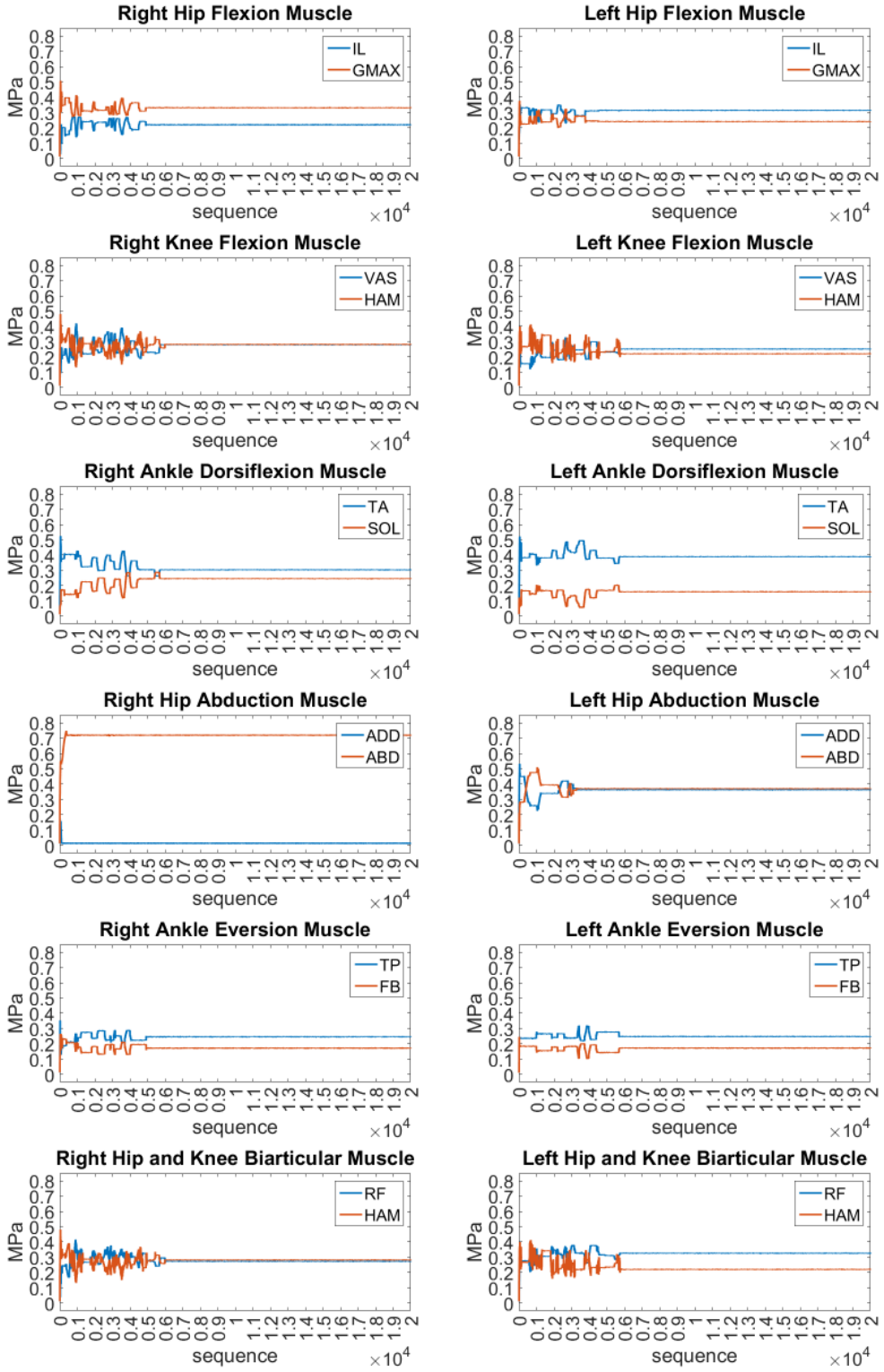


Fig.5.6: Muscle pressure on postural balance

5.2 Jumping Performance

The first dynamic motion done was jumping motion. Starting from an initial posture previously achieved, we tried to do the experiment based on the method discussed in subsection 3.4.1. The jumping motion control command needed to be done from squatting position. Therefore, we made the robot doing squatting from initial posture by releasing the pressure on sagittal plane muscles and made the robot squatting by gravitational force. We release the pressure until the desired squatting position was achieved. Squatting position was measured by knee joint angles. We set our desired knee angle as -25° so that as soon as angle sensors detect this value, the jumping motion will be triggered.

Muscle activation command was designed based on Fig. 3.6. Using our muscle configuration on the robot, during Active 1 Phase we activated GMAX as hip extensor, VAS and RF as knee extensors with the maximum pressure of 0.8 MPa to give maximum force. On Active 2, we activated SOL as ankle plantar flexor with 0.6 MPa. SOL was not pressurized with maximum pressure since SOL also contributes horizontal ground reaction force is in the backward direction. The SOL force vector for different pressure value can be seen in Fig. 5.7.

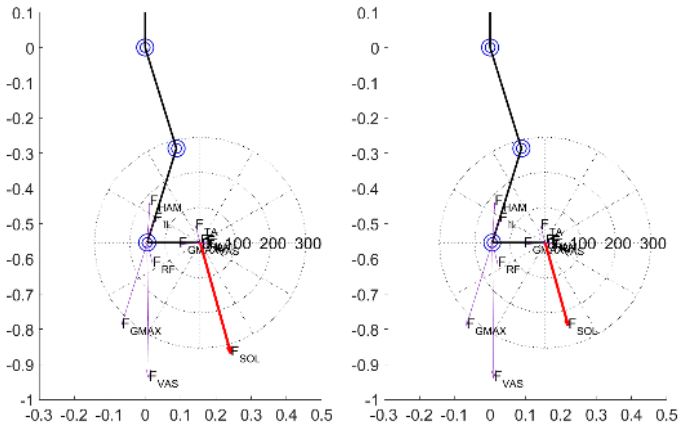


Fig.5.7: SOL force vector (red arrow) on jumping motion for 0.8 MPa (left) and 0.6 MPa (right)

For the muscle command, we did not activate only the extensors muscles, but also their antagonistic pairs. This was done in order to preserve some muscle pressure for landing motion as the post-motion strategy after jumping. This was needed as McKibben muscle has a drawback of slow time response. So that the flexors did not interfere with jumping performance, we activated the flexor pairs with 25% of extensors pressure. Timing between Active 1 and Active 2, t_{act1}, was chosen around 10 ms. This value was obtained from trial and error from several jumping experiments. All muscle activation command is shown in Fig. 5.8. The muscle pressure command before initial posture was not determined since it depended on the output of Postural Control. Therefore ABD, ADD, TP,

and FB values in Fig. 5.8 were assumed with values referenced from the initial posture.

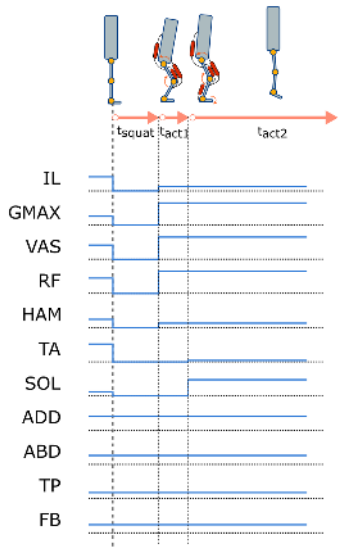


Fig.5.8: Muscle activation command for jumping motion

The muscle command defined as shown in Fig. 5.8 was realized for jumping motion with the fabricated robot. Fig. 5.9 shows recorded muscle pressure working on the robot. Time response for the muscle was good for exhausting air pressure, due to the implementation of quick exhaust valves. For supplying air pressure, slower time response occurred on higher pressure command. It can be inferred from the activation on Active1 phase that time response for getting 0.6 MPa was fast, but it was getting slower from 0.6 MPa to reach 0.7 MPa. This was caused by pressure drop on air buffer tanks because of simultaneous high-pressure consumption by many muscles.

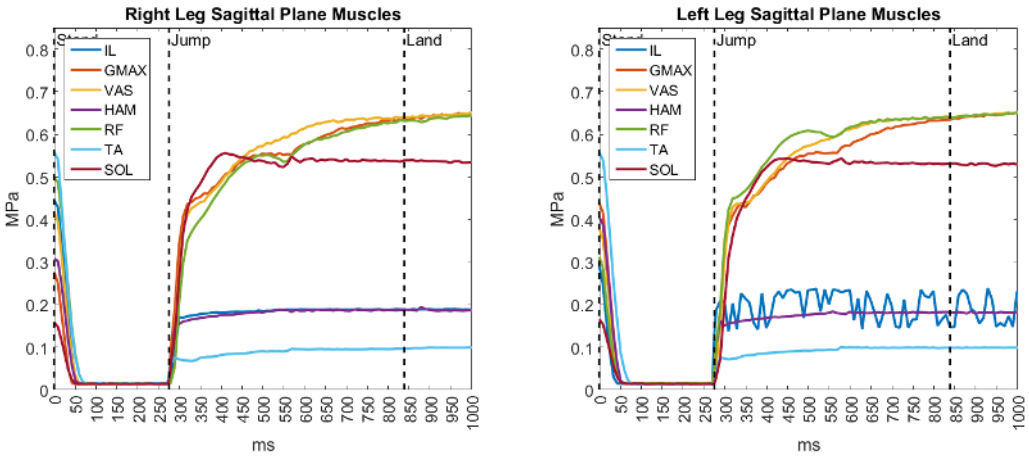


Fig.5.9: Muscle pressure for jumping experiment

Robot state was identified based on joint angles data and IMU data. We can see the

timing for the motion based on the recorded sensor data from the experiment. From IMU data shown in Fig. 5.10, it is shown that the Active 1 phase started at 277 ms from the start time and this was marked as the starting time of the jumping motion. Joint angle data for the experiment was recorded and shown in Fig. 5.11. From the joint angles, it is shown that after the Active 1 started, it took around 150 ms until the robot started moving upward. A landing was detected from ankle dorsiflexion when positive change happened at 839 ms.

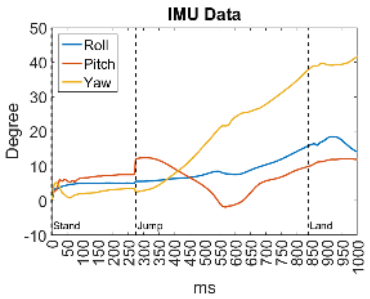


Fig.5.10: IMU data for jumping experiment

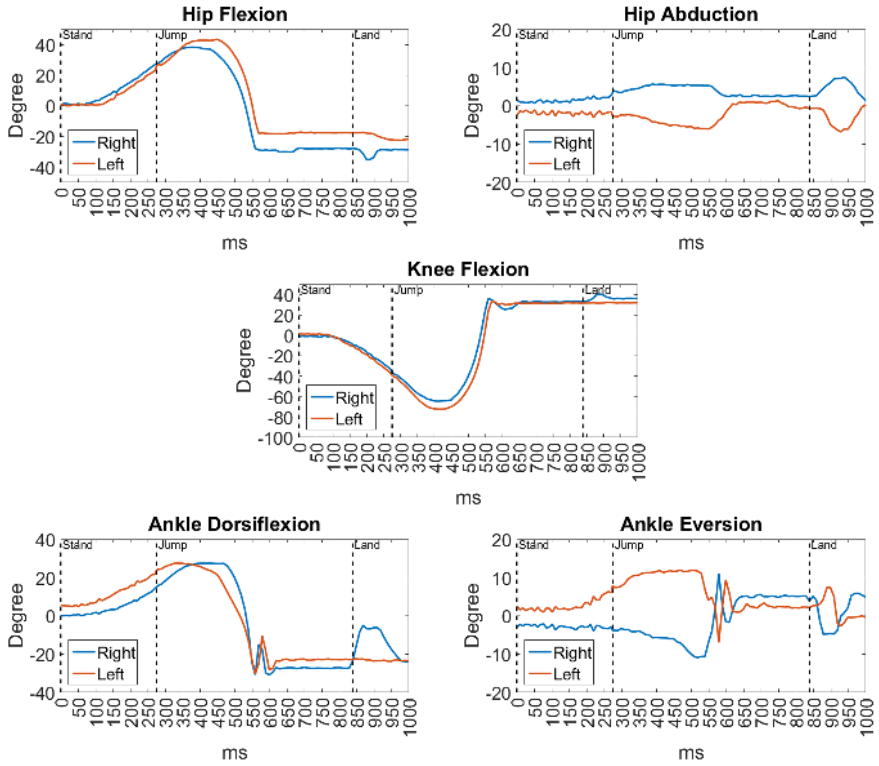
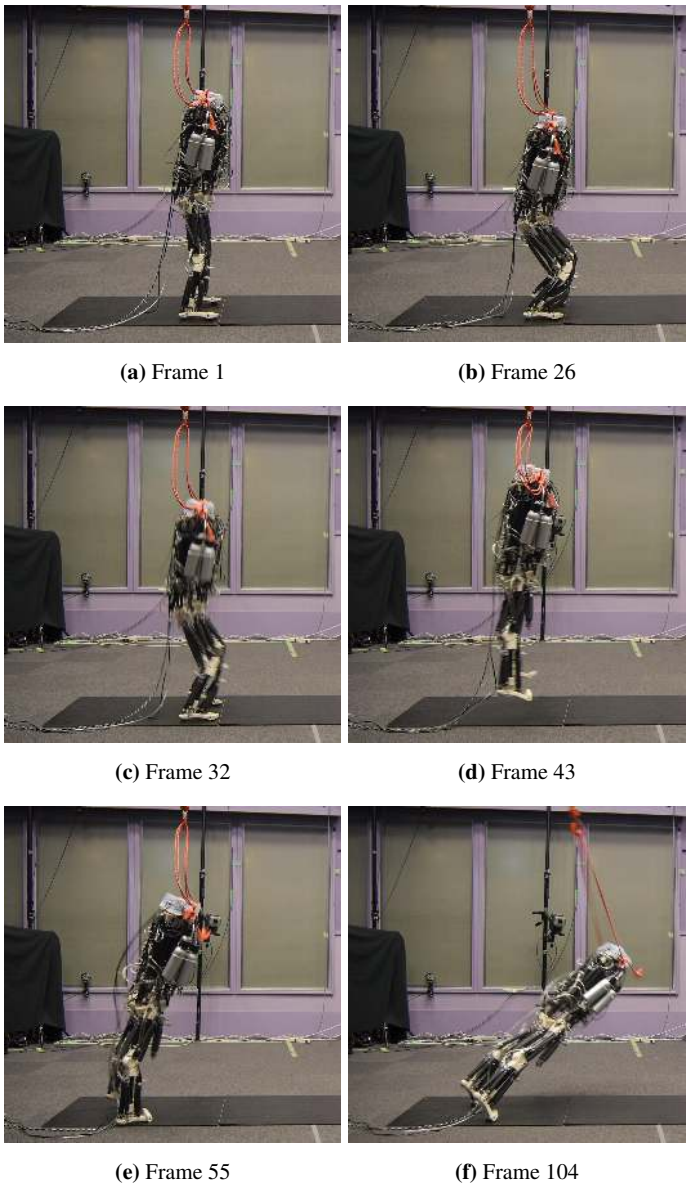


Fig.5.11: Joint angles data for jumping experiment

The result of jumping motion experiment was recorded with a camera with 59.94 frames per second. Frame showing important robot states was shown in Fig. 5.12. Frame

1 shows the initial posture of the robot, the same as the previous experiment on postural balance control. Frame 26 shows maximum squatting position, where right after this position the robot started to jump. Frame 43 shows maximum jumping height. In this state, robot muscles were already activated. It is shown that using maximum pressure resulted in a posture with maximum knee extension. The experiment was done without landing command as a post-motion strategy. Therefore, on Frame 55 it is shown that the robot landed with a stiff leg.



(a) Frame 1 **(b)** Frame 26

(c) Frame 32 **(d)** Frame 43

(e) Frame 55 **(f)** Frame 104

Fig.5.12: Jumping experiment recorded with 59.94 frames per second video

5.3 Single Stepping Experiment

Next dynamic motion was the stepping motion. Independent stepping motion experiment was done, with the objective of knowing the performance of a single stepping motion. Same with jumping motion, this stepping experiment was done starting from initial posture. The control method was implemented based on the control method discussed in subsection 3.4.1.

Direct after getting initial posture, the robot initiated Gait 1 for swing leg and pre thrust for support leg. In the experiment, we chose right leg as support leg and left leg as swing leg. This decision was based on the initial posture of the robot which slightly inclined toward the right side. Therefore, it was easier to do swing leg with left leg. In this experiment, the duration for both Gait 1 and Prethrust phases were same. This timing parameter was decided by trial and error. We defined the time parameter for the stepping motion as: t_{gait1} was 100 ms; and t_{gait2} was 300 ms. The control method designed as shown in Fig. 3.8 was actually only concerning sagittal plane movement. However, in real robot and real environment, a shift of the center of mass in the frontal plane also needed. Therefore, we also activated ABD on support leg to shift the projection of the center of mass to be supported by the support leg. The value of each muscle during each Gait phase is shown in Fig. 5.13.

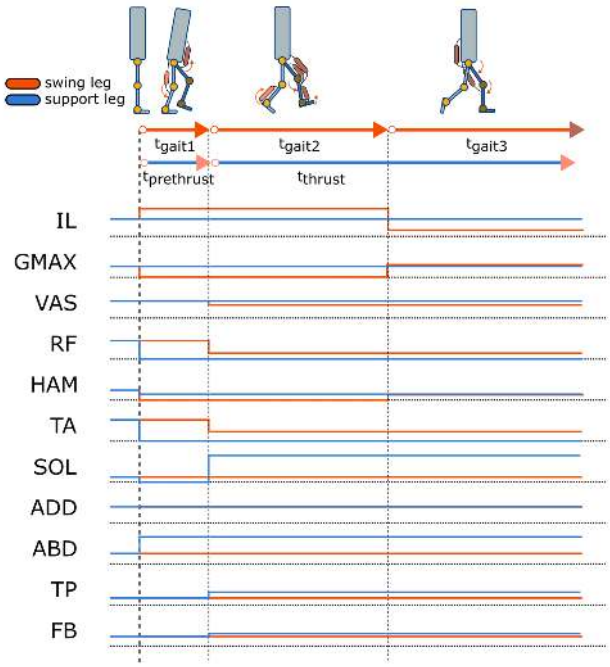


Fig.5.13: Muscle activation command for stepping motion

From the measured muscle pressure in Fig. 5.14, gait phases can be observed. Gait 1 started as soon as the robot was initiated to do the stepping motion by activating left

IL. Gait 2 and Thrust phase started after 84 ms of Gait 1 when right SOL started to be activated as the main thrust muscle. Finally, Gait 3 started at 348 ms after start time, when activation between left IL and GMAX were switched.

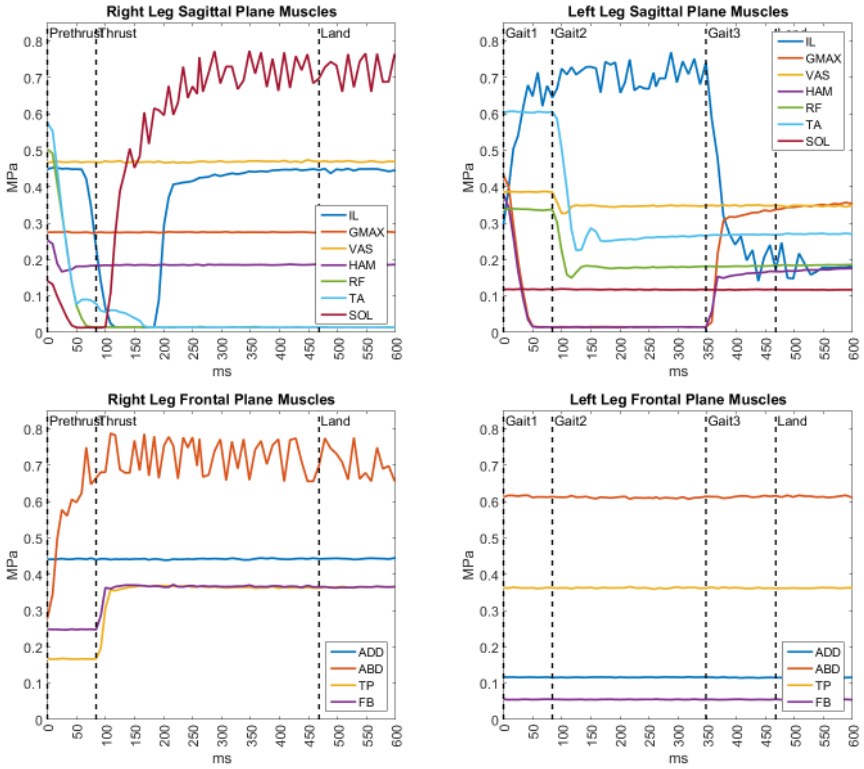


Fig.5.14: Muscle pressure for stepping experiment

Based on joint angles data in Fig. 5.15, the control method resulting in desired angle movement for stepping from our control method. For swing leg, the hip was flexed to lift swing leg and extended again for finishing the step on the last phase. Swing leg knee was flexed when the leg was lifted at Gait 1, extended for foot clearance at Gait 2, and flexed again at Gait 3. On the other hand, support leg flexed knee to make leg more compliant. On thrusting phase, support leg ankle dorsiflexed and plantarflexed in short time to give explosive thrusting. End of the stepping motion was defined as heel strike event. Heel strike was identified as changed in ankle dorsiflexion and eversion as the heel strike gave impact and changed those angles. Heel strike then was identified at 468 ms after the start time.

Recorded video of stepping experiment shows the result of the implementation of the control command in the real environment. For this experiment, we tried to record the result from front and side view to get both sagittal and frontal plane movement of the robot. The frames of key events of the motion are shown in Fig. 5.15 Frame 1 shows the initial posture for stepping motion. Frame 20 shows the posture of the robot just after Gait 1 already took effect and phase changed into Gait 2 and Thrust phase. At Frame 27, Gait 2 extended the swing leg knee and thrusting took effect on the motion. Because

of the rotation, the direction of the swing leg changed not in the forward direction but in leftward. Finally, at Frame 32 it shows when the heel struck the ground.

Since the ABD muscle was not capable of making big hip abduction, Gait 1 was not realized ideally as the control method proposed. The configuration of robot muscles did not specifically assign one muscle work only in one plane. For example, HAM muscle was placed in a placement that it also affect hip abduction. Due to this configuration, HAM on right leg for pre-thrust phase and left hip flexion from Gait 1 caused a rotation to happen at Frame 20. This result can be seen in change on yaw of the robot from IMU data shown in Fig. 5.16.

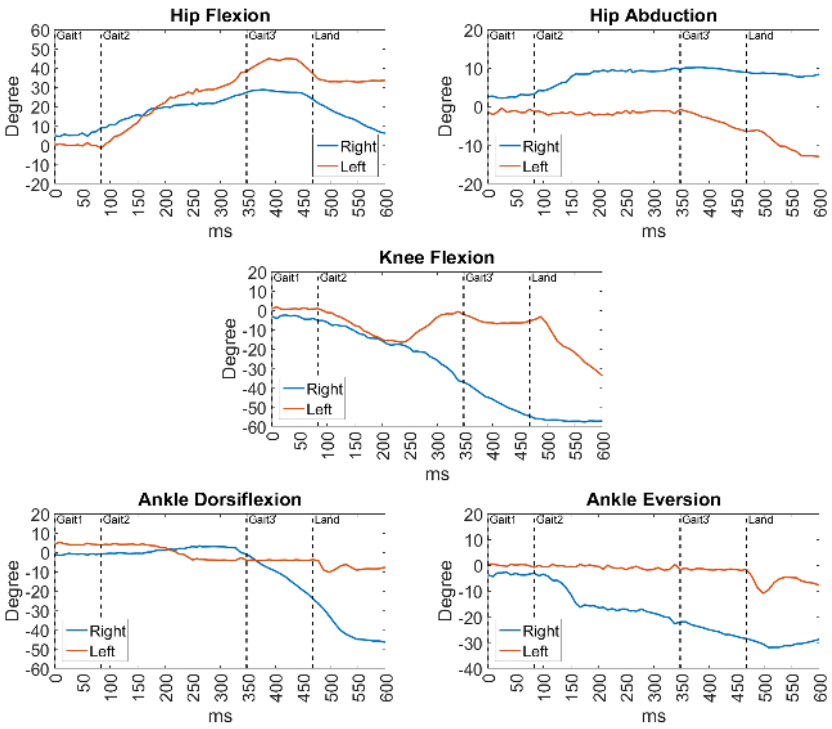


Fig.5.15: Joint angles for stepping experiment

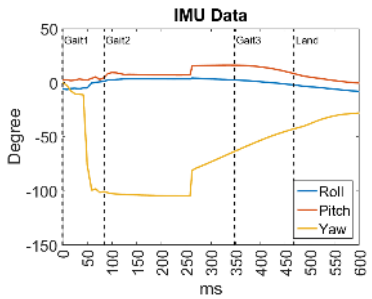
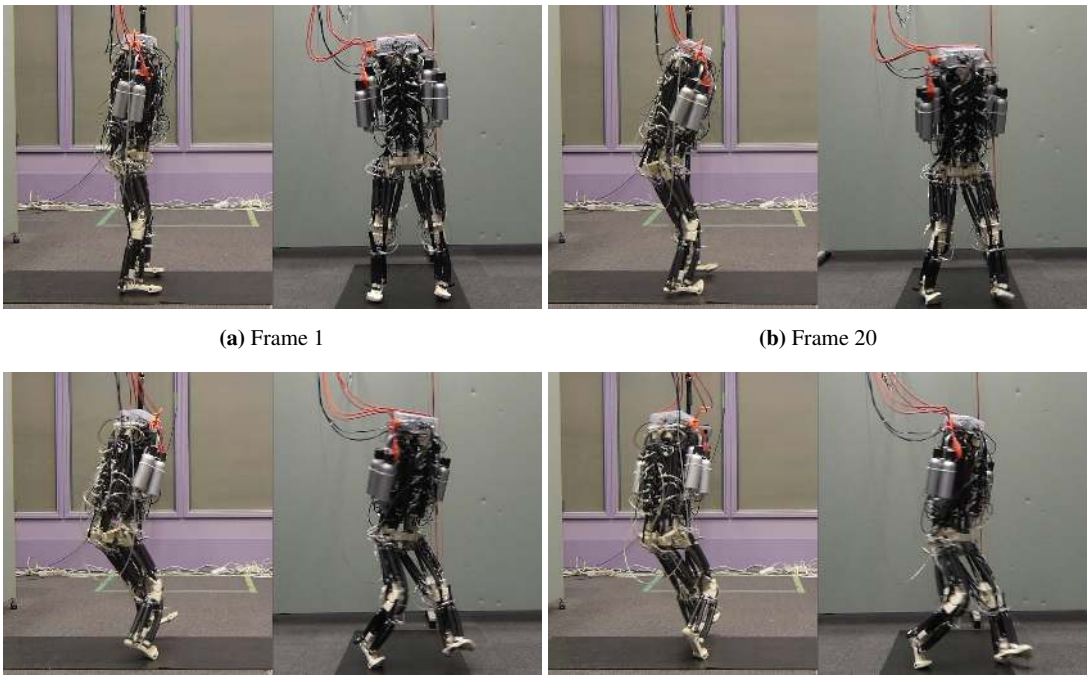


Fig.5.16: IMU data for stepping experiment

From this experiment, it was shown that the control command for stepping motion was

applicable to move swing leg into new position. However, the final result was not in the forward direction. Rotational movement around the yaw-axis happened towards the end of the stepping action which led to robot falling sideways. Therefore, stepping command resulted in robot achieved the stepping but then started to rotate during the action and fell. Stepping performance was defined as the total stepping time from start time of Gait 1 to heel strike event on swing leg which indicated stepping done by swing leg. From joint angle data (Fig. 5.15), stepping time for the independent stepping motion was 468 ms from standing.



(a) Frame 1 **(b)** Frame 20

(c) Frame 27 **(d)** Frame 32

Fig.5.17: Stepping experiment recorded with 59.94 frames per second video from side and front view

5.4 Sequential Jumping-Stepping

Previous experiments for dynamic motions was done independently. Through the experiments, it was shown that the muscle control method for both jumping and stepping motion was sufficient to realize the dynamic motions. In order to answer big question in this research about the capability of doing dynamic motion in sequence, we continued with next experiment. We tried to integrate both jumping and stepping muscle control command, to realize sequential jumping-stepping motion.

It was difficult to realize sequential motion if the control commands were just integrated as they were. From the jumping experiment in section 5.2, it was shown that the muscle command proposed in section 3.4.1 was able to realize jumping without concern-

ing landing state of the robot. Therefore, as we discussed in , we need to add post-motion strategy in jumping motion as landing control command.

Fig 5.18 shows the muscle command for our robot in this experiment. Key points of the muscle command here were on Flexion phase for landing command and Pre-thrust phase. Because of Flexion phase, now we concerned on the timing of Active 2 phase for jumping. Duration of t_{act2} determined the timing to flex the leg after the jump. Early activation of Flexion phase might affect the jumping performance. After the landing, stepping motion was directly initiated. The timing for starting the Gait 1 for swing leg and Pre-thrust for support leg was important here, which was determined by the duration of Flexion phase t_{flex}.

Since the dynamics for the initial state between jumping-stepping and stepping only were different, we also changed the command for Pre-thrust phase for support leg. We did not activate right ABD muscle since the Gait 1 was easier to do after jumping. This was because swing leg did not have to support the body before performing Gait 1. Instead, we activated right ADD muscle, since the landing state of the robot was inclined to the right side due to it initial posture.

For the experiment, we chose the following values as timing parameters for the command: t_{act1} for 10 ms; t_{act2} for 200 ms; t_{flex} for 200 ms; $t_{prethrust}$ for 50 ms ; t_{gait1} for 100 ms; and t_{gait2} for 300 ms; We changed the Pre-thrust phase duration to be faster since the motion became faster than stepping from standing. The measured muscle pressure during the experiment was shown in Fig. 5.19.

IMU data and Joint angles data are shown in Fig. 5.20 and Fig. 5.21 respectively. Methods for identifying key events of the motion were same with individual jumping and stepping motion which were based on these sensors data. Jumping motion started as

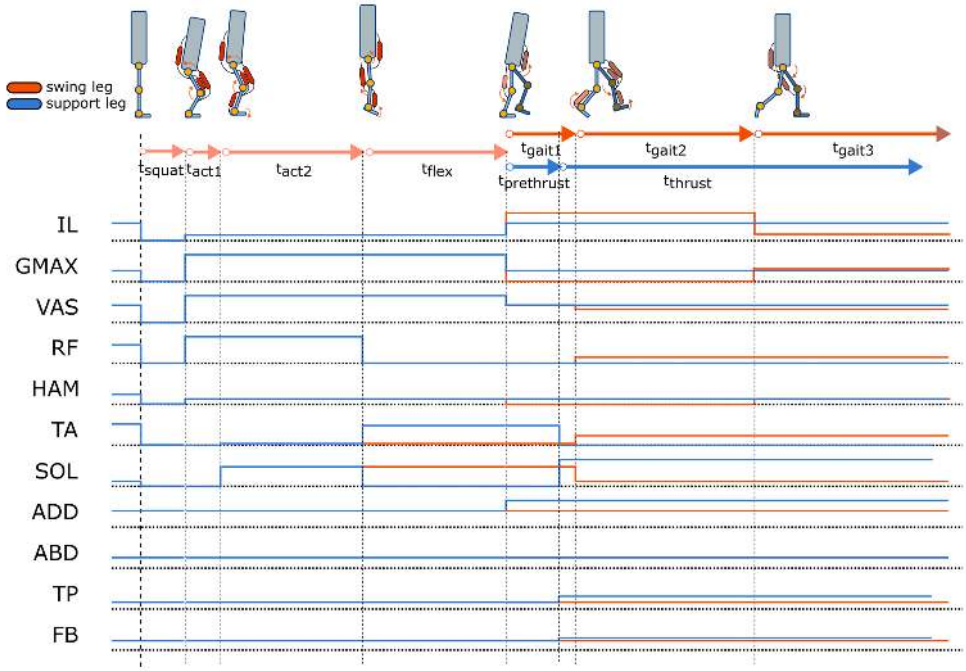


Fig.5.18: Muscle activation command for sequential jumping-stepping motion

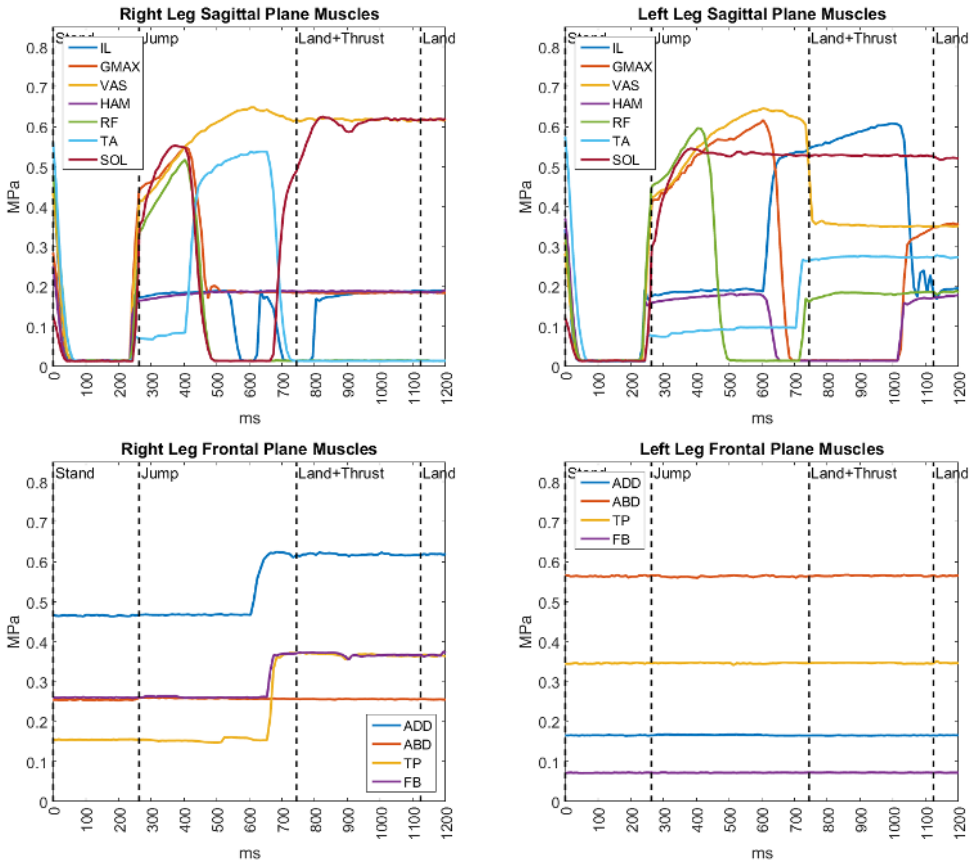


Fig.5.19: Muscle pressure for sequential jumping-stepping experiment

the orientation of the trunk had a sudden change in pitch as shown in IMU data. From the robot state, it is shown that jumping started at 263 ms after the start time. Landing for jumping was identified based on ankle joints as described in jumping and stepping experiments. From the data, it can be identified that the landing of jumping happened at 744 ms. We defined this landing time as the start time for stepping motion. The heel strike which indicated the end of stepping was identified at 1124 ms. Therefore, the time required for finishing the stepping was 380 ms.

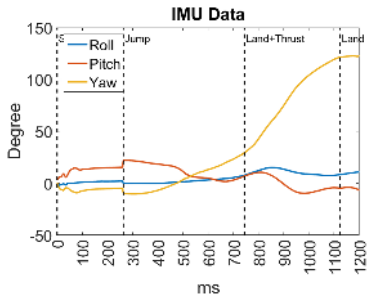


Fig.5.20: IMU data for sequential jumping-stepping experiment

Captured video frames from the jumping-stepping experiment are shown in Fig. 5.22.

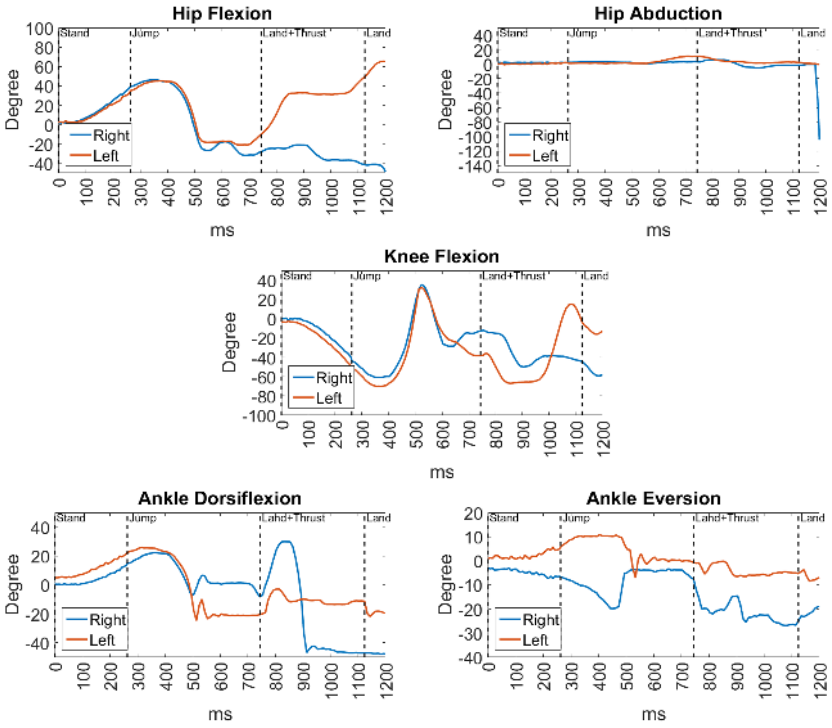
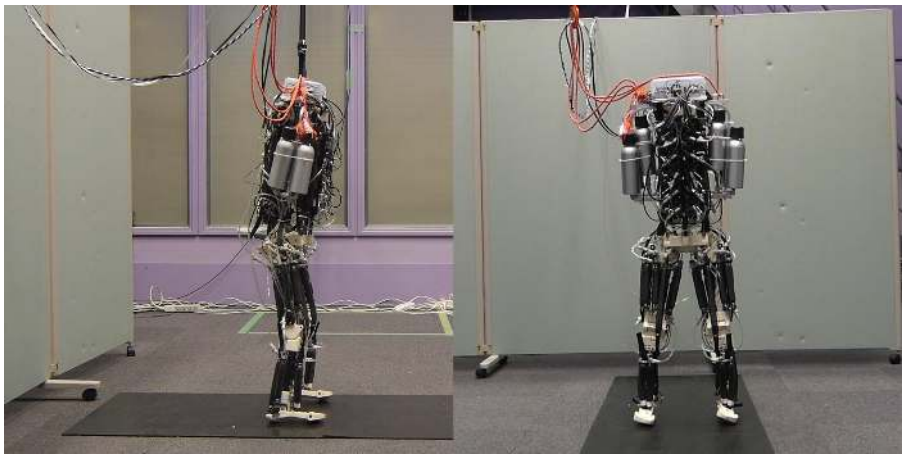


Fig.5.21: Joint angle for sequential jumping-stepping experiment

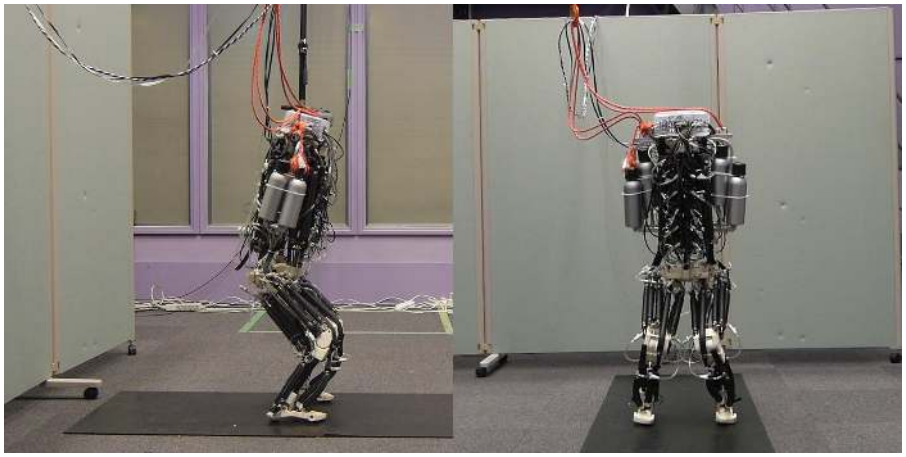
Important key events of the motion are shown in captured frames. Similar to jumping experiment, from Frame 1 to Frame 39 robot did jumping until reached maximum height. Until this key events, the motion was the same with individual jumping motion. The joint angles data in Fig. 5.21 show the extension of all joint happened caused by the jumping motion.

After jumping was done, while still being in the air, Flexion phase for landing strategy was initiated. From Fig. 5.19, Flexion phase happened at 404 ms, indicated by increasing pressure in TA and reduced pressure in SOL, RF, and GMAX. After flexion of the joints, Gait 1 for stepping motion was initiated before landing happened¿ It is shown that SOL, TP, and FB as thrust muscles on support leg were activated after flexion and before the landing time. The activation resulted in thrust muscles already had increasing pressure when landing and thrusting phase happened. This method tackled the problem of the slow response of muscle on high pressure.

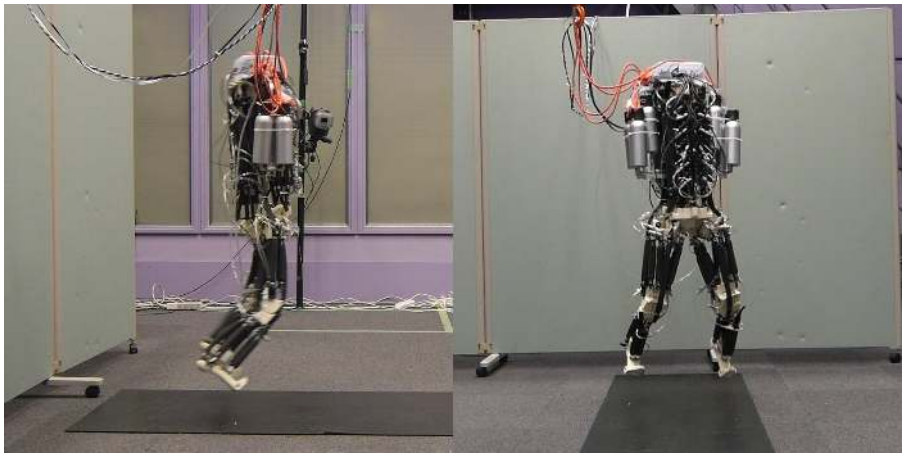
The landing posture of the robot is shown in Frame 50 in Fig. 5.22. As Gait 1 happened before landing, hip flexion of swing leg was increased, resulting in support leg landed first and the posture became as in Frame 50. This condition also affected in rotation of the robot, identified in yaw rotation data. As the robot rotated, in Frame 59 the thrusting resulting in stepping into leftward direction. Finally, the heel strike event happened in Frame 68, ending the sequential jumping-stepping motion in 1124 ms.



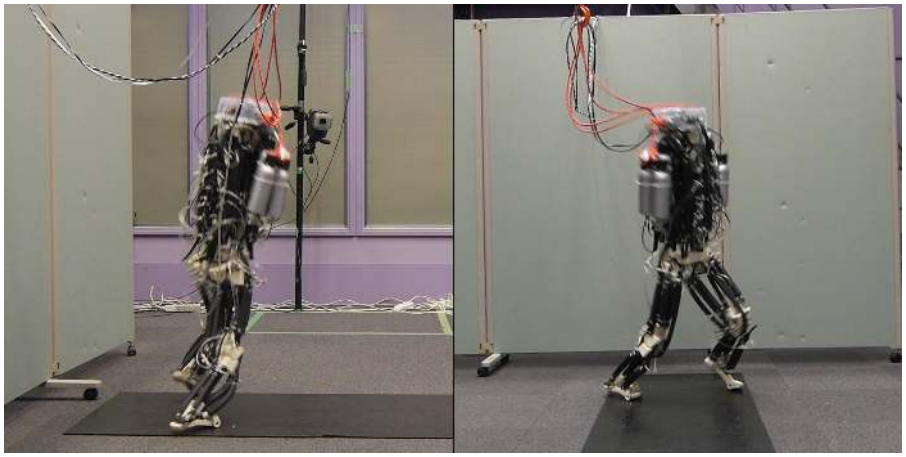
(a) Frame 1



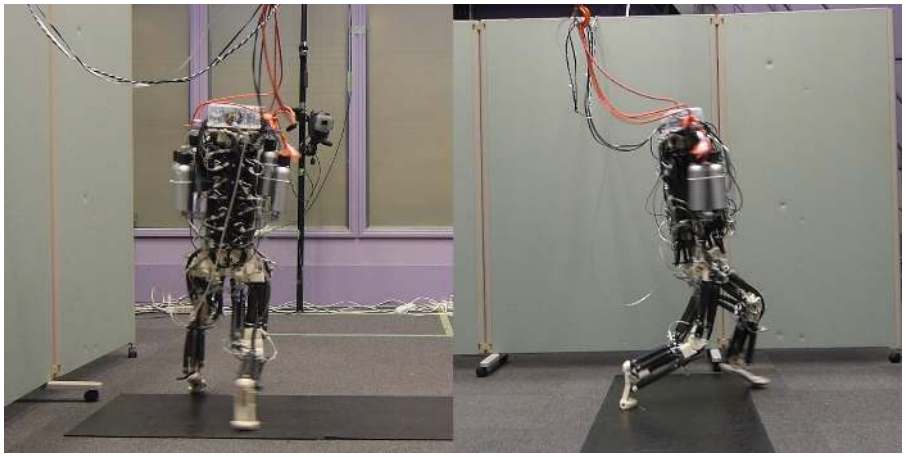
(b) Frame 22



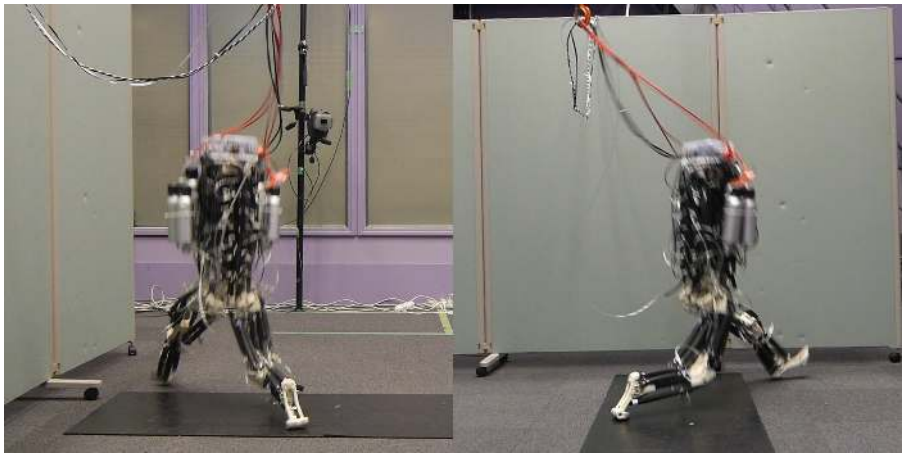
(c) Frame 39



(d) Frame 50



(e) Frame 59



(f) Frame 68

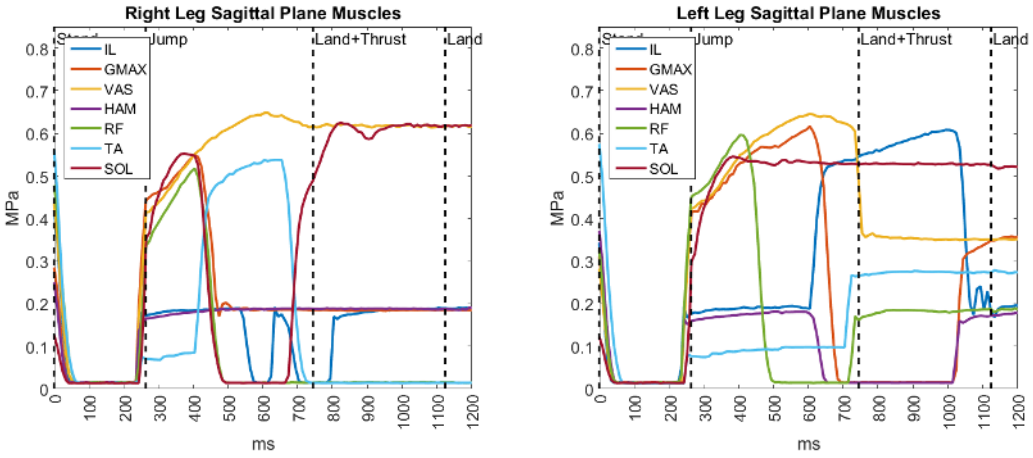
Fig.5.22: Sequential jumping-stepping experiment recorded with 59.94 frames per second video from side and front view

5.5 Discussion

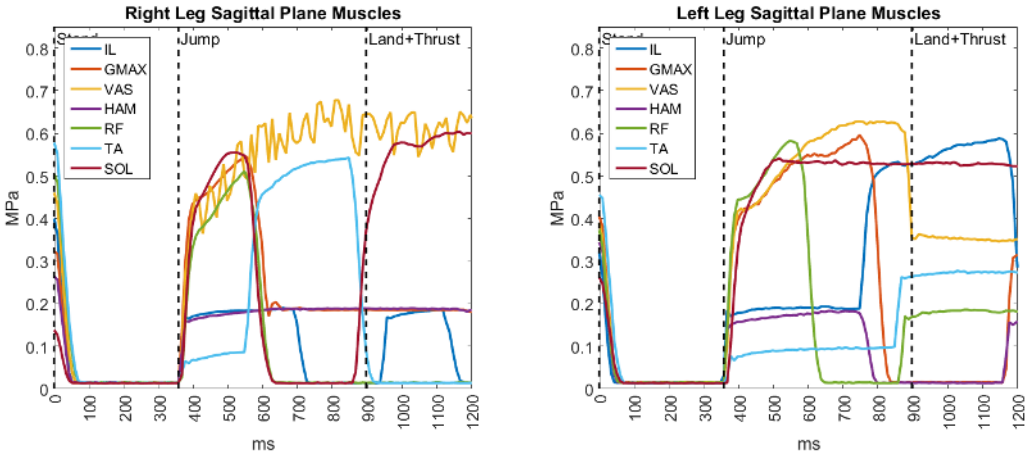
5.5.1 Effect of Flexion Timing Parameter

Proper flexion timing parameters before landing and thrusting for stepping were important for motion performance. We did experiments with varied timing parameters for t_{act2} and t_{flex} which affected the duration of flexion before landing and thrusting time for stepping. The difference is shown in Fig. 5.23. The timing for Fig. 5.23 (a) was with t_{act2} = 200 ms and t_{flex} = 200 ms, while Fig. 5.23 (b) was with t_{act2} = 300 ms and t_{flex} = 200 ms.

The difference resulted in unsuccessful stepping on slower timing. Even though landing and thrusting had happened, the final stepping was not done on the experiment with slower timing. It is shown in Fig. 5.24(b) that there is no change resulted from a heel



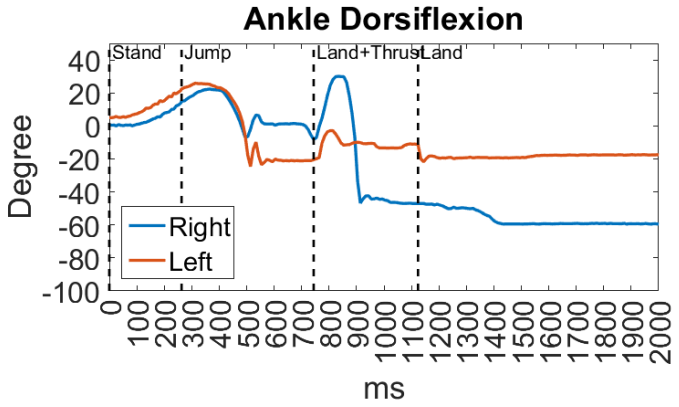
(a) Proper timing



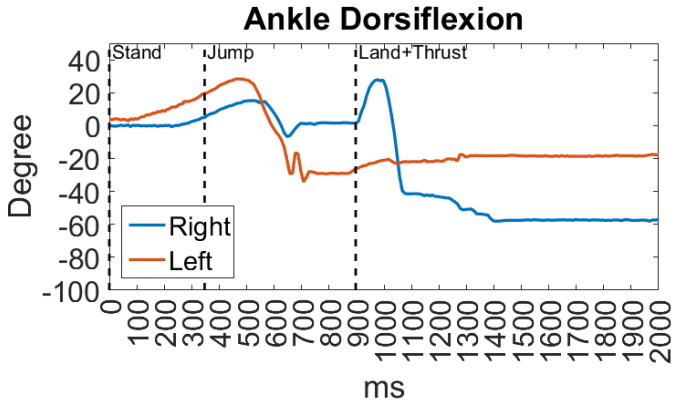
(b) Slower timing

Fig.5.23: Different timing for flexion and thrusting

strike on the slower timing parameter experiment.



(a) Ankle dorsiflexion with proper timing



(b) Ankle dorsiflexion with slower timing

Fig.5.24: Stepping performance for different muscle activation timing

We tried to verify the relation between flexion timing parameter and jumping time from jumping experiment. We conducted several jumping experiments to know the jumping time since first muscle command started until the robot landed. Jumping command time t_{act} was defined as the total of time of Active 1 and Active 2 phase so that

$$t_{act} = t_{act1} + t_{act2} \tag{5.4}$$

The jumping time data were collected from jumping experiments with both good and bad landing posture, as long as landing key event can be detected and jumping time can be measured. From 24 jumping experiments, the total jumping time data were obtained as shown in Fig. 5.25. For a successful sequential jumping-stepping motion, the effective total of t_{act} and t_{flex} should not exceed the total jumping time.

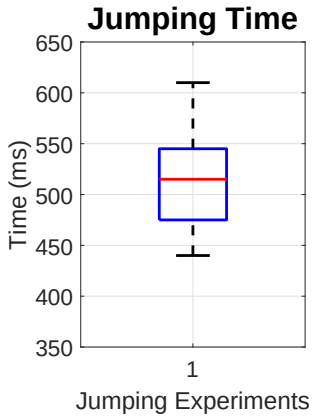


Fig.5.25: Jumping time data from 24 jumping experiments

5.5.2 Stepping without and with Pre-jumping Comparison

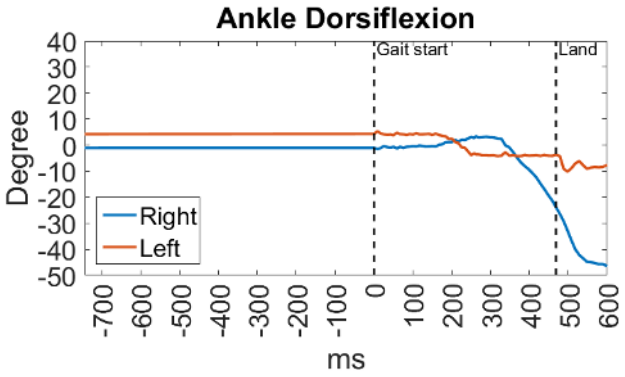
From the result, it was shown that sequential jumping-stepping motion resulted in faster stepping than single stepping motion. In order to make comparison easier, we set the timing for stepping and jumping-stepping to have same zero references on the start of stepping. Stepping motion was plotted as stepping without pre-jumping. The motion started at 0 ms. Jumping-stepping motion was plotted as stepping with pre-jumping. Motion started at -744 ms where pre-jumping started. Landing and stepping happened at 0 ms. The plot is shown in Fig. 5.26.

The experiment results showed that stepping without pre-jumping took 468 ms, where stepping with pre-jumping took 380 ms for finishing the step. Faster stepping was achieved due to energy transfer from earlier jumping motion to stepping motion, causing explosive movement on the thrusting phase of stepping. Similar energy transfer happened on squatting to the jumping motion.

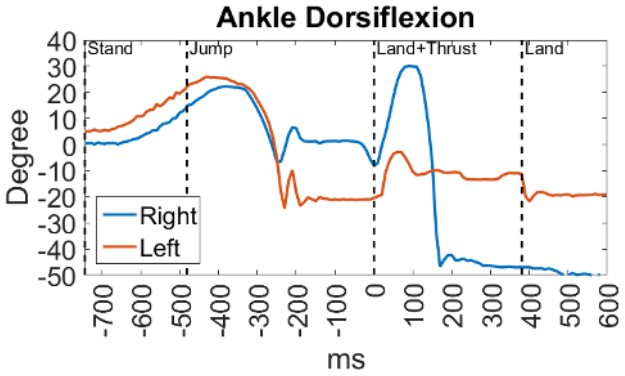
In order to verify this result, we conducted several experiments for both motions: five stepping experiments and six jumping-stepping experiments. We analyzed the stepping time with two-sample t-test. The results showed that the jumping-stepping motion had faster stepping time than the singe stepping motion with p-value = 0.0054.

5.5.3 Faster Thrusting on Jumping-Stepping motion

From the joint angle data for jumping-stepping motion shown in Fig. 5.21, it can be observed that high angular velocity happened on jumping-stepping motion during landing and thrusting key event. There are two possible reasons for this. The first reason is early activation of thrusting muscle before landing happened. From muscle pressure data on jumping-stepping motion in Fig. 5.19, it can be observed that thrusting was activated before landing. This compensated air supply delay for muscles and resulted in faster time response for thrusting at the time of landing. Another possible reason is energy transfer from jumping to stepping by muscle elasticity.



(a) Ankle dorsiflexion showing stepping time without pre-jumping



(b) Ankle dorsiflexion showing stepping time with pre-jumping

Fig.5.26: Comparison between stepping without pre-jumping and with pre-jumping)

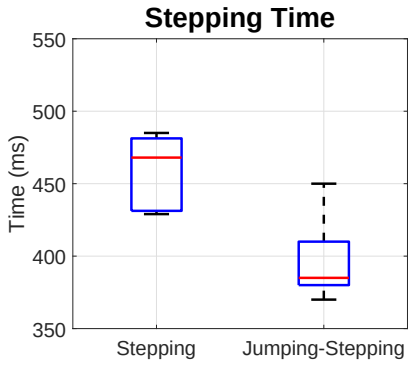


Fig.5.27: Stepping performance from five stepping motion experiments and six jumping-stepping motion experiments

5.5.4 Reproducibility

Experiments were done several times for each motion. For jumping experiments, we gave parameter for a successful jumping with landing feet achieved at the end of the jumping motion. With the last setup, the experiments were successful for 8 out of 12 experiments. We chose the one that was discussed in this thesis based on the best landing posture.

Jumping-stepping experiments with setup the same in this thesis were done for 11 times. The successful experiments were only 6 out of 11. We chose the best performance based on the posture of the robot, during and after the motion. Most of the successful experiments were successful in doing stepping, but having a backward inclined trunk.

Chapter 6

Conclusion and Future Work

6.1 Conclusion

Dynamic motion ability is a challenge in robotics. A musculoskeletal robot was one kind of robot which has a capability of doing the explosive movement for agile locomotion. There is still a challenge in the musculoskeletal robot by expanding its ability to perform different dynamic motions, especially in sequence.

We have done preliminary experiments with postural balance control on a PAM-based musculoskeletal humanoid robot. Postural balance control can be implemented in a musculoskeletal robot with PAM. However, precise control is difficult due to nonlinear property and slow time response of PAM. Implementing a deadband on the controller may be one solution to increase the performance of the controller. With pressurized muscles, the musculoskeletal robot can stand passively without any command. This can be achieved when the pressurized muscle is already with enough pressure to make the robot stand. Posture control can be used as the initial posture for performing the dynamic motion.

We also have performed several experiments on the musculoskeletal humanoid robot with dynamic motions. Vertical jumping can be done by transferring energy stored by squatting from standing to the jumping motion. This is possible by using PAM actuator by its high power-to-weight ratio. The direction of the jumping is affected by each muscle activated for jumping. Ankle plantar flexor muscles may be one of the most important muscles for performing jumping and changing the jumping direction.

Stepping motion can be defined as a sequence of phases, involving specific muscular activity on both support leg and swing leg. For stepping motion, timing is crucial for realizing a good stepping motion. Ankle plantar flexor muscles are also important in stepping motion as main thrusting muscle to move the body when stepping. With this, it is shown that ankle plantar flexion takes an important role in dynamic motions.

From the results, we found that performance of doing pre-jumping before stepping is better than stepping without pre-jumping. The robot performed stepping with pre-jumping for 380 ms of stepping time, while performed stepping without pre-jumping for 468 ms of stepping time. The result shows that doing sequential dynamic motion may beneficial for faster and more agile locomotion.

Combination of dynamic motion can be realized on a musculoskeletal robot with a

proper strategy. In order to realize a sequential dynamic motion, transition strategy is needed between first motion and second motion. The transition acts as post-motion of first motion and as preparation for fulfilling required posture or state for performing the second motion. In sequential jumping-stepping motion, landing command acted as the transition. Strategy to flex the knee and make the leg compliant is needed to absorb and transfer the energy from jumping to stepping. Flexing ankle also needed, in order to make the support leg able to give thrust on mid-swing phase of stepping.

6.2 Future Work

The big picture of the study is to realize a humanoid robot capable of various dynamic motion, as much as a human can do. Not only one task, but also in combination for a better performance.

Current study focused only on adding another dynamic motion after one dynamic motion. The motions were also specific to jumping and stepping. Therefore, the strategy is limited to a post-motion strategy for jumping, and preparation for stepping. In order to realize a robot capable of doing various agile locomotion, it will be better if a more general strategy can be developed.

We started with postural balance, with intention of applying postural balance as a post-motion strategy after a motion, to make the robot stop and return to its stability in a static posture. However, due to the limitation of the actuators, it is hard to realize such complex post-motion strategy. In that sense, improvement in actuator may be beneficial for that objective.

Better control approach may also be applicable for the musculoskeletal robot. Since the structure and mechanism are inspired from human and animal, control approach based on human and animal may also be beneficial, more than the common strategy for the motor-driven robot. By finding better control strategy and developing better actuator, we are hoping robot will be able to do more complex dynamic and agile motion.

Acknowledgement

All praise and gratitude be to Allah, Lord of the worlds, that His grace and blessing enables me to accomplish this research.

This research was conducted under supervising of Asst. Prof. Ryuma Niiyama from Intelligent Systems and Informatics (ISI) Laboratory, Department of Mechano-Informatics, Graduate School of Information Science and Technology, The University of Tokyo. I would first like to thank Asst. Prof. Niiyama for giving me advice on the basic foundation and good objective for the research. Asst. Prof. Niiyama also helped me in completing procedures for my scholarship even from when I was still in Indonesia.

I would also like to thank Prof. Yasuo Kuniyoshi as the director of ISI Lab for giving me advice with a broader view during Lab meetings. Many critical views and concrete suggestions for my research were given. I am really thankful for this opportunity to develop research skill in this lab.

I would also like to thank Research Associate Satoshi Nishikawa, for always giving me technical advice for my research from designing the robot, fabricating robot, and controlling the robot. Thank you for help with hardware problem or even presentation slide suggestion, with immediate response. I would also like to thank Researcher Kazutoshi Tanaka for giving me help on using the hardware and Beaglebone Black, also for ideas on the method for the research during a formal and informal discussion. Also, gratitude goes to other researchers on ISI Lab for giving me advice and suggestions on Lab meeting and other discussions. I am also grateful to be in Nishikawa-group and would like to thank all member of the group for their advice and good discussion. Thank also goes to Xi Chen who taught me to make my first McKibben muscle.

I would also like to thank Ministry of Education, Culture, Sports, Science, and Technology (MEXT) of Japan for giving me an opportunity to study at The University of Tokyo by supporting me with a scholarship.

I am also thankful to all my Indonesian Student Association at The University of Tokyo for helping me with minor problem and adaptation when first arriving in Japan.

Last but not least, I would like to express my gratitude to my mother, father, and sister, who continuously support me spiritually and give me the courage to chase my dream.

References

[1] Arthur J van Soest and Maarten F Bobbert. The contribution of muscle properties in the control of explosive movements. *Biological cybernetics*, Vol. 69, No. 3, pp. 195–204, 1993.

[2] Kazuo Hirai, Masato Hirose, Yuji Haikawa, and Toru Takenaka. The development of honda humanoid robot. In *Proc. IEEE International Conference on Robotics and Automation*, Vol. 2, pp. 1321–1326. IEEE, 1998.

[3] Jin'ichi Yamaguchi, Eiji Soga, Sadatoshi Inoue, and Atsuo Takanishi. Development of a bipedal humanoid robot-control method of whole body cooperative dynamic biped walking. In *Proc. IEEE International Conference on Robotics and Automation*, Vol. 1, pp. 368–374. IEEE, 1999.

[4] Keita Ogawa, Kenichi Narioka, and Koh Hosoda. Development of whole-body humanoid "pneumat-bs" with pneumatic musculoskeletal system. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4838–4843. IEEE, 2011.

[5] Ryuma Niiyama, Satoshi Nishikawa, and Yasuo Kuniyoshi. Athlete robot with applied human muscle activation patterns for bipedal running. In *IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, pp. 498–503. IEEE, 2010.

[6] Satoshi Nishikawa, Kazutoshi Tanaka, Kazuya Shida, Toshihiko Fukushima, Ryuma Niiyama, and Yasuo Kuniyoshi. A musculoskeletal bipedal robot designed with angle-dependent moment arm for dynamic motion from multiple states. *Advanced Robotics*, Vol. 28, No. 7, pp. 487–496, 2014.

[7] Koh Hosoda, Takashi Takuma, Atsushi Nakamoto, and Shinji Hayashi. Biped robot design powered by antagonistic pneumatic actuators for multi-modal locomotion. *Robotics and Autonomous Systems*, Vol. 56, No. 1, pp. 46–53, 2008.

[8] Ryota Uzu, Masahiro Shinya, and Shingo Oda. A split-step shortens the time to perform a choice reaction step-and-reach movement in a simulated tennis task. *Journal of Sports Sciences*, Vol. 27, No. 12, pp. 1233–1240, 2009.

[9] Marko JJ Nieminen, Jarmo M Piirainen, Jukka A Salmi, and Vesa Linnamo. Effects of neuromuscular function and split step on reaction speed in simulated tennis response. *European journal of sport science*, Vol. 14, No. 4, pp. 318–326, 2014.

[10] Yuki Asano, Toyotaka Kozuki, Soichi Ookubo, Masaya Kawamura, Shinsuke Nakashima, Takeshi Katayama, Iori Yanokura, Toshinori Hirose, Kento Kawaharazuka, Shogo Makino, et al. Human mimetic musculoskeletal humanoid kengoro toward real world physically interactive actions. In *IEEE-RAS 16th Interna-*

tional Conference on Humanoid Robots (Humanoids), pp. 876–883. IEEE, 2016.

[11] Katayon Radkhah, Christophe Maufroy, Moritz Maus, Dorian Scholz, Andre Seyfarth, and Oskar Von Stryk. Concept and design of the biobiped1 robot for human-like walking and running. *International Journal of Humanoid Robotics*, Vol. 8, No. 03, pp. 439–458, 2011.

[12] Maziar Ahmad Sharbafi, Christian Rode, Stefan Kurowski, Dorian Scholz, Rico Möckel, Katayon Radkhah, Guoping Zhao, Aida Mohammadinejad Rashty, Oskar von Stryk, and Andre Seyfarth. A new biarticular actuator design facilitates control of leg function in biobiped3. *Bioinspiration & biomimetics*, Vol. 11, No. 4, p. 046003, 2016.

[13] Gill A Pratt and Matthew M Williamson. Series elastic actuators. In *Intelligent Robots and Systems 95.'Human Robot Interaction and Cooperative Robots', Proceedings. 1995 IEEE/RSJ International Conference on*, Vol. 1, pp. 399–406. IEEE, 1995.

[14] Katayon Radkhah and Oskar Von Stryk. A study of the passive rebound behavior of bipedal robots with stiff and different types of elastic actuation. In *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5095–5102. IEEE, 2014.

[15] Jin'ichi Yamaguchi and Atsuo Takanishi. Development of a biped walking robot having antagonistic driven joints using nonlinear spring mechanism. In *Proc. IEEE International Conference on Robotics and Automation*, Vol. 1, pp. 185–192. IEEE, 1997.

[16] Ikuo Mizuuchi, Yuto Nakanishi, Yoshinao Sodeyama, Yuta Namiki, Tamaki Nishino, Naoya Muramatsu, Junichi Urata, Kazuo Hongo, Tomoaki Yoshikai, and Masayuki Inaba. An advanced musculoskeletal humanoid kojiro. In *IEEE-RAS International Conference on Humanoid Robots*, pp. 294–299. IEEE, 2007.

[17] Yuto Nakanishi, Tamon Izawa, Masahiko Osada, Nobuyuki Ito, Shigeki Ohta, Junichi Urata, and Masayuki Inaba. Development of musculoskeletal humanoid kenzoh with mechanical compliance changeable tendons by nonlinear spring unit. In *IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pp. 2384–2389. IEEE, 2011.

[18] Yuto Nakanishi, Yuki Asano, Toyotaka Kozuki, Hironori Mizoguchi, Yotaro Motegi, Masahiko Osada, Takuma Shirai, Junichi Urata, Kei Okada, and Masayuki Inaba. Design concept of detail musculoskeletal humanoid "kenshiro" -toward a real human body musculoskeletal simulator. In *IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, pp. 1–6. IEEE, 2012.

[19] Kazuhito Hyodo and Hiroaki Kobayashi. A study on tendon controlled wrist mechanism with nonlinear spring tensioner. *Journal of the robotics Society of Japan*, Vol. 11, No. 8, pp. 1244–1251, 1993.

[20] Bertrand Tondu and Pierre Lopez. Modeling and control of mckibben artificial muscle robot actuators. *IEEE control systems*, Vol. 20, No. 2, pp. 15–38, 2000.

[21] Ryuma Niiyama and Yasuo Kuniyoshi. A pneumatic biped with an artificial musculoskeletal system. In *Proc. International Symposium on Adaptive Motion of Animals and Machines*, pp. 80–81, 2008.

[22] Björn Verrelst, Ronald Van Ham, Bram Vanderborght, Frank Daerden, Dirk Lefeber, and Jimmy Vermeulen. The pneumatic biped "lucy" actuated with pleated pneumatic artificial muscles. *Autonomous Robots*, Vol. 18, No. 2, pp. 201–213, 2005.

[23] Koh Hosoda, Yuki Sakaguchi, Hitoshi Takayama, and Takashi Takuma. Pneumatic-driven jumping robot with anthropomorphic muscular skeleton structure. *Autonomous Robots*, Vol. 28, No. 3, pp. 307–316, 2010.

[24] Yoshihiro Nakata, Hiroshi Ishiguro, and Katsuhiro Hirata. Dynamic analysis method for electromagnetic artificial muscle actuator under pid control. *IEEJ Transactions on Industry Applications*, Vol. 131, No. 2, pp. 166–170, 2011.

[25] Yoshihiro Nakata, Atsuhiro Ide, Yutaka Nakamura, Katsuhiro Hirata, and Hiroshi Ishiguro. Hopping of a monopedal robot with a biarticular muscle driven by electromagnetic linear actuators. In *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3153–3160. IEEE, 2012.

[26] Frank Daerden and Dirk Lefeber. Pneumatic artificial muscles: actuators for robotics and automation. *European journal of mechanical and environmental engineering*, Vol. 47, No. 1, pp. 11–21, 2002.

[27] Ching Ping Chou and Blake Hannaford. Measurement and modeling of mckibben pneumatic artificial muscles. *IEEE Transactions on Robotics and Automation*, Vol. 12, No. 1, pp. 90–102, 1996.

[28] Ryuma Niiyama, Satoshi Nishikawa, and Yasuo Kuniyoshi. Biomechanical approach to open-loop bipedal running with a musculoskeletal athlete robot. *Advanced Robotics*, Vol. 26, No. 3-4, pp. 383–398, 2012.

[29] Vijaya Krishnamoorthy, MarkL Latash, JohnP Scholz, and VladimirM Zatsiorsky. Muscle synergies during shifts of the center of pressure by standing persons. *Experimental Brain Research*, Vol. 152, No. 3, pp. 281–292, 2003.

[30] Lena H Ting and Jane M Macpherson. A limited set of muscle synergies for force control during a postural task. *Journal of neurophysiology*, Vol. 93, No. 1, pp. 609–613, 2005.

[31] Vincent CK Cheung, Andrea d'Avella, Matthew C Tresch, and Emilio Bizzi. Central and sensory contributions to the activation and organization of muscle synergies during natural motor behaviors. *Journal of Neuroscience*, Vol. 25, No. 27, pp. 6419–6434, 2005.

[32] Hiroaki Hirai, Kazuhiro Matsui, Taiki Iimura, Keigo Mitsumori, and Fumio Miyazaki. Modular control of limb kinematics during human walking. In *IEEE RAS and EMBS International Conference on Biomedical Robotics and Biomechatronics (BioRob)*, pp. 716–721, 2010.

[33] Taiki Iimura, Keita Inoue, Hang TT Pham, Hiroaki Hirai, and Fumio Miyazaki. A preliminary experiment for transferring human motion to a musculoskeletal robot-decomposition of human running based on muscular coordination. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4496–4501, 2011.

[34] Eichi Watanabe, Takanori Oku, Hiroaki Hirai, Kanna Uno, Mitsunori Uemura, and Fumio Miyazaki. Exploiting invariant structure for controlling multiple muscles

in anthropomorphic legs: An inspiration from electromyography analysis of human pedaling. In *IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*, pp. 88–93, 2015.

[35] Toru Oshima, Tomohiko Fujikawa, Osamu Kameyama, and Minayori Kumamoto. Robotic analyses of output force distribution developed by human limbs. In *Proc. IEEE International Workshop on Robot and Human Interactive Communication*, pp. 229–234. IEEE, 2000.

[36] Ryuma Niiyama and Yasuo Kuniyoshi. Design principle based on maximum output force profile for a musculoskeletal robot. *Industrial Robot: An International Journal*, Vol. 37, No. 3, pp. 250–255, 2010.

[37] Minayori Kumamoto, Toru Oshima, and Tomohisa Yamamoto. Control properties induced by the existence of antagonistic pairs of bi-articular muscles—mechanical engineering model analyses. *Human Movement Science*, Vol. 13, No. 5, pp. 611–634, 1994.

[38] Marcus G Pandy, Felix E Zajac, Eunsup Sim, and William S Levine. An optimal control model for maximum-height human jumping. *Journal of biomechanics*, Vol. 23, No. 12, pp. 1185–1198, 1990.

[39] T Spägele, A Kistner, and A Gollhofer. Modelling, simulation and optimisation of a human vertical jump. *Journal of biomechanics*, Vol. 32, No. 5, pp. 521–530, 1999.

[40] Shinya Aoi and Tetsuro Funato. Neuromusculoskeletal models based on the muscle synergy hypothesis for the investigation of adaptive motor control in locomotion via sensory-motor coordination. *Neuroscience research*, Vol. 104, pp. 88–95, 2016.

[41] Ruta Desai and Hartmut Geyer. Muscle-reflex control of robust swing leg placement. In *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2169–2174. IEEE, 2013.

[42] Friedrich Paulsen and Jens Waschke. *Sobotta Atlas of Human Anatomy*, Vol. 1. Urban & Fischer, 15 edition, 2011.

[43] Dogan Ibrahim. *Microcontroller based applied digital control*. John Wiley & Sons, 2006.