

國立臺灣大學電機資訊學院電機工程學研究所



碩士論文

Graduate Institute of Electrical Engineering
College of Electrical Engineering and Computer Science

National Taiwan University

Master Thesis

基於深度學習實現運動預測控制上肢復健外骨骼機器人
Deep Learning based Motion Prediction for Exoskeleton Robot
Control in Upper Limb Rehabilitation

任嘉梁

Jia-Liang Ren

指導教授：傅立成 博士

Advisor: Li-Chen Fu, Ph.D.

中華民國 107 年 7 月

July 2018

誌謝



能完成這項研究承蒙許多人的幫助與鼓勵，首先想感謝我的指導教授 傅立成老師。在研究路上總是適時給予我許多的激勵及啟發，讓我能夠不斷地思考改進目前的問題，不斷挖掘自己的潛能讓，面對各式挑戰時永不放棄。同時也感謝口試委員賴金鑫老師、陸哲駒老師及盧璐老師所提供專業而具體的建議讓此研究更臻於完善，在此對所有的老師表示深摯的感謝。

在碩士生涯中，復健機器人醫療團隊成員的情誼將是值得且難忘的回憶。謝謝凱文學長與展翔多次提供即時的硬體架構的指教與協助，謝謝尚鶴學長與立宇學長在軟體開發上珍貴的經驗傳承，謝謝助理惠真姐在醫院行政上的協助，謝謝治療師啟倫提供充沛的臨床復健知識，謝謝研究夥伴力愷和緯軒的互相勉勵與共同成長，謝謝恩宇和雅慧與我切磋討論並分擔工作，期待你們在未來將有亮眼的表現。

謝謝 ACL 實驗室所有的夥伴們，謝謝俊緯學長、明理學長、安陞學長與皇志學長在我求學過程中不吝提供學術指導，謝謝助理懿萱、小寧、郁璇、詩詠在行政事務莫大的協助，謝謝一同努力的彥程、宗澤、子俊、侑寰、佐薪、孟浩、竣棠、秉蒼、柯文森與莫凱森的互相打氣和經驗交流，感謝各組學弟妹平日的關懷。也感謝更多未列名於此的朋友，你們的友情支援讓我的碩士生涯能順利的完成。

特別感謝我最親愛的家人一直以來的無私付出與包容，讓我能夠無後顧之憂的進行求學與研究，並讓我從低潮中再次振作起來。最後再次感謝所有關心著我的人。

任嘉梁 謹致於 2018 年 8 月

中文摘要



因中風導致神經麻痺是種很常見的疾病，臨床症狀常見為肌力不足、肌肉痙攣及無法自主控制關節活動等病徵。通過長期反覆性的復健治療後，能夠有效幫助患者恢復原有運動機能，並且可以防止發生二次併發症。以機器人輔助提供上肢復健，可以為患者提供更好的復健療程同時減少治療師的負擔。本研究利用慣性測量單元還原手臂運動的姿態，並結合肌電訊號來訓練深度學習模型，預測使用者手臂想要動作的位置(方向和速度)，達到機器人主動控制與引導控制。

於相關的文獻中，使用力/力矩感測器或肌電訊號的控制方式建立人機互動模型在多軸主動控制上有較高的難度。本研究提出的深度學習模型架構與傳統模型以及其他深度模型架構相比，擁有更高的準確率且對不同受試者的影響較小。此模型更是可以透過少量的數據對特殊的病人進行微調，以實現更好的結果。

本研究提出的方法經過三位健康受試者在線測試，並實現在上肢復健外骨骼機器人 NTUH-II 上。實驗結果顯示本研究提出的方法於復健任務中的表現優於相關的研究。此外，本方法能夠簡單的擴展為各種不同復健療程。

關鍵字：復健機器人、手臂姿態、肌電訊號偵測、機器學習、主動控制、引導控制、NTUH-II

ABSTRACT



Neural paralysis due to stroke is a common disease, and clinical symptoms are often characterized by insufficient muscle strength, muscle spasms and inability to control joint activity. Long-term repetitive rehabilitation treatment can effectively help patients to restore their original motor function and can prevent the secondary complications. Robot-assisted upper limb rehabilitation can provide patients better rehabilitative treatment while reducing the burden on the therapist. In this study, the inertial measurement unit is used to estimate arm dynamics and is combined with muscle electromyography to train deep learning model for human arm joint angles prediction. This model can be applied to the active control and guide control of the robot arm.

In the relevant literature, the use of force/torque sensors or myoelectric signals based control has a higher difficulty in establishing a human-robot interaction model for active rehabilitation. In this thesis, a learning model is proposed. Compared with the traditional model and other architecture of deep learning model, the proposed model in this study has a higher accuracy rate and has less impact on different subjects. This model can be fine-tuned to adapt special patients through a small amount of data to achieve better results.

The method proposed in this study was online tested by three healthy subjects and implemented on the upper limb rehabilitation exoskeleton robot NTUH-II. The experimental results show that it outperforms than relevant research works. In addition, the method can be simply extended to various rehabilitation therapies.

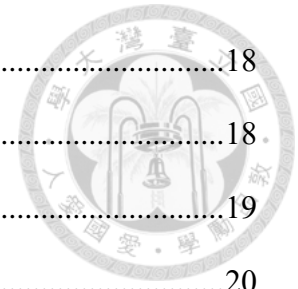
Keywords: rehabilitation robotics, arm dynamics, EMG sensing, machine learning, active control, guide control, NTUH-II

CONTENTS



口試委員會審定書	#
誌謝	i
中文摘要	ii
ABSTRACT	iii
CONTENTS	iv
TABLE OF ACRONYMS	vii
LIST OF FIGURES	ix
LIST OF TABLES	xii
Chapter 1 Introduction.....	1
1.1 Motivation.....	1
1.2 Literature Survey	4
1.3 Contribution.....	7
1.4 Thesis Organization	8
Chapter 2 System Overview and Preliminaries	9
2.1 Upper Limb Rehabilitation Robot NTUH-II	9
2.1.1 Mechanical Structure	9
2.1.2 Software	12
2.1.3 Safety Issue	13
2.2 IMU and EMG Instrument	15
2.3 Therapeutic Exercises	16
2.3.1 Active Mode.....	16
2.3.2 Guide Mode.....	17
2.4 Complementary Filter.....	17

2.5	Traditional Regression Model	18
2.5.1	Support Vector Regression	18
2.5.2	K-Nearest Neighbor Regression	19
2.6	Deep Learning Model	20
2.6.1	Convolutional Neural Network	20
2.6.2	Recurrent Neural Network	21
2.6.3	Convolutional LSTM	23
Chapter 3	Design Motion Prediction based Control System.....	26
3.1	Estimate Human Arm Dynamics and Muscle Activity.....	27
3.1.1	IMU subsystem and signal pre-processing	27
3.1.2	Human Arm Angle Calibration	30
3.1.3	EMG Subsystem and Signal Pre-processing.....	33
3.1.4	Data Acquisition.....	35
3.2	Motion Prediction Regression Model.....	38
3.2.1	One Stream LSTM Model.....	38
3.2.2	Multi-stream LSTM Dueling Model.....	39
3.3	Fine Tune of the Model	42
3.4	Control of Robot System	44
3.4.1	Guide Mode.....	45
3.4.2	Active Mode.....	47
Chapter 4	Experimental and Results	48
4.1	Model Training	48
4.2	Offline Evaluation Indexes and Result	49
4.3	Real-time Experiment Protocol	55
4.4	Real-time Experiment Result.....	58



4.4.1	Performance for Bilateral Mode Exercise.....	58
4.4.2	Performance for Lead Mode Exercise.....	64
4.4.3	Performance for Active Mode Exercise	71
Chapter 5	Conclusion	74
REFERENCE		76

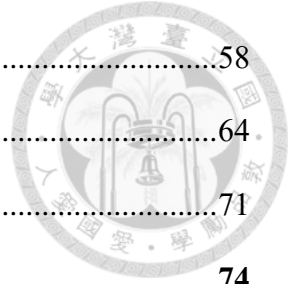
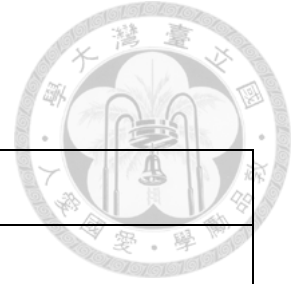
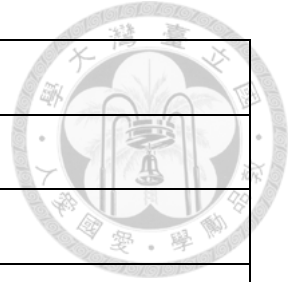


TABLE OF ACRONYMS



abd./add.	abduction/adduction
ADL	activities of daily living
ANN	artificial neural network
CNN	convolutional neural network
ConvLSTM	convolutional long-short time memory
DOFs	degrees of freedom
EMG	electromyography
ext./int.	external/internal
flex./ext.	flexion/extension
F/T sensor	force/torque sensor
GH	glenohumeral
GUI	graphical user interface
IMU	inertial measurement unit
k-NN	k-nearest neighbor
LSTM	long-short time memory
MAE	mean absolute error
MPRM	motion prediction regression model
MS-LSTM	Multi-stream long-short time memory
NRMSE	normalized root-mean-square error
NTUH-ARM	National Taiwan University Hospital Arm
NTUH-II	National Taiwan University Hospital II
RBF	radial basis function

RNN	recurrent neural network
ROM	range of motion
STFT	short-time Fourier transform
SVR	support vector regression
VR	virtual reality

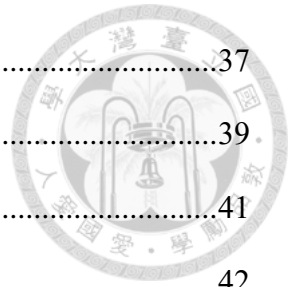


LIST OF FIGURES

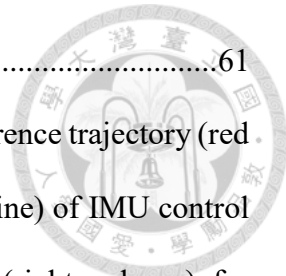


Fig. 2.1 The mechanical structure of NTUH-II.	10
Fig. 2.2 The flowchart of the human-machine interface.	13
Fig. 2.3 The emergency stop buttons and a convenient handy pause button.....	14
Fig. 2.4 Myo armband (left) and USB receiver (right).....	16
Fig. 2.5 Basic complementary filter.....	18
Fig. 2.6 (A) An example input volume in red and example volume of neurons in the first convolutional layer. (B) computation process in one neural.	21
Fig. 2.7 Basic RNN architecture and unfold like.	22
Fig. 2.8 LSTM cell contains four interacting layers.....	22
Fig. 2.9 Inner structure of ConvLSTM.....	25
Fig. 3.1 Placement of two Myo armbands (left arm).....	28
Fig. 3.2 The architecture of complementary filter which can complement each one of the shortcomings of the signal. Through the complementary filter, we can get the accuracy of the object's rotation angle and angular velocity.....	28
Fig. 3.3 Two Myo armbands' principal axes.	30
Fig. 3.4 Example of an inaccurate sensor measurement caused by muscle deformation.....	30
Fig. 3.5 Result of muscle calibration (A) shoulder flex./ext., (B) horizontal abd./add., (C) ext./int. rotation, (D) elbow flex./ext.	32
Fig. 3.6 (A) Example of a triangular window for moving average. (B) Example of the Hamming window.....	35
Fig. 3.7 Flow of signal processing.....	35

Fig. 3.8 GUI for our application.	37
Fig. 3.9 One Stream LSTM model.	39
Fig. 3.10 Multi-stream LSTM model.	41
Fig. 3.11 Multi-stream LSTM dueling model.	42
Fig. 3.12 Two Stream LSTM model fine-tune architecture.....	44
Fig. 3.13 Control diagram for guide mode and active mode.....	45
Fig. 3.14 (A) In lead mode, two Myo armbands are worn on the therapist’s arm with the same side of patients impaired arm. (B) In bilateral mode, two Myo armbands are worn on the patient’s healthy arm.....	46
Fig. 3.15 Control diagram without MPRM for guide mode.....	47
Fig. 3.16 In active mode, two Myo armbands are worn on the patient’s impaired arm..	47
Fig. 4.1 Predict trajectory of four joint angle based on MS-LSTM Dueling model.	50
Fig. 4.2 Predict trajectory of four joint angle based on SVR model.	51
Fig. 4.3 Compare hand position is 3D space reconstruct from arm joint angles of the different model.....	52
Fig. 4.4 Boxplot of (A) different models and (B) different training data volumes for all subjects’ result distribution.	54
Fig. 4.5 Desired trajectory for subjects	56
Fig. 4.6 Real arm angle (green dashed line), predict joint angle as reference trajectory (red solid line) joint angle and robot joint angle (black solid line) of IMU control (left column) and MS-LSTM Dueling model control (right column) for single-joint tasks in bilateral rehabilitation.....	59
Fig. 4.7 Real arm angle (green dashed line), predict joint angle as reference trajectory (red solid line) joint angle and robot joint angle (black solid line) of IMU control (left column) and MS-LSTM Dueling model control (right column) for	



feeding (two-joint) task in bilateral mode.	61
Fig. 4.8 Real arm angle (green dashed line), predict joint angle as reference trajectory (red solid line) joint angle and robot joint angle (black solid line) of IMU control (left column) and MS-LSTM Dueling model control (right column) for greeting (four-joint) task in bilateral mode.....	63
Fig. 4.9 Real arm angle (green dashed line), predict joint angle as reference trajectory (red solid line) joint angle and robot joint angle (black solid line) of IMU control (left column) and MS-LSTM Dueling model control (right column) for single-joint tasks in lead mode.	66
Fig. 4.10 Real arm angle (green dashed line), predict joint angle as reference trajectory (red solid line) joint angle and robot joint angle (black solid line) of IMU control (left column) and MS-LSTM Dueling model control (right column) for feeding (two-joint) task in lead mode.	67
Fig. 4.11 Real arm angle (green dashed line), predict joint angle as reference trajectory (red solid line) joint angle and robot joint angle (black solid line) of IMU control (left column) and MS-LSTM Dueling model control (right column) for greeting (four-joint) task in lead mode.	69
Fig. 4.12 End-point interactive force for single-joint tasks in active mode.	71
Fig. 4.13 End-point interactive force for multi-joint tasks (feeding and greeting) in active mode.	71



LIST OF TABLES

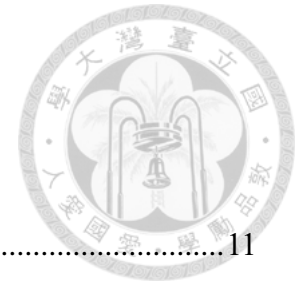
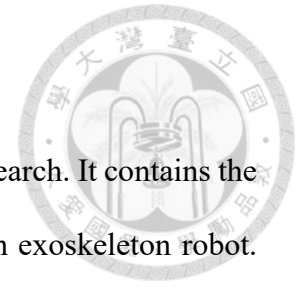


TABLE 2.1 Denavit–Hartenberg parameters of NTUH-II.	11
TABLE 2.2 Introduction and purpose for each joint of NTUH-II.....	12
TABLE 3.1 Measurement error test on five subjects with muscle compensate.	32
TABLE 3.2 Measurement error test on five subjects without muscle compensate.	32
TABLE 4.1 Performance of different models.....	53
TABLE 4.2 Total mean absolute error (degrees) of different models in single-joint.....	53
TABLE 4.3 Performance of MS-LSTM Dueling model trained on different training data volumes.....	54
TABLE 4.4 Total mean absolute error (degrees) in single-joint.....	54
TABLE 4.5 Experiment protocol of single-joint and multi-joint tasks	56
TABLE 4.6 Evaluation results of single-joint tasks in bilateral mode.	60
TABLE 4.7 Evaluation results of feeding task in bilateral mode.	62
TABLE 4.8 Evaluation results of greeting task in bilateral mode.....	64
TABLE 4.9 Evaluation results of single-joint tasks in lead mode.....	66
TABLE 4.10 Evaluation results of feeding task in lead mode.	68
TABLE 4.11 Evaluation results of greeting task in lead mode.	70
TABLE 4.12 Evaluation results of single-joint tasks in active mode.....	73
TABLE 4.13 Evaluation results of multi-joint tasks in active mode.....	73

Chapter 1 Introduction

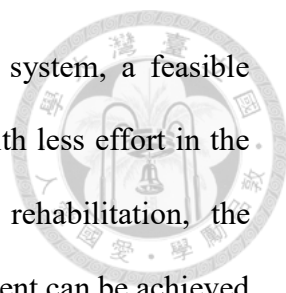


In this chapter, first, we will introduce the motivation of this research. It contains the background of upper limb rehabilitation therapies and rehabilitation exoskeleton robot. Then, we will discuss the related state-of-the-art studies in rehabilitation robotics in the section of literature survey, which is followed by the contribution of this research. The last section so this chapter shows the organization of this thesis.

1.1 Motivation

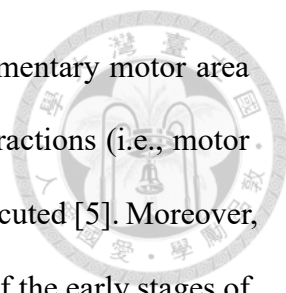
In some research, statistics show that stroke is one of the common causes of severing disability and may cause total paralysis or half paralysis of the upper limb [1]. In the United State, from 2003 to 2013, each year about 795,000 people continue to experience a new or recurrent stroke (ischemic or hemorrhagic). Approximately 610,000 of these are first events and 185,000 are recurrent stroke events [2]. Disability of the upper limb caused by neurological or orthopedic clinically show characteristics of inadequate muscular strength, altered muscle group firing pattern or inability to voluntarily control the joint which may lead to worse conditions such as pain, stiffness or shoulder impingement syndrome [3]. Clinical studies have shown that long-term repetitive rehabilitation can help these patients regain their motor function and prevent the occurrence of complications.

Traditionally, rehabilitation programs require therapists to help the patient perform repetitive and time-consuming movements [4]. However, with the trend of population aging, current medical service system is unable to meet the needs of every patient who needs it because of lack of human resources. Since each assistance requires a considerable amount of time and effort from the therapist, it is not possible to provide high quality and stable remedial exercises support by the therapy anytime, anywhere.



In order to overcome these problems in the medical service system, a feasible method is to involve robot-assisted devices to help the therapists with less effort in the rehabilitation program. With the intervention of robot-assisted rehabilitation, the requirement of offering high quality and stable training for every patient can be achieved because of the stability and accuracy of the robot system. The robotic device can not only provide the passive and active range of motion (ROM) exercise but also can record the biomedical and kinetic measurements online through bio-electrical signals, inertial measurement unit and kinetic measurements sensors in the rehabilitation program. These measurements can be used to further assess the improvement of the patient's motor function by the therapists or doctors. In addition, in order to motivate patients to more enthusiastically carry out rehabilitation programs, robotic systems can be combined with virtual reality (VR) technology, which can transform the dull exercise into amusement games and simultaneously display the patient's movement performance on the screen, which can speed up the process of functional recovery. Moreover, the performance of the patient in the VR game can also be used as another index for assessing the patient's motor function.

The robot-assisted rehabilitation therapeutic exercises basically can be divided into 4 types: passive, active, active-assistive, and active-resistive. Depending on the condition of motor impairment, therapists or doctors will recommend the appropriate type of rehabilitation exercise to the patient. In the passive exercise, the patient's affected side of the arm is guided by the robot arm to move, the common practice is doing repetitive motions along a predetermined trajectory. In clinical practice, another common treatment is bilateral rehabilitation, which can categorize as a special passive exercise. The patient's impaired arm is driven by the machine to do the same exercise as the healthy side arm. Specifically, voluntary movements of the intact limb may facilitate voluntary movements



in the paretic limb. Activating the primary motor cortex and supplementary motor area for the intact limb increases the likelihood of voluntary muscle contractions (i.e., motor synergies) in the impaired limb when symmetrical movements are executed [5]. Moreover, bilateral training is proven to be more effective in the rehabilitation of the early stages of stroke [6]. The other 3 types of exercise require the exoskeleton robot to follow the patient's movement and then provide suitable human-robot cognitive interaction. In order to achieve that, the robot system needs to get the motion intention of the impaired arm or contralateral arm and then follow the corresponding movement. However, the extraction of motion intention is a tricky and time-consuming problem in the related research of exoskeleton robot.

As we live in a big data era, transforming big data into valuable knowledge becomes much more important than ever [7]. Machine learning has been one of the most widely used methods in order to extract knowledge from a large amount of data in bioinformatics. The machine learning algorithm uses training data to reveal the underlying pattern, builds the model, and then make predictions on the new data based on the model. Conventional machine learning algorithms have limitations in dealing with raw data forms. So researchers have spent a lot of efforts to translate the original form into a suitable high abstract level feature with considerable field expertise [8]. On the other hand, deep learning is a new machine learning algorithm, which recently appeared in the capacity of big data and has overcome the former limitations, parallel and distributed computing, and sophisticated algorithms. But deep learning algorithms also shows that hard to converge and easy overfitting problems. In this thesis, we propose a new deep learning structure to decode human bio-signals to identify motion intentions. The proposed model can predict angles of the human arm, which can be used as a reference trajectory for any robotic arm to achieve human-machine synchronous movement.



1.2 Literature Survey

According to the clinical and research findings, a current prominent rehabilitation technique is bilateral movement training. This protocol applies sound neurological interlimb coordination postulates in activating motor synergies between limbs [9]. Training patients with two-handed tasks improve the efficiency of arm movements on the impaired side [10] with changes accompanied by a reorganization of brain mappings on the affected hemisphere. Bilateral tasks require to operate the two arms together so that they cooperate to accomplish the aimed function. Evidences indicate that the simultaneous movement of both limbs helps the neuro-muscular system to regain some stability and improve usage of the impaired limb [11].

Robot-assisted therapy is able to provide high intensive and accurate movement and can lower the demand for therapists. In order to track the subject's movement in the real world, the rehabilitation robot system is equipped with sensors to measure the user's motion intention data from which we can extract human motion intention. There are several kinds of rehabilitation upper limb robot have been developed [12]. To date, in the field of rehabilitation robotics, the representative types of the subject's movement intent on the upper limb include force/torque sensor (F/T sensor), inertial motion unit (IMU) and electromyography (EMG).

F/T sensor is a mechanical technique mounted on exoskeleton robot. It is used to measure the interactive force and torque between the user and the robot. This type of control strategy requires a dynamic model to compute the joint force or kinematics that represents the human motion intention. The model is formed as a combination of inertia, gravitation, Coriolis and centrifugal effects. Here are some related works such as [13], [14], [15] which address active control for upper limb rehabilitation robotics using F/T

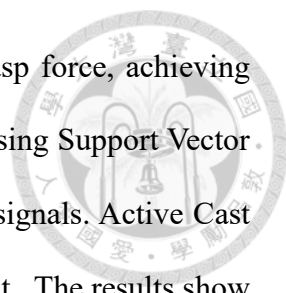
sensor.

ETS-MARSE [13] map human intention by measured value from F/T sensor, but human joint torque is highly nonlinear and is controlled based on mathematical model, which causes the system to slightly vibrate. Artificial neural network (ANN) based on radial basis function (RBF) to solve the mapping problem has been used in [14], but it needs a long time training. In our previous work [15], a mathematical model based on active control along with a three-stage gravity compensation and reactive motion rescheduling model were proposed to improve the intention mapping problem and implement on NTUH-II.

It is unreasonable to measure the force at a specific position of the arm using an F/T sensor, because human arms has multiple degrees of freedom, and the force point is multiple. Also, the interactive force and torque value of users are different, which results in that the patterns exerted by different subjects in the same task are not the same. Hence, active control based on the dynamic model is more difficult to establish when we want to extract motion intention information from the measured interaction force and torque. Moreover, if the user's muscle strength is insufficient to exert the interactive force and torque, the F/T sensor cannot obtain the intention information.

Another sensor that can be used to detect the motion intention of a human arm is the EMG sensor which detects the bioelectrical signal directly. There are some works use EMG to measure the contralateral hand's motion intention to achieve single joint bilateral rehabilitation [16], [17], or to measure the patient impaired arm EMG signal to achieve the single joint active rehabilitation task [18], for example, detect biceps and triceps EMG pattern for elbow rehabilitation task. Besides, other works [19], [20] use multi-channel EMG signals to estimate user's arm position or joint angle with different types of model.

Bravo [16] is a hand exoskeleton which can measure the EMG signals in extensor



digitorum, flexor digitorum and adductor pollicis to control the grasp force, achieving hand bilateral rehabilitation task. WEP [17] control wrist flex./ext. using Support Vector Machine (SVM) to classify different force intensity from the EMG signals. Active Cast [18] use frequency analysis of EMG signals to control elbow flex./ext.. The results show that the classifier is better than the regression model to classify the force into different value categories.

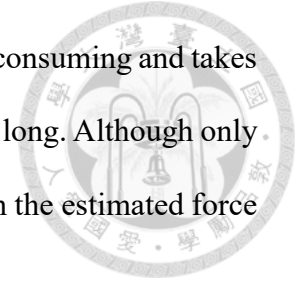
Support Vector Regression (SVR) has been used to detect human arm motion position [19] and implement on teleoperate DLR Light-Weight Robot III. The same technique can also be used in the bilateral task. A musculoskeletal model has been used on BOTAS [20] that can estimate the joint angles of elbow and wrist. Their model assumes that each muscle is modeled as a property of a linear spring. Then, the joint angle is estimated under the assumption that the interaction force is zero.

Although EMG can represent the pure intentions of humans, there are some limitations due to the complexity of the musculoskeletal system. When the motion pattern is only slightly different, it might cause a huge change in EMG signals. In order to more accurately decode human multi-joint intention, the inertial measurement unit (IMU) sensor is also used. There are some works [21], [22] only using IMU or combine IMU and EMG to decode human arm movement intention.

In our previous work [21], NTUH-ARM uses IMU sensor to get human arm joint angle as a desired control input. The robot can achieve mirror therapy and can be controlled over a larger range of motion. Another work [22] combines EMG and IMU signals through ANN to predict elbow and forearm angles and implement on VR environment. The result shows that if only estimating current joint angles, we can get stable results. But there will be a large error in predicting.

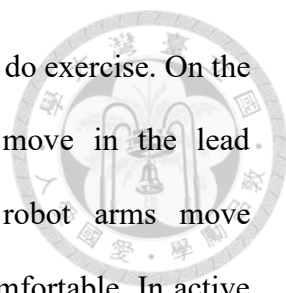
Consequently, most of the works related to EMG or EMG combined with IMU need

to train the model when a user starts to perform the task. This is time-consuming and takes a lot of effort due to the time for collecting data and training model is long. Although only one joint motion is considered, there is still a large estimation error in the estimated force or angle.



1.3 Contribution

This research will review the National Taiwan University Hospital-II (NTUH-II), a 7 DOF exoskeleton upper arm rehabilitation robot, which has been approved clinical testing by the Department of Health IRB and National Taiwan University Hospital IRB. NTUH-II can individually control elbow and shoulder or simultaneously. First, a complementary filter is designed to accurately and stably estimate human arm joint angles and angular velocities. Second, we found that the muscular deformation of the arm causes the measurement error at large angles, so we design muscle compensation method to get more accurate and stable results. Third, a deep learning method based Multi-stream LSTM Dueling (MS-LSTM Dueling) model is proposed to predict the human arm trajectory. The mode inputs are filtered IMU signals and EMG features we extracted from pre-processed EMG signal using time-frequency analysis. Compared to the traditional regression model or other architecture of the deep learning model, MS-LSTM Dueling model can more accurately predict the human arm trajectory. Fourth, we design a fine-tune method to let pre-trained model get a better result by using a small training data from a special user. Fifth, by using the predicted trajectory, the robotic arm can be coordinated with the human arm. Thus, the rehabilitation robot can implement the treatment exercises of active rehabilitation and guide rehabilitation. For the active rehabilitation, robot will follow subject's volitional movement. The guide rehabilitation can be divided into two parts, one is bilateral rehabilitation, and the other is lead rehabilitation. During bilateral



rehabilitation, patient's health arm will guide his/her impaired arm to do exercise. On the contrary, therapist's arm will guide patient's impaired arm to move in the lead. rehabilitation. The MS-LSTM Dueling model will guarantee robot arms move simultaneously with no delay in guide mode, so the patient feels comfortable. In active rehabilitation, the reference trajectory generated by MS-LSTM Dueling mode can make robot follow the user arm motion.

1.4 Thesis Organization

In this thesis, we will concentrate on the development of active therapies on upper limb exoskeleton rehabilitation robot, NTUH-II. The organization of this thesis is listed as follows.

Chapter 1 has introduced the motivation, contribution of this research, and a general description of state-of-the-art studies in human motion intention extraction by both F/T sensor and EMG.

In Chapter 2, a detailed description of the mechanical structure of the upper limb rehabilitation exoskeleton robot, NTUH-II, IMU, EMG and related devices will be given. Next, we will give an overview of some preliminary theories behind this research and categories of robot stroke rehabilitation exercise.

Chapter 3 is the core of this research, where we elaborate the details of theory step by step. First, the sensor device and signal pre-processing of IMU and EMG are introduced. The architecture of the motion prediction regression model and the control strategy are also addressed in this chapter.

Chapter 4 shows the experimental results implemented on NTUH-II based on the proposed theory. The effectiveness of the proposed method is verified in the experiments.

Finally, the conclusions of this research are shown in Chapter 5.

Chapter 2 System Overview and Preliminaries

This chapter introduces the hardware and software of the rehabilitation robot system. And then, the external sensor of IMU and EMG and the general categories of robot-assisted rehabilitation exercise will be introduced. The last part of this chapter will show traditional regression algorithm and the state-of-the-art deep learning method.

2.1 Upper Limb Rehabilitation Robot NTUH-II

In this section, an overview of the newly developed upper limb exoskeleton rehabilitation robot NTUH-II is presented.

Compared with our previous designed rehabilitation robot NTUH-ARM and other rehabilitation robots Hocoma, NTUH-II [23] contains more intuitive human-robot structure with 8 degrees of freedom (DOF) and also possesses larger ROM which approaches human-like ROM. Thus, all kinds of the training program can be implemented for a patient with orthopedic or neurologic motor function disorders on either left or right side of the upper limb. Two F/T sensors install on NTUH-II and external IMU and EMG devices allow this robot system to assist patient with not only active mode but also guide mode of rehabilitation exercise.

2.1.1 Mechanical Structure

In order to provide more rehabilitation treatment for upper limb dyskinesia patients, NTUH-II developed more subtly by combining the previous experience of NTUH-ARM. The main upgrade of NTUH-II contains the following points 1). Shoulder joint have larger ROM and sufficient DOF to fit the human activity angle, 2). Intuitive human-robot joint mapping relationship, and 3). easy to switch the left and right arm setting (*i.e.*, it can provide therapy for either left or right arm).

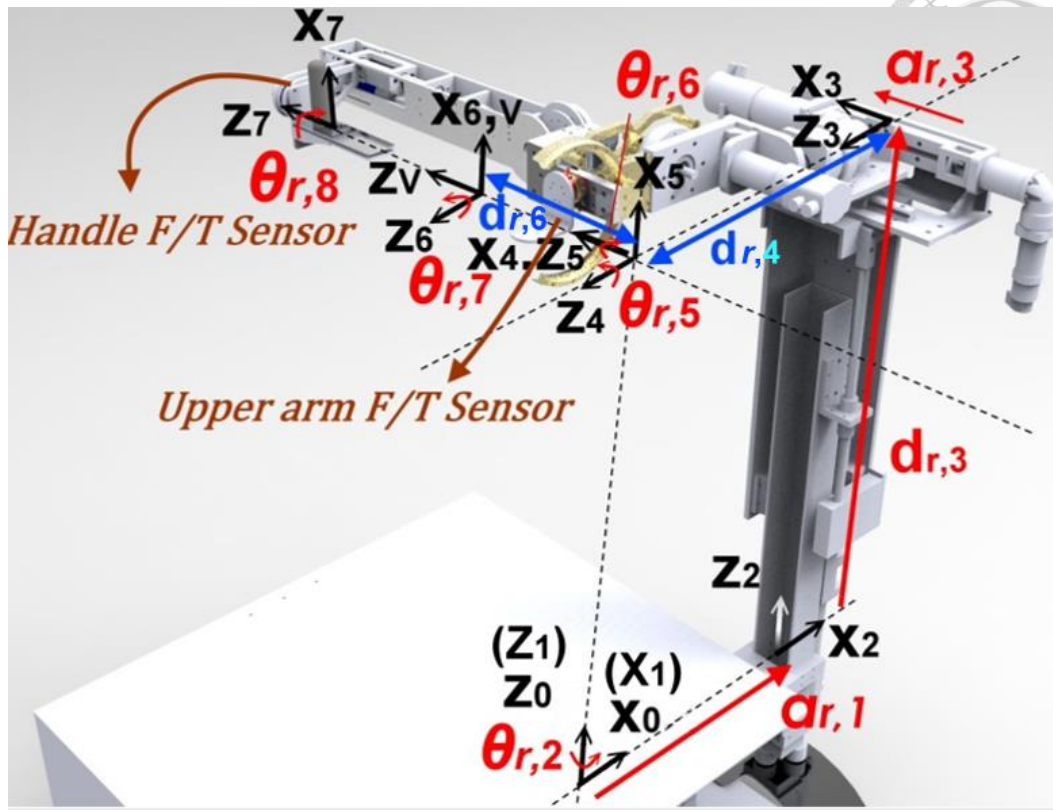


Fig. 2.1 The mechanical structure of NTUH-II.

NTUH-II is an 8-DOF exoskeleton-type robot arm which contains 8 electrical actuators and 2 ATI 6-axis F/T sensors. The mechanical structure of NTUH-II is shown in Fig. 2.1, where the variables X_i and Z_i ($i=0,1,\dots,V,\dots,7$) describe the coordinate of joint i , additionally, the joint v is a virtual joint for the purpose of building Denavit–Hartenberg (D-H) model, the variables $\theta_{r,2}, \theta_{r,5}, \theta_{r,6}, \theta_{r,7}$ and $\theta_{r,8}$ represent the 5 active revolute joints, and the rest of variables $a_{r,1}, d_{r,3}$ and $a_{r,3}$ represent the 3 active prismatic joints on robot NTUH-II. The above subscript r means robot arm. This figure also illustrated the locations of two F/T sensors mounted on NTUH-II, the first is placed at the upper arm and the second is at the hand grip. The detailed D-H symbolic parameters of the robot kinematics are list in TABLE 2.1

TABLE 2.1 Denavit–Hartenberg parameters of NTUH-II.

Axis	$\theta_{r,i}$	$d_{r,i}$	$a_{r,i}$	$\alpha_{r,i}$	Home
1	0	$d_{r,1}$	$a_{r,1}$	0	0
2	$\theta_{r,2}$	0	$a_{r,2}$	0	0
3	0	$d_{r,3}$	$a_{r,3}$	$-\pi/2$	$\pi/2$
4	0	$d_{r,4}$	0	0	0
5	$\theta_{r,5}$	0	0	$-\pi/2$	$-\pi/2$
6	$\theta_{r,6}$	$d_{r,6}$	0	$\pi/2$	0
V	$\theta_{r,7}$	0	0	$-\pi/2$	0
7	0	$d_{r,7}$	0	0	0
8	$\theta_{r,8}$	$d_{r,8}$	0	0	0

Among these 8 active DOFs, there are 2 prismatic DOFs utilized to adjust the position of robot's shoulder joint to match human's glenohumeral (GH) joint, which causes a critical issue during shoulder motion [24-26]. Moreover, $a_{r,1}$ corresponds to horizontal direction and $d_{r,3}$ corresponds to vertical direction. There are 4 DOFs related to shoulder movements, which are $\theta_{r,2}$, for horizontal abduction/adduction, $\theta_{r,5}$, for shoulder flexion/extension (or abduction/adduction if it is used with $\theta_{r,2}$), $\theta_{r,6}$, for external/internal rotation, and $a_{r,3}$, for traction. The remaining 2 DOFs for elbow and wrist motion include $\theta_{r,7}$ corresponding to elbow flexion/extension, and $\theta_{r,8}$, related to wrist pronation/supination.

Two ATI force/torque sensors, installed on the upper arm and hand grip, respectively, are used to measure the applied forces from the subjects. The function specification,

ROMs of NTUH-II and of human[27], and maximum allowable torque in continuous operation of respective joints are shown in TABLE 2.2.



TABLE 2.2 Introduction and purpose for each joint of NTUH-II.

Joint	DOF	Function	NTUH-II ROM	Human ROM	Max Torque (Nm)
1	$a_{r,1}$	Left and Right Adjustment	66 (cm)	-	127.79
2	$\theta_{r,2}$	Horizontal Abduction/Adduction	140 / 20°	90 / 45°	57.92
3	$d_{r,3}$	Up and Down Adjustment	52 (cm)	-	40.91
4	$a_{r,3}$	Forward and Backward Adjustment (or Traction)	20 (cm)	-	18.4
5	$\theta_{r,5}$	Shoulder Flexion/Extension	180 / 0°	180 / 60°	108.64
		Shoulder Abduction/Adduction	180 / 0°	180 / 45°	108.64
6	$\theta_{r,6}$	Shoulder Internal/External Rotation	70 / 90°	70 / 90°	57.92
7	$\theta_{r,7}$	Elbow Flexion/Extension	120 / 0°	150 / 0°	103.04
8	$\theta_{r,8}$	Wrist Pronation/Supination	90 / 90°	80 / 80°	15.36

2.1.2 Software

The control strategy and graphical user interface (GUI) of this machine are developed using LabVIEW 2011 on the PC side. The flowchart showed in Fig. 2.2 is used in the rehabilitation process

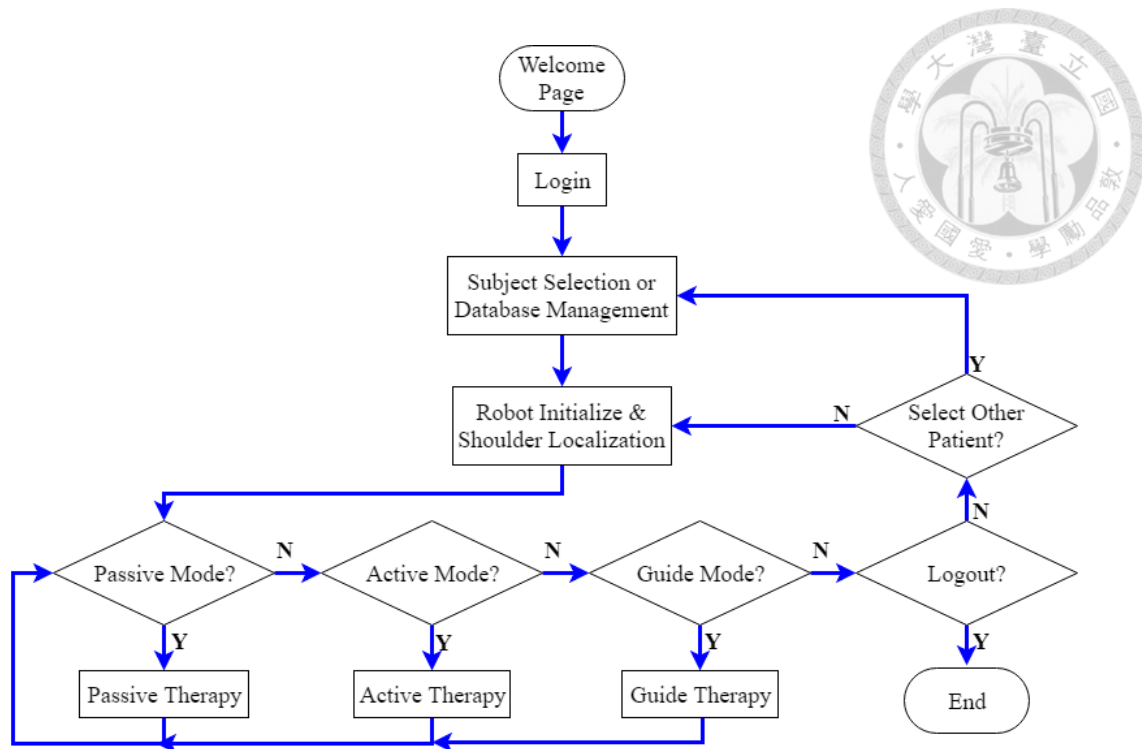


Fig. 2.2 The flowchart of the human-machine interface.

At the beginning of the treatment, when the welcome page is ready for the user to log in to therapist's or engineer's account, the login page will pop up. The next page allows the therapist to select the patient from the patient database. In this step, the system also provides a rich interface, such as creating/modifying a patient's information, searching for information, or recording data for a certain subject. After selecting the subject, the therapist can adjust the robot configuration so that the robot shoulder rotation center is aligned with the patient's GH joint, and the initial shoulder flexion angle is set to make the patient feel comfortable. Once the above settings are complete, the therapist can choose a different mode of treatment for the patient.

2.1.3 Safety Issue

Safety is one of the most important problems to be concerned in the process of rehabilitation. In robot-assisted rehabilitation, the patient's limb is supported by a robot rather than therapists, so it is important to establish a system to ensure safe operation

during the entire movement, including emergency human intervention

During the robot movement, the emergency stop button on the power cabinet and the convenient handy emergency stop button as shown in Fig. 2.3(a) are available for the therapist to design for emergency use. The two emergency stop buttons are connected to the power supply, and when one of the buttons is pressed, the motor closes immediately, then triggers the permanent magnet brakes to stop each movement, including a natural descent caused by gravity. When any danger or discomfort occurs, the therapist can use the emergency stop button to stop the action. In addition, a convenient handy pause button as shown in Fig. 2.3(b) is prepared for the patient. If the patient feels uncomfortable, it can be pressed to pause all movements at any time

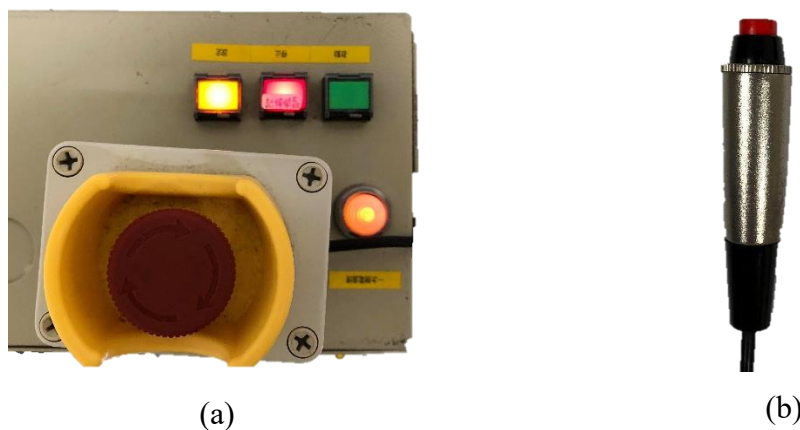


Fig. 2.3 The emergency stop buttons and a convenient handy pause button.

In addition to the safety of the hardware design, we also embed a variety of security mechanisms and virtual stop/pause buttons in the control strategy and GUI. The safety system will always monitor the joint position, angular velocity and current of each motor during the movement. Once an anomaly occurs or any of the above measurements exceed its security threshold, the movement will stop immediately and the robot configuration will be automatically locked by the motor brakes. There is also a virtual button in the GUI that commands the motor to move along the direction so that the robot's joints move down

the position immediately, rather than pause the motor. When the patient feels uncomfortable, the function helps to release the patient's arm. All of these hardware and software safeguards provide a more secure rehabilitation exercise.



2.2 IMU and EMG Instrument

In order to measure the kinematics and muscles activation of human upper limb, we use 2 Myo armbands to acquire IMU and EMG signals. The details of the device introduction will be presented below.

Myo armband (Fig. 2.4 left) is a wireless sensor including 8 EMG channels and 9-axis IMU sensor (3-axis gyroscopes, 3-axis accelerometers, and 3-axis magnetometers) which is manufactured by Thalmic Labs. The data measured by Myo armbands will be transmitted to the computer via Bluetooth or USB receiver (Fig. 2.4 right). The sampling rates of the Myo armband are 50 Hz for IMU and 200 Hz for EMG. Myo armbands can be worn on the human upper arm and forearm to measure the arm position, velocity, acceleration, and muscles activity. With this device, we can reduce plenty of cables that have the possibility of dragging and inconvenient when performing exercises in the exoskeleton robot. Thalmic Labs also provides software development toolkits (SDK) for C++ language. We reused it implement on python language and made a GUI for easy to use. The data communicating between Myo armband (python) and NTUH-II control program (LabVIEW) is through web socket (UPD or TCP).



Fig. 2.4 Myo armband (left) and USB receiver (right).

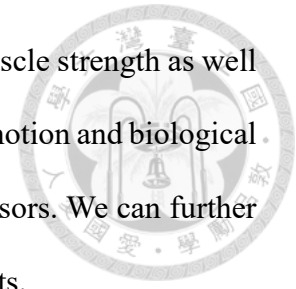
2.3 Therapeutic Exercises

Basically, the therapeutic exercises of robot-assisted rehabilitation can be categorized into 4 types, namely, passive mode, active mode, active-assistive mode, and active-resistive mode of treatment. All these modes of treatment are designed with patterns similar to those conventional rehabilitation programs for both neurologic and orthopedic motor impairments. The doctor or physical therapist can suggest a suitable mode of therapeutic exercise for a patient according to different condition or stage of motor impairment. Inspired by the bilateral model of traditional rehabilitation therapy, we set up a new robot-assisted mode named guided control. The detail of the guide mode will show below.

2.3.1 Active Mode

Patients with mild disability (Brunnstrom stage in high motor level) and enough muscle strength in the affected arm (Manual Muscle Testing \approx grade 3) are suitable for active mode. In active mode therapy, patients perform a full range of exercise exercises by voluntarily moving the injured limb without extra help from the exoskeleton robot. In other words, the patient moves freely and the robot is placed to follow the patient's movements. The main purpose of active therapy is to enhance motor learning (through a series of processes related to repetitive exercise leading to long-term changes in motor

ability) and further induce brain lesions to rearrange and improve muscle strength as well as impaired arm flexibility. Furthermore, during active exercise, the motion and biological data of patients' impaired arm can be measured and recorded by sensors. We can further use this data to evaluate and analysis of motor function improvements.



2.3.2 Guide Mode

The guide mode exercise is suitable for the patient with lower motor stages (Brunnstrom stage), resulting in limited limb or restricted limb. During exercise, patients' impaired limb will be fully supported by the rehabilitation robot to complete the exercise. This mode can be used as bilateral exercise or lead exercise. The difference between bilateral exercise and lead exercise is that bilateral training is suitable for patients with semi-paralysis, the robotic arm will do the same mirror action as the healthy arm, while lead training is suitable for patients both arms are impaired, the robotic arm will do the motion as same as therapist's health arm motion. The goal of this mode is to prevent secondary complications due to fixation (such as joint degeneration, joint contracture, muscle atrophy, etc.) caused by stroke. In addition, rehabilitation robot training is performed in this guide mode to induce neural facilitation or cortical activation, and to contribute to a higher level of recovery.

2.4 Complementary Filter

In order to get accuracy and stable kinematics signal, we need a good filter to filter signals. A simple estimating technique that is often used in the flight control industry to combine measurements is the complementary filter [28]. This filter is usually designed without mentioning the Wiener or Kalman filter, although it is related to them. Fig. 2.5 shows the basic complementary filter where x and y are noise measurements of some signal z and \hat{z} is the estimate of z produced by the filter. Assume that the noise in x is

mostly low frequency and the noise in y is mostly high frequency. Then $G(s)$ can be made a high-pass filter to filter out the low frequency noise in x . If $G(s)$ is high-pass, then $[1 - G(s)]$ is the complement, i.e., a low-pass filter which filter out the low-frequency noise in y . No detailed description of the noise processes is considered in complementary filtering.

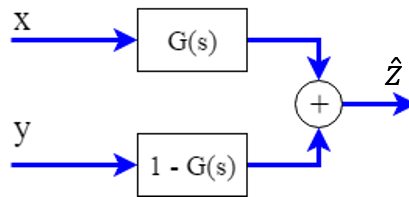


Fig. 2.5 Basic complementary filter.

2.5 Traditional Regression Model

2.5.1 Support Vector Regression

Support Vector Machine (SVM) [29] can be used to deal with classification problems in addition to regression problems. The so-called regression refers to each entity corresponding to the instance's label is a continuous real number, rather than discrete different categories (in SVM is often represented by integers). SVM, which deals with regression problems, is called Support Vector Regression (SVR).

Similar to SVM, the goal of SVR is to find the optimum hyperplane in space. Unlike SVM looking for a hyperplane that divides data into one, while the SVR is looking for a hyperplane that can accurately predict the distribution of data. Suppose the training data is represented as $(x_1, y_1), \dots, (x_n, y_n) \in \mathbb{R}^d \times \mathbb{R}$, where x is input attributes, y is the target value, \mathbb{R}^d denotes the space of the input patterns. In $\mathcal{E} - SV$ regression, the goal has been to find a function $f(x)$ that has at most \mathcal{E} deviation from the actually obtained targets y_i for all the training data and meanwhile as flat as possible. The case of linear



function f has been described in the form as

$$f(x) = \langle \omega, x \rangle + b \quad (2.1)$$

For this, it is required to minimize the Euclidean norm i.e. $\|\omega\|^2$. Formally this can be written as a convex optimization problem by requiring

$$\begin{aligned} & \text{minimize} \quad \frac{1}{2} \|\omega\|^2 + C \sum_{i=1}^l (\xi_i + \xi_i^*) \\ & \text{subject to} \quad \begin{cases} y_i - \langle \omega, x_i \rangle - b \leq \varepsilon + \xi_i \\ \langle \omega, x_i \rangle + b - y_i \leq \xi_i^* \\ \xi_i, \xi_i^* \geq 0 \end{cases} \end{aligned} \quad (2.2)$$

where ξ_i, ξ_i^* are slack variables to cope with otherwise infeasible constraints of the optimization problem. The constant $C > 0$ determines the tradeoff between the flatness of f and the amount up to which deviations larger than ε are tolerated.

2.5.2 K-Nearest Neighbor Regression

Nonparametric regression is a collection of techniques for fitting curves with little prior knowledge of their shape. The simplest algorithm to implement it is the k-nearest neighbor (k-NN) [30].

The k-NN regression is commonly based on the distance between a sample and the training dataset. Assume x is input with d features (x_1, x_2, \dots, x_d) and training set D with n samples. Normally we choose Euclidean distance as distance function, is defined as

$$\text{dist}(x, D_j) = \sqrt{\sum_{i=1}^k (x_i - D_{ji})^2} \quad (2.3)$$

The predicted value of x is decided by the great number of the value among k nearest sample in D .

Since K-NN is an instance-based learning method, given the stored training sets, the new samples' value is got only to find the most similar records in a training set. Therefore, when the training set grows, the memory space and computation time should be

considered.

2.6 Deep Learning Model



We will briefly introduce some important deep learning architecture used in this thesis. More detail about each model can be found in the reference.

2.6.1 Convolutional Neural Network

Convolutional neural networks (CNN) [31], which has successfully been applied to analyzing the image, is a class of deep feed-forward artificial neural networks that can explore the relationship between adjacent positions. CNN is proposed to reduce the number of parameters and extract feature pattern better than fully-connected layer. They are also known as shift invariant or spatially invariant artificial neural networks, based on their shared weight structure and translational invariance characteristics.

The convolutional layer is the core component of the CNN, which completes most of the computational heavy work. When we dealing with high-dimensional inputs such as images, it is impractical to connect neurons to all neurons in the previous layers. Instead, we will connect each neuron to only a local region of the input volume. The connections are local in space (along in width and high) but always fill along the entire depth of the input volume. Convolutional layer uses a weighting matrix called a filter (or feature) to handle the image, which detects specific properties such as diagonal edges, vertical edges and so on. Moreover, as the image progresses through each layer, the filters are able to recognize more complex attributes. Compared with other image classification algorithms, CNN uses relatively few preprocessing. This means that the filters in the traditional algorithms of network learning are designed by hand. This feature, independent of prior knowledge and human design, is a major advantage. Fig. 2.6 (A) shows an example in convolutional layer. Each neuron in the convolutional layer is connected only to a local

region in the input volume spatially. (B) shows the whole computation in one neuron, same as neuron networks.

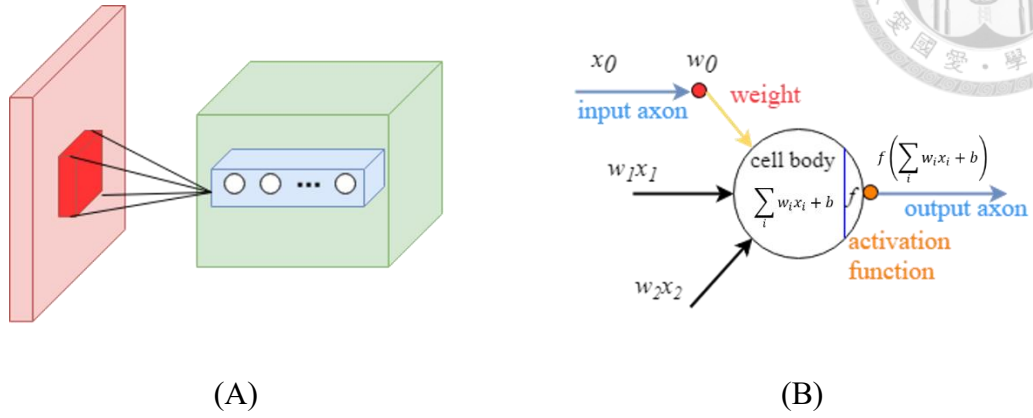


Fig. 2.6 (A) An example input volume in red and example volume of neurons in the first convolutional layer. (B) computation process in one neural.

2.6.2 Recurrent Neural Network

Recurrent neural network (RNN) not only takes the current input samples but also takes their previously perceived content as their input, normal neural network doesn't have this characteristic. RNN can be considered multiple copies of the same network, and each copy sends a message to a successor. As shown in Fig. 2.7, this kind of chain-like property reveals that RNN are closely related to sequences and lists. One of the attractions of RNN is that they may be able to connect the previous information to the current task, such as using previous video frames might inform the understanding of the present frame. It is often said that RNN has memory, in other words, RNN shares weights over time. The basic structure is described by the following set of equations [32]:

$$\mathbf{h}_t = h(\mathbf{W}_{xh}\mathbf{x}_t + \mathbf{W}_h\mathbf{h}_{t-1}) \quad (2.4)$$

$$\mathbf{y}_t = h(\mathbf{W}_{hy}\mathbf{h}_t) \quad (2.5)$$

where $\mathbf{h}_t \in \mathbb{R}^m$ is the state vector, $\mathbf{x}_t \in \mathbb{R}^p$ is the input, and $\mathbf{y}_t \in \mathbb{R}^m$ is the output. The function $h(\cdot)$ apply to vector pointwise and commonly set to $\tanh(\cdot)$. For the

coefficient matrices, we have $W_{xh} \in \mathbb{R}^{m \times p}$, $W_h \in \mathbb{R}^{m \times m}$, $W_{hy} \in \mathbb{R}^{m \times m}$.

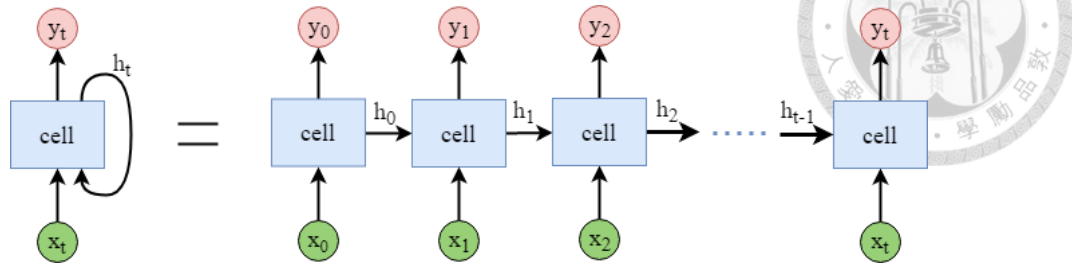


Fig. 2.7 Basic RNN architecture and unfold like.

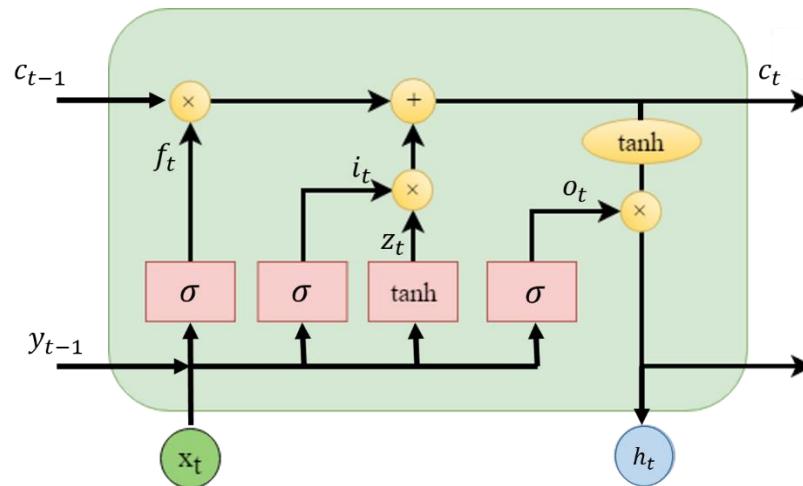


Fig. 2.8 LSTM cell contains four interacting layers.

In theory, RNN is absolutely capable of handling case of “long-term dependencies”, but in practice, RNN doesn’t seem to be able to learn them. In fact, the basic RNN usually facing gradient vanishing and gradient exploding problem which make RNN get bad results. This is partly because the information flowing passes many stages of multiplication. To address this problem, a variation of RNN with so-called Long Short-Term Memory units (LSTM) was proposed [33] and is largely used in sequences generation [34]. LSTM contain information outside the normal flow of the recurrent network in a gated cell. Cells decide what to store and when to allow reads, writes, and erasures by gates that open and close. LSTM single cell shown in Fig. 2.8, architecture is defined by the following set of equations

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i) \quad (2.6)$$

$$f_t = \sigma(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f) \quad (2.7)$$

$$c_t = f_t c_{t-1} + i_t \tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \quad (2.8)$$

$$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}c_t + b_o) \quad (2.9)$$

$$h_t = o_t \tanh(c_t) \quad (2.10)$$



where σ is the logistic sigmoid function, and i , f , o and c are respectively the *input gate*, *forget gate*, *output gate*, *cell* and *input activation* vectors, all of which are the same size as the hidden vector h . The weight matrix subscripts have the obvious meaning, for example, W_{hi} is the hidden-input gate matrix, W_{xo} is the input-output gate matrix etc. The weight matrices from the cell to gate vectors (e.g. W_{ci}) are diagonal, so element m in each gate vector only receives input from element m of the cell vector. The bias terms (which are added to i , f , c and o) have been omitted for clarity.

During training stage, we use backpropagation through time algorithm [35] to update network parameters. One difficulty when training LSTM with the full gradient is that the derivatives sometimes become excessively large, leading to numerical problems. To prevent this, all the experiments should clip the derivative of the loss with respect to the network inputs to the LSTM layers (before the sigmoid and *tanh* functions are applied) to lie within a predefined range.

2.6.3 Convolutional LSTM

The medical data like EMG signals with a lot of channels record continuously. The signal of the same channel at a different time is related, the signal between different channels also has the mutual influence. The fully connected LSTM (FC-LSTM) framework shown in the previous section provides a general framework for sequence

learning problems. However, the FC-LSTM does not take spatial correlation into consideration. Convolutional LSTM (ConvLSTM) network [36] have been proposed to overcome this difficulty. In order to model well the spatiotemporal relationships, they extend the idea of FC-LSTM to ConvLSTM which has convolutional structures in both the input-to-state and state-to-state transitions.

The main disadvantage of FC-LSTM in processing spatiotemporal data is that it must expand the input into a one-dimensional vector before processing, result in the loss of all spatial information in the process. To overcome this problem, all inputs X_1, \dots, X_t , cell outputs C_1, \dots, C_t , hidden states H_1, \dots, H_t , and gates i_t, f_t, o_t of the ConvLSTM are 3D tensors whose last two dimensions are spatial dimensions (rows and columns). The ConvLSTM determines the future state of a certain cell in the grid by the inputs and past states of its local neighbors. This can easily be achieved by using a convolution operator in the state-to-state and input-to-state transitions as shown in Fig. 2.9. The key equations of ConvLSTM are shown below

$$\begin{aligned}
 i_t &= \sigma(W_{xi} * X_t + W_{hi} H_{t-1} + W_{ci} \circ C_{t-1} + b_i) \\
 f_t &= \sigma(W_{xf} * X_t + W_{hf} * H_{t-1} + W_{cf} \circ C_{t-1} + b_f) \\
 C_t &= f_t \circ C_{t-1} + i_t \circ \tanh(W_{xc} * X_t + W_{hc} * H_{t-1} + b_c) \\
 o_t &= \sigma(W_{xo} * X_t + W_{ho} * H_{t-1} + W_{co} \circ C_t + b_o) \\
 H_t &= o_t \circ \tanh(C_t)
 \end{aligned} \tag{2.11}$$

where ‘*’ denotes the convolution operator and ‘ \circ ’, as before, denotes the Hadamard product. If we view the states as the hidden representations of moving objects, a ConvLSTM with a larger transitional kernel should be able to capture faster motions while one with a smaller kernel can captures slower motions.

To ensure that the state has the same number of rows as the input and the same number of columns, padding is needed before applying the convolution operation. Here,

the padding of the hidden state on the boundary point can be considered as using the state of the outside world for calculation. Usually, before the first input comes, we initialize all the states of LSTM to zero which corresponds to “total ignorance” of the future. Similarly, if we perform zero-padding on the hidden states, we are actually setting the state of the outside world to zero and assume no prior knowledge about the outside. By padding states, we can handle boundary points in different ways, which is useful in many cases.

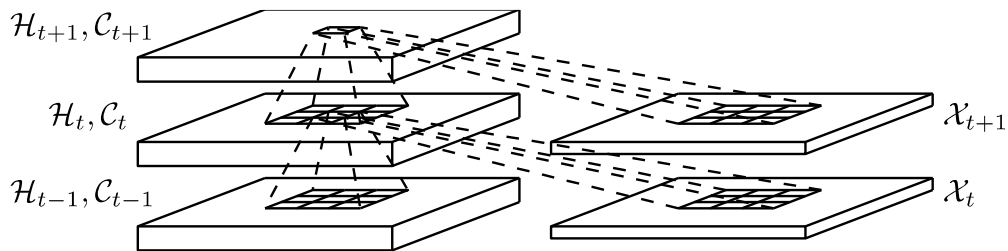


Fig. 2.9 Inner structure of ConvLSTM.

Chapter 3 Design Motion Prediction based Control System



The hardest part to control an exoskeleton robot is human-robot synergy. Usually, the movement of the machine has a delay relative to the movement of the human arm, which can make the user feel more resistance or be restrained. If we can predict the motion of the arm and let the machine move along with the human intent, it is an effective way to reduce the delay, so that users feel smooth in the motion. Traditionally, researchers use F/T sensors to get human arm movement intention or EMG signal to recognize movement intention direction. Those methods are hard to decode multi-joint movement, and it remains a problem with the delay which makes subjects feel uncomfortable and difficult to control the exoskeleton robot.

For the purpose of decoding multi-joint arm intention and reducing control delay time, we use human arm dynamics and EMG signals which can reflect movement intention as model input. We have proposed Multi-stream LSTM Dueling (MS-LSTM Dueling) model which can accurately predict joint angles of the human arm. After that, the predicted human joint angle becomes the input of the robot controller that can control the robot arms to move synchronously with the human arm.

In order to train our deep learning model, we need to collect the arm movement data from different subjects. Because of the huge differences between individuals, the researchers traditionally trained a model for each participant in a specific rehabilitation task. This is bound to require a large amount of data to be collected in each task and each participant, and it will take a lot of time in experimental setup and data collection. In fact, our data collection process does not require human arm to perform motions with the robot arm, unlike [37] which needs subjects to perform motion on the robot arm, in our case

subjects can perform motions in free space and do the motion whatever they want. The whole process here is compared with the traditional collection methods are more flexible and easier, and collection time is also greatly reduced. Moreover, subjects can also collect their data at home using the method we have designed. Through the proposed data collection method, we can collect the angles of the arm joints, the movement speed and the EMG signal of the subject.

3.1 Estimate Human Arm Dynamics and Muscle Activity

3.1.1 IMU subsystem and signal pre-processing

In this work, we use two Myo armbands, each with 3-axis magnetometer, 3-axis gyroscope, and 3-axis accelerometer, which altogether are called IMU. IMU signals sampling rate is 50 Hz and is located on the channel 4 of the Myo armband. One Myo armband is placed in the upper arm such that it is at 5 cm above the middle of lateral and medial epicondyle with channel 3 being located on biceps brachii, whereas the other armband is placed in the forearm such that it is at 6 cm below the middle of lateral and medial epicondyle with channel 3 being located on supinator. Both Myo armband's channel 4 will be on the top of the arm. The placement of both Myo armbands is as shown in Fig. 3.1. We obtain the stable rotation angle and angular velocity of each armband by using the complementary filter from IMU data. Finally, the angle of the human arm joint is obtained from the rotation angle of the two Myo armbands.

The 3-axis accelerometer allows us to obtain the rotation angles w.r.t X- and Y-axis, but the accelerometer is only suitable for measuring the angle in the stationary state, and the measurement accuracy under movement is poor. Similarly, the 3-axis magnetometer can obtain the angles between X-, Y- and Z-axis and the geomagnetic field repetitively, whereby they can also be converted to the rotation angles, however subject to

measurement of turbulence due to motion and vulnerability to the surrounding magnetic field interference. On the contrary, the gyroscope can measure the angular velocity of the object with sufficient accuracy, so that the angle of rotation through integration of the velocity can be obtained, despite the error of the angle may accumulate over time. This shows that magnetometer and accelerometers have high-frequency noise, gyroscope has low-frequency noise, and such a situation is very suitable for the use of complementary filters. The design of the filter is shown in the Fig. 3.2.

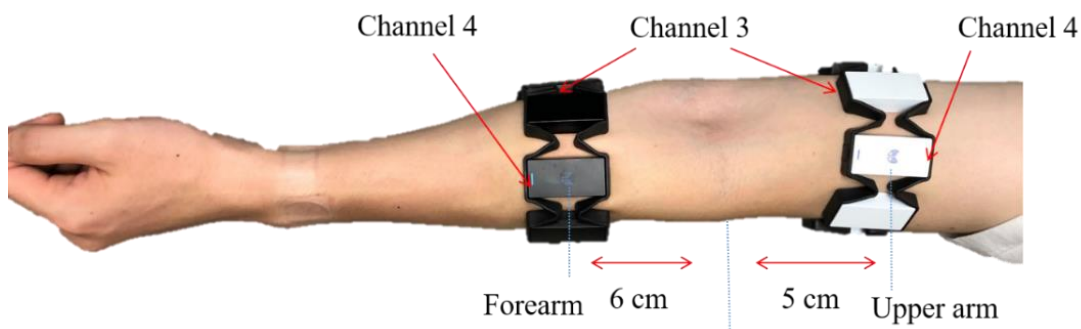


Fig. 3.1 Placement of two Myo armbands (left arm).

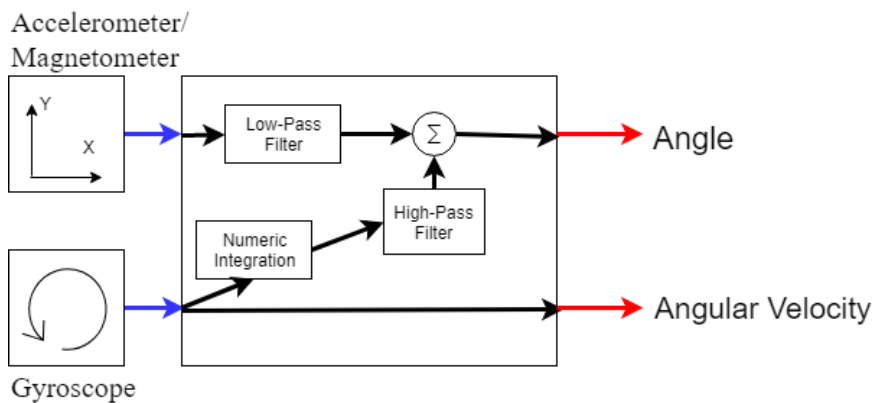


Fig. 3.2 The architecture of complementary filter which can complement each one of the shortcomings of the signal. Through the complementary filter, we can get the accuracy of the object's rotation angle and angular velocity.

What is shown below are equations for the digital complementary filter.

$$\theta_{yaw,t} = a \times (\theta_{yaw,t-1} + v_t^{gyro} \Delta t) + (1-a) \times \theta_{yaw,t}^{mag} \quad (3.1)$$

$$\theta_{X,t} = a \times (\theta_{X,t-1} + v_t^{gyro} \Delta t) + (1-a) \times \left(\frac{\theta_{X,t}^{mag} + \theta_{X,t}^{acc}}{2} \right) \quad (3.2)$$

$$a = \frac{\tau}{\tau + \Delta t} \quad (3.3)$$

where $\theta_{yaw,t}$ is the current filtered angle of yaw-axis, $\theta_{yaw,t-1}$ is the angle of previous time, $\theta_{X,t}$ is the current filtered angle of roll-axis and pitch-axis v_t^{gyro} is the angular velocity measurement of current time from the gyroscope, $\theta_{X,t}^{acc}$ is the current measurement angle from accelerometer, $\theta_{X,t}^{mag}$ is the current measurement angle from magnetometer, and a is determined from time constant τ . The time constant of a filter is the relative duration of signal it will act on. For a low-pass filter, signals much longer than the time constant pass through unaltered while signals shorter than the time constant is filtered out. The opposite is true for a high-pass filter. So, when the desired time constant and the sample rate are decided, the filter coefficient a will be determined.

Fig. 3.3 shows each Myo armband's coordinate system on which the rotation angle we obtained is based. After we get each axis's rotation angle, namely, $\theta_{0,roll}$, $\theta_{0,pitch}$, $\theta_{0,yaw}$, $\theta_{1,roll}$, $\theta_{1,pitch}$ and $\theta_{1,yaw}$, where θ_0 represent forearm coordinate system, θ_1 represent upper arm coordinate system, and we can use these as arm joint rotation angles.

The transformation equation is shown below

$$\begin{aligned} \text{shoulder flex./ext.} &= \theta_{1,pitch} \\ \text{horizontal abd./add.} &= \theta_{1,yaw} \\ \text{ext./int. rotation} &= \theta_{1,roll} \\ \text{elbow flex./ext.} &= \theta_{0,pitch} - \theta_{1,pitch} \end{aligned} \quad (3.4)$$

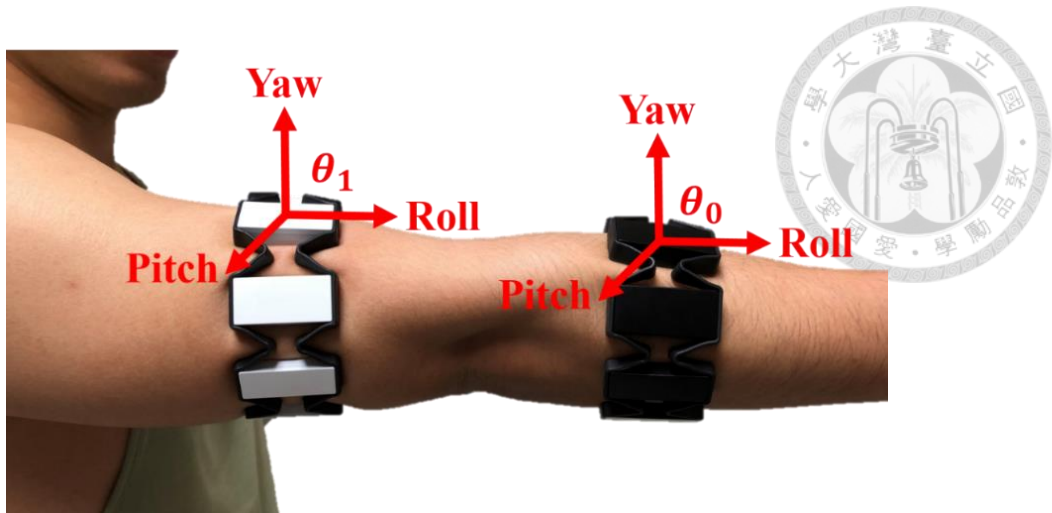


Fig. 3.3 Two Myo armbands' principal axes.

3.1.2 Human Arm Angle Calibration

Because the Myo armband is a ring-mounted sensor attached to the human epidermis, and human arm is not a homogeneous cylinder, the position of the sensor changes when the arm is moving, it causes the wrong calculation of the angles. Fig. 3.4 shows an example of doing elbow flexion causes sensor measurement inaccuracy. It's easy to find that when the elbow joint reaches 90° , the IMU sensor only reaches about 75° .



Fig. 3.4 Example of an inaccurate sensor measurement caused by muscle deformation.

In order to counteract the muscle-shape variation of each individual (there's a lot of difference in muscle deformation for everyone), a calibration measurement is needed to obtain the compensation value for muscle deformation before the experiment begins. We use four basic motions which are shoulder flex./ext. from 90° to 170° , horizontal

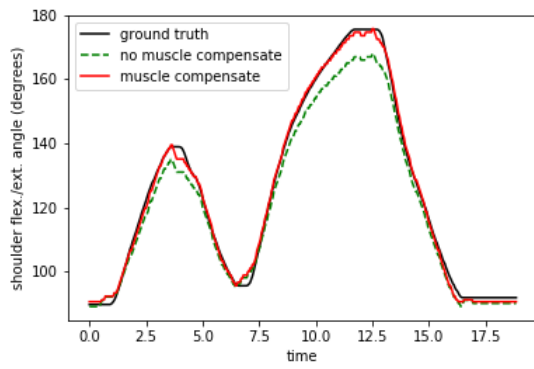
abd./add. from 0° to 80° , ext./int. rotation from -30° to 30° , and elbow flex./ext. from 0° to 85° to get the value of the compensation. This process needs to be done when the human arm sits on the exoskeleton because the robotic arm can provide an accurate angle of measurement and we will use it as ground truth. The robot arm will passively move while we record both the robot joint angle and estimated joint angle of the human arm.

$$k = \frac{\theta_{real} - \theta_h}{\theta_{real}} \quad (3.5)$$

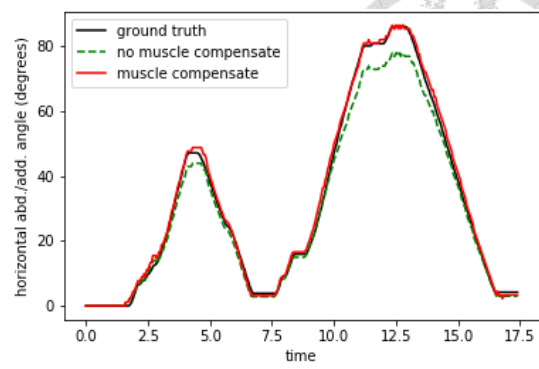
$$\hat{\theta}_h = \theta_h (1 + k) \quad (3.6)$$

where θ_{real} is ground truth joint angle, θ_h is human joint angle, k is muscle compensation value, and $\hat{\theta}_h$ is our final estimated human joint angle.

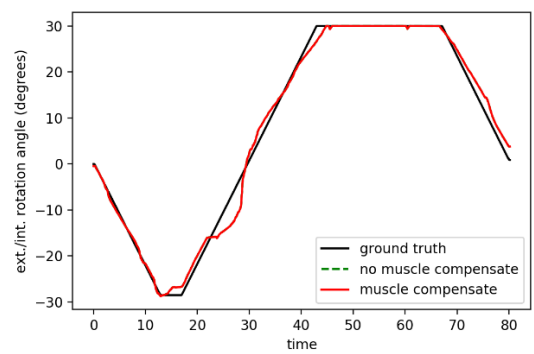
Fig. 3.5 shows the result of doing muscle calibration. The black line represents robot arm angle which means ground truth angle, the red line represents the measured angle after muscle compensation, and the green dash line represents the measured angle without muscle compensation. It's clear to see that if no muscle compensation, there is always a bias between our measured angles and the ground truth. Because of the design of the special mechanism, our robot arm produces a certain amount of deformation when it is horizontally moving, which will result in a small amount of error in our final measurements as shows in Fig. 3.5 (A) during 3s to 5s. Fig. 3.5 (C) shows that ext./int. rotation angle will be influenced by muscle deformation. After testing on 5 subjects, the measured mean absolute error with muscle compensation is shown in TABLE 3.1, which reveals that our method can accurately measure the joint angle of the human arm. For the sake of comparison, the measurement mean absolute error without muscle compensation is also shown in TABLE 3.2.



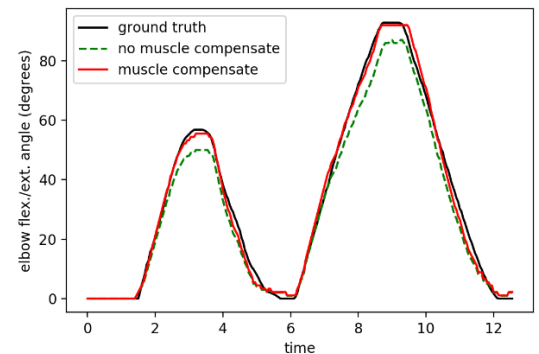
(A)



(B)



(C)



(D)

Fig. 3.5 Result of muscle calibration (A) shoulder flex./ext., (B) horizontal abd./add., (C) ext./int. rotation, (D) elbow flex./ext.

TABLE 3.1 Measurement error test on five subjects with muscle compensate.

Motion type	Horizontal abd./add.	Shoulder flex./ext.	Ext./int. rotation	Elbow flex./ext.
Mean absolute error (degrees)	1.23°	1.2°	1.19°	1.63°

TABLE 3.2 Measurement error test on five subjects without muscle compensate.

Motion type	Horizontal abd./add.	Shoulder flex./ext.	Ext./int. rotation	Elbow flex./ext.
Mean absolute error (degrees)	2.53°	3.72°	1.19°	3.78°

3.1.3 EMG Subsystem and Signal Pre-processing

In previous work, Two Myo armbands were used to get arm dynamics. Each device can also get 8 EMG channel signals to record the status of muscle activation of upper arm and forearm at the same time. The EMG sampling rate of Myo armband is 200 Hz.

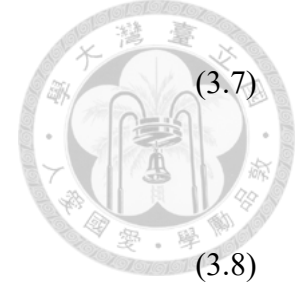
The signals we measure are mixed with various noises [38]. Moreover, because the Myo armband is a ring-shaped sensor, there will be a very serious crosstalk problem [39] on the obtained signals. As a result, the actual EMG signal from the muscles is reduced. Therefore, the original signal needs to be pre-processed by a series of signals processing to increase the signal-to-noise ratio.

In addition to the problem of crosstalk, there are two main noises in the EMG signal. The first is artifact noise, which occurs when motion comes up and where electrodes, skin, and muscle shift with respect to one another. The frequency of artifact noise is low-frequency noise and is in the range of 1-10Hz. The other kind of noise is electromagnetic noise which is also called power line noise. The human body can be regarded as a conductor and is constantly receiving electrical as well as electromagnetic radiation, especially from the power line. The magnitude of electromagnetic noise is usually greater than that of the EMG signal.

In order to eliminate the artifact noise, the raw EMG signal is filtered by a high-pass filter (fourth-order Butterworth filter) with cut-off frequency 10 Hz. For the electromagnetic noise, we apply a 60 Hz notch filter to remove it. After filter the signal, we have two methods to extract time domain features and frequency domain features. In the time domain, for the purpose of obtaining the smooth signal, we use full wave rectification and then processed through moving average with triangular sample window (length 100ms) to quantify muscle activities over a time period. The moving average method can be expressed as

$$MA_i[n] = \sum_{k=0}^{N-1} \frac{k+1}{N} \cdot EMG_i[n-k] \quad (3.7)$$

$$MA = \begin{bmatrix} MA_1 \\ MA_2 \\ \vdots \\ MA_m \end{bmatrix} \quad (3.8)$$



where N is length (samples) of window function, EMG_i is the i -th channel of filtered EMG signal, MA_i is the i -th channel of EMG signal after moving average, MA is the vector form of signals after moving average, and m is the number of channels ($m = 16$ in this work).

In the frequency domain, we use Short-time Fourier Transform (STFT) to transform signals to the frequency domain, and the window function we choose hamming window has 19 samples as shown in Fig. 3.6 (B). Fig. 3.6 (A) shows the window function of Myo armband which has 19 samples ($N = 19$) and the total weight of all samples are triangle. Due to our system is digital we use discrete-time STFT, the equation shown below

$$\mathbf{STFT}\{x[n]\}(m, \omega) = \sum_{n=-\infty}^{\infty} x[n]w[n-m]e^{-j\omega n} \quad (3.9)$$

where $x[n]$ is input signal and $w[n]$ is window function, m is discrete and ω is continuous, but the STFT is performed on a computer using the Fast Fourier Transform (FFT), so both variables are discrete and quantized. The number of frequency band is selected 10, so the dimension of EMG features are 10×16 , where 16 is number of channels.

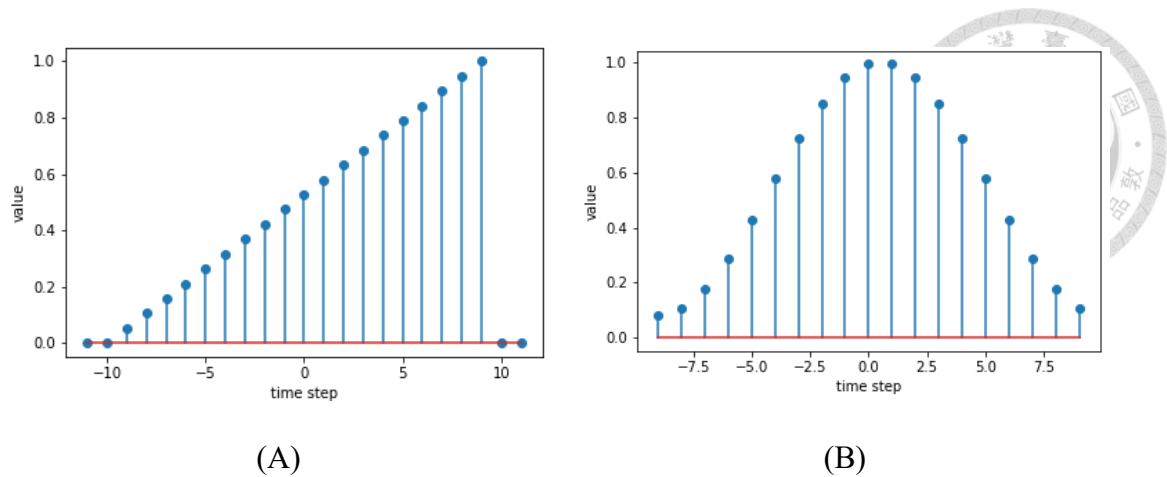


Fig. 3.6 (A) Example of a triangular window for moving average. (B) Example of the Hamming window.

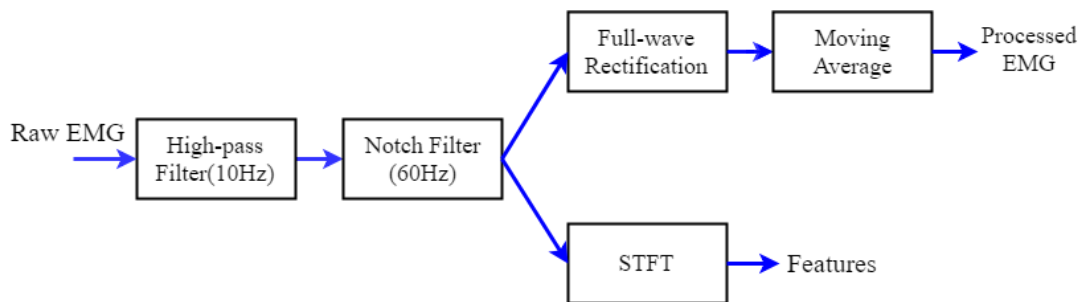
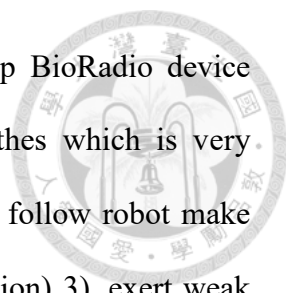


Fig. 3.7 Flow of signal processing.

The full signal processing is shown in Fig. 3.7. Note that, after applying STFT, which means time-frequency analysis, on a segmented window. The frequency domain features we get which are complex values, but we only use its magnitude as our model input features. Meanwhile, the processed EMG in one hand can also be input features. On the other hand, it can also serve as an important information for future assessment of patients' rehabilitation status.

3.1.4 Data Acquisition

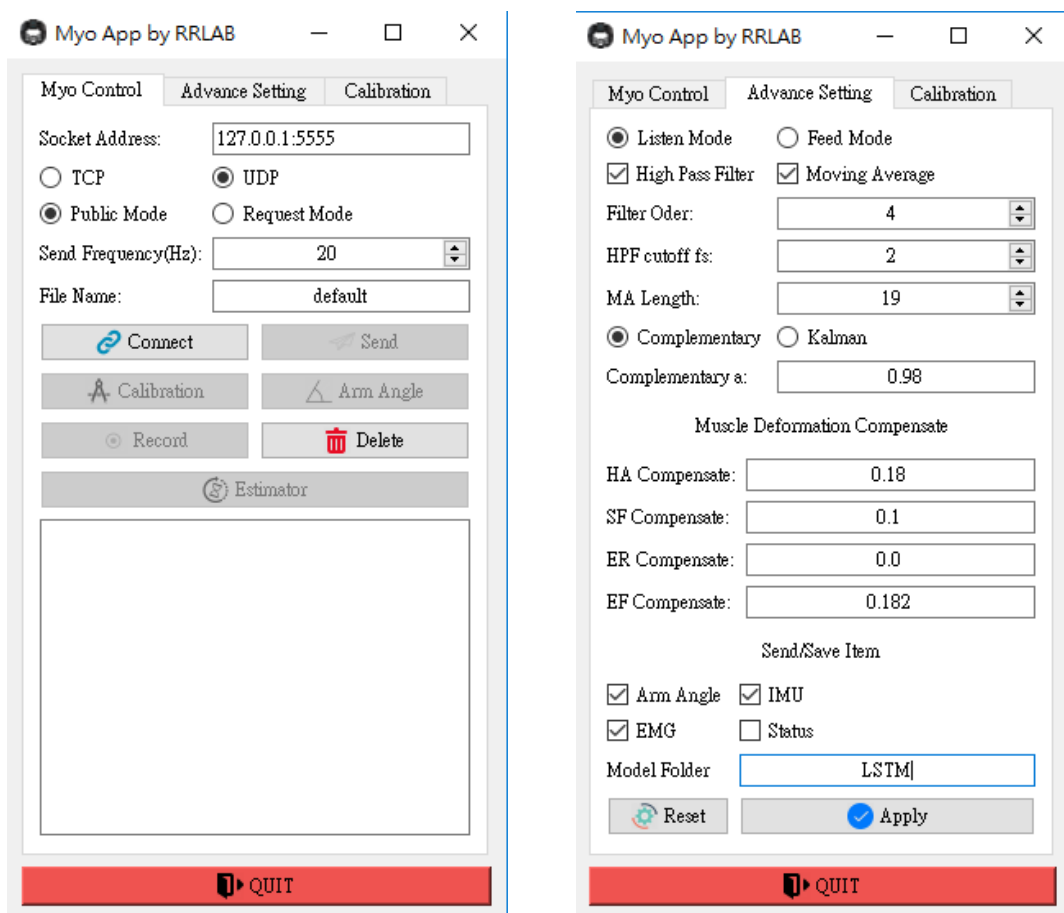
Before training our designed model, we need to collect training data. Training data are collected on both the health side of the subject which can make basic motions. Unlike our previous work [37], not only use Myo armband but also use BioRadio which is also



an EMG data collections devices through bipolar electrodes. Setup BioRadio device needs a lot of time and subjects also need to take off their clothes which is very inconvenient. Moreover, subjects have to perform 1). relax arm 2). follow robot make interactive force as small as possible (mimicking isometric contraction) 3). exert weak force on robot 4). exert strong force on exoskeleton robot actively. More irritating things is that each action needs to be repeated three times and requires at least 10 minutes, and for each different rehabilitation therapy need to repeat the same collection process. For example, if we have four rehabilitation therapies the total time we need at least 160 minutes. This collection method needs to be performed on every subject before experiments.

In our work, we have simplified the whole process, and we just need to get the subjects to move their arms in the free space after wear Myo armbands and no need to be on the robot arm. We ask subjects to move their arm in air arbitrary in one-minute and then after one-minute rest perform next session. For each subject, we need to collect eight sessions and the total time are less than 30 minutes. During movement, we record subjects' arm dynamics which include four arm joint angles, all sensors' angular velocities and all sensors' accelerometer reading (total 16-dimensional). Also, processed EMG signals (16-dimensional) and EMG features (160-dimensional) are recorded simultaneously. Since the period of software control loop in our exoskeleton robot, NTUH-II is 50ms (20Hz), the arm dynamics (IMU signals) and pre-processed EMG signals sampling rate are also downsampled at 20Hz. Additionally, we design a Graphical User Interface (GUI) for convenience. With this application, for those who are not professionals such as physical therapists or doctors can also easily use to collection patients' data or use for joints estimate. Fig. 3.8 shows our designed interface. (A) shows our main operating interface and can also be used to connect Myo armbands to transmit measurement data via UDP or

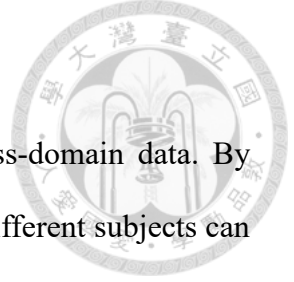
TCP to any other application, also user can set the sampling rate for a special case. (B) shows our advance setting page. The default value is the best value we choose for easy use; the user can modify to other values they want. Our data collection method only needs to collect data before training model or when need to fine-tune a special subject. More often, when new participants come in, they can use our existing models directly. Compared to the previous approach, the method we propose has little time for new participants to spend on data acquisition.



(A)

(B)

Fig. 3.8 GUI for our application.



3.2 Motion Prediction Regression Model

In this section, we use deep learning model to fuse two cross-domain data. By discovering common patterns in a large number of data, the use of different subjects can also show good generalization. We also design the corresponding fine-tune method for our deep learning model to be better adapted to new subjects.

3.2.1 One Stream LSTM Model

Since we have cross-domain features, it is important to make the most of this features rather than just using a single domain usually can get better results. There are many ways to fuse data, the simplest of which is to concatenate all the features before input to the deep learning model. This approach can often get good results; the neural network can learn to assign different weights to different domain features. Thus, based on the model structure used in the [40], we change the input part to the fusion of the arm dynamic data and the EMG signals. In order to facilitate the comparison of the latter, we name it LSTM model (one stream LSTM). Compared with the traditional regression model, the deep learning model can do end-to-end training, which means we can directly get all arm joint angle in one model, not four models. This approach can reduce both the number of models and the global optimization since the optimization of each small part of the model does not guarantee global optimization.

Fig. 3.9 shows the whole model architecture. First, we concatenate the arm dynamics and EMG frequency domain features, where D_t represent arm dynamics which include human arm joint angles, angular velocities, and acceleration read from IMU sensor at the current time, E_t represent EMG features extract through STFT at the current time. Second, we input consecutive data to linear LSTM layers (basic LSTM). Thus, the model not only inputs current data but also inputs previous n time steps data (we set $n = 20$).

The dependency on consecutive data is a significant issue. Quite a few models use sliding context windows to process partial data of interest [41], which precludes the dependent data outside the context window. LSTM have the ability to selectively contain and pass information across time steps with hidden states. Third, the LSTM layer's output will input to fully connected layer and finally generate future time arm joint angles.

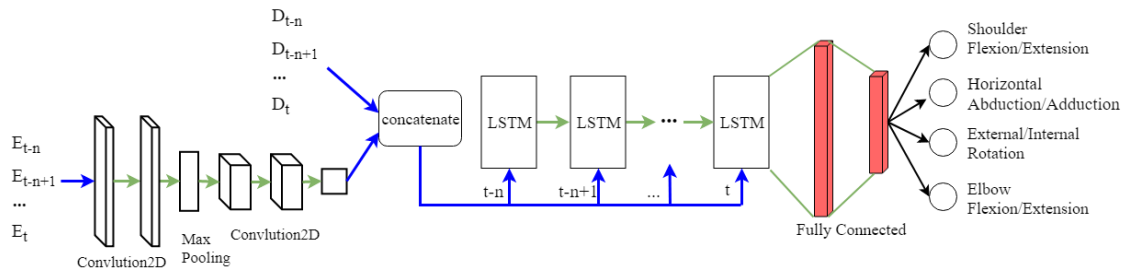


Fig. 3.9 One Stream LSTM model.

3.2.2 Multi-stream LSTM Dueling Model

In general, when initializing a neural network, each weight is with the same random distribution, which for example, can be a normal distribution with zero mean and standard deviation equal 0.05. The scope of parameter initialization is critical for the mode update, and their parameter updates tend to be in different directions and values for cross-domain features. Obviously, the data that come out of the arm dynamics, keep a strong linear relationship, whereas the EMG signals hold a non-linear relationship. As a result of these differences, the use of different architectures to handle data of different characteristics can be better. The idea of using multiple information sources in neural network language modeling and video action recognition modeling has been implemented before and it was shown that the neural network models benefitted from the extra information provided. In [42], the author proposed two stream LSTM language model which leads to a lower word error rate (WER). Similarly, Multi-Stream Bi-Directional Recurrent Neural Network was proposed in [43] to predict an action label. Those model have two common properties: 1)

multiple input sequences are fed to the network as parallel streams, and inputs from each stream are propagated to parallel hidden layers; 2) Multiple parallel hidden layers are then connected to the same output layer. Usually, the two stream outputs are concatenated and sent to one or two fully-connected layers and are then output as final results.

Fig. 3.10 shows our first design Multi-stream LSTM model. The first stream network we use FC-LSTM to decode arm dynamics because it is good for solving linear relational data. The design of architecture in the second stream network is inspired by [40], but we change FC-LSTM to ConvLSTM [36]. The EMG features will go through two convolutional layers, pooling layers, and another two convolutional layers and pooling layer, and then the final pooling layer before they reach ConvLSTM layer. Finally, we concatenate the outputs of FC-LSTM and ConvLSTM together through three fully-connected layers to generate our results. For the hyper-parameters of our model here we set, the first two convolutional layers have 16 filters, each with size 3×1 , and last two convolutional layers have 32 filters, each with size 3×1 , where max pooling is used in our pooling layers, and *relu* is used as the activation function for convolutional layers. The FC-LSTM layer, hidden size is 64 and ConvLSTM layer has 64 filters in total, each with size 3×1 . The two fully-connected layers have 64 units and the output layer has 4 units. The activation function of fully-connected layers are all linear function.

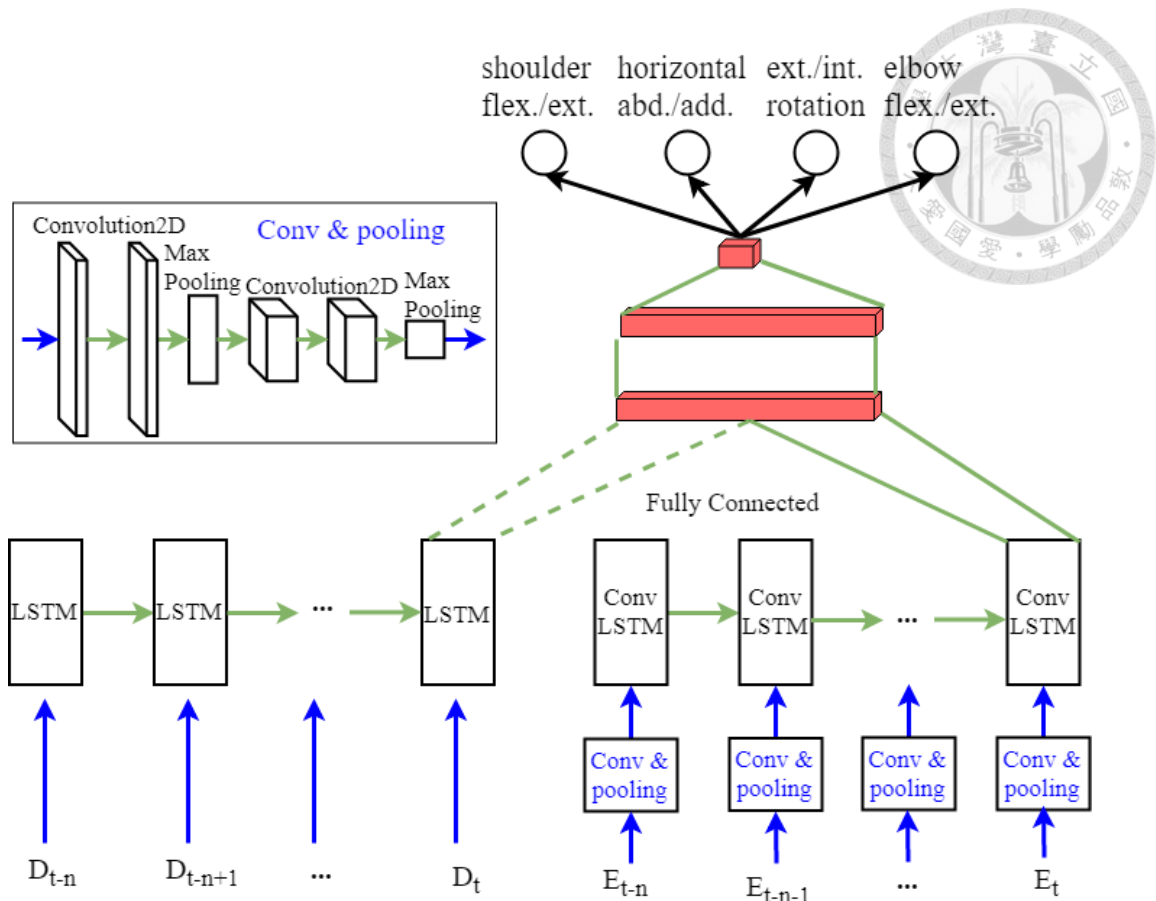


Fig. 3.10 Multi-stream LSTM model.

However, simply concatenating multiple streams may not be a good way. Inspired by Dueling Network [44] which has gained success in deep reinforcement learning, it lets the final layer separate value and advantage, and then add them elementwise to get final results. So, we let first stream output as the value and the second stream output as the stimulation, and add them elementwise in the final output to obtain four arm joint angles. Fig. 3.11 shows our new designed Multi-stream LSTM Dueling (MS-LSTM Dueling) model. The model's hyper-parameters are the same as those of previously designed model, but the difference is that the value layer has 4 units and the stimulation layer also have 4 units. The total number of parameters is less than that of the previous model but can get higher accuracy and more robust results.

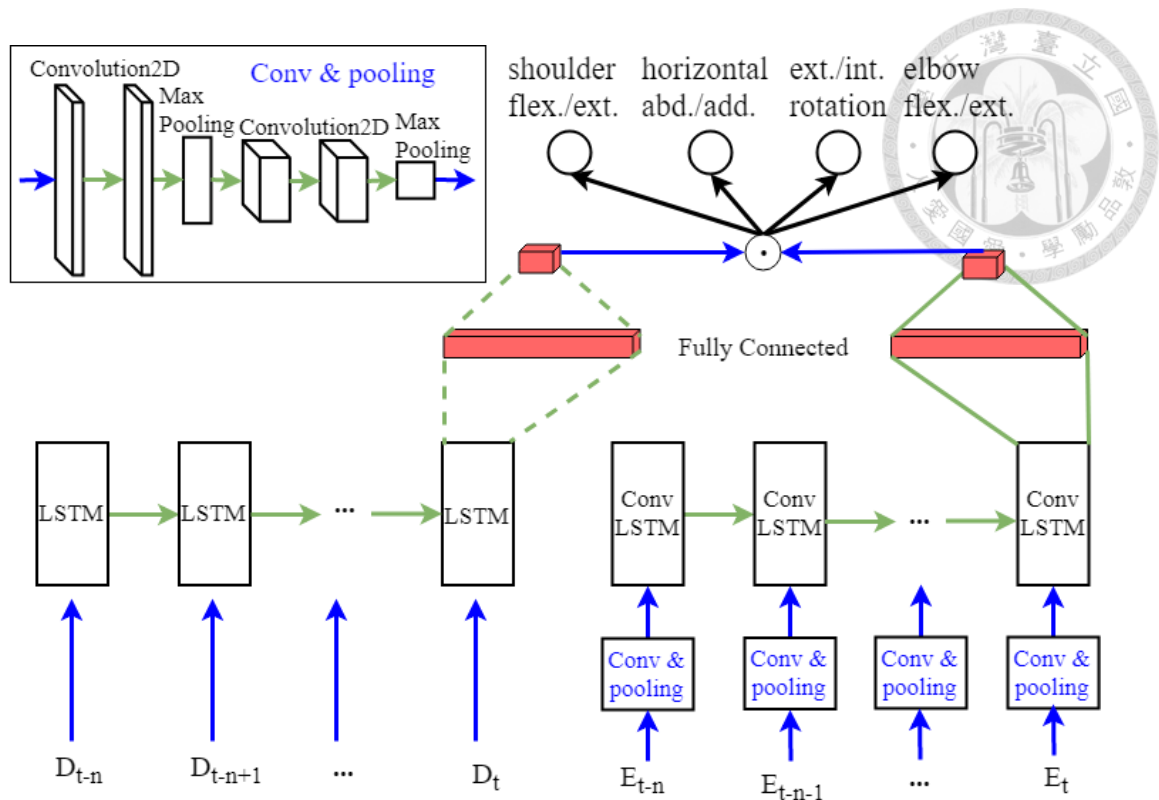
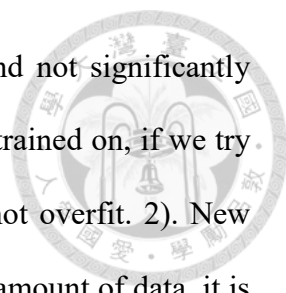


Fig. 3.11 Multi-stream LSTM dueling model.

3.3 Fine Tune of the Model

In practice, deep neural networks like CNN or LSTM has a huge number of parameters, often in the range of millions. Training a deep learning model on a small dataset greatly affects the neural network's ability to generalize, often result in overfitting. As a result, it is more common in practice to fine-tune the existing networks, which are trained on large datasets, by continuing to train the smaller datasets we own. Assuming that our dataset is not quite different from the context of the original dataset, the pre-trained model will have learned the characteristics associated with our own problem. For instance, the pre-trained network on a large and diverse dataset like the ImageNet captures universal features like curves and edges in its early layers, that are relevant and useful to most of the classification problems. However, not all situations can be fine-tuned, but there are some common rules of thumb own the 4 major scenarios. 1). New dataset is



large and similar to the original dataset. As we have more data and not significantly different from the context of dataset which the pre-trained model is trained on, if we try to fine-tune the entire network, we can be confident that we will not overfit. 2). New dataset is small and similar to original dataset. Because of the small amount of data, it is not a good idea to fine-tune the entire neural network due to the overfitting problem. Since the data is similar to the origin data, we expect that the higher-level features in neural network are also relevant to this dataset as well. Hence, the best idea might be fixed low-level layers of the neural network and tune the high-level layers' parameters. 3). New dataset is large and very different from the original dataset. Since the dataset is very large, we may expect that we are able to train the neural network from scratch. However, in practice, initialization from the weight of the pre-trained model is usually still beneficial. In this case, we would have enough data and confidence to fine-tune the entire network. 4). New dataset is small but very different from the original dataset. Since dataset is small, it is likely best to only train a linear classifier or regressor. Furthermore, since the dataset is very different, it may not be the best to train the classifier or regressor at the top of the neural network, which might contain more dataset-specific features. Instead, it might better train a traditional classifier or regressor, like SVM/SVR, to be activated somewhere in the early part of the network.

Obviously, there must be a difference in the amount of muscle activity in each person, so the second stream is the most influential in our MS-LSTM Dueling model. If we want to improve the performance of the model for the new subjects, we need to collect some data from the subjects before the experiment begins. We will use new collected data to fine-tune the pre-trained model, which is in line with the second scenario mentioned above. Fig. 3.12 shows our designed fine-tuning method, where gray dash line represents the fixed layer, which means we won't change its weights during fine-tuning stage. This

is because the first few layers capture common features related to our new problems, such as curves and edges. We want to keep these weights unchanged. Instead, we let the network focus on learning the dataset-specific features of subsequent layers. Besides, we use a smaller learning rate to train the network. Since our pre-trained weights are pretty good already as compared to the random initialization weights, we don't want to distort them too quickly and too much. We make the initial learning rate 5 times smaller than the initial learning rate for initial training.

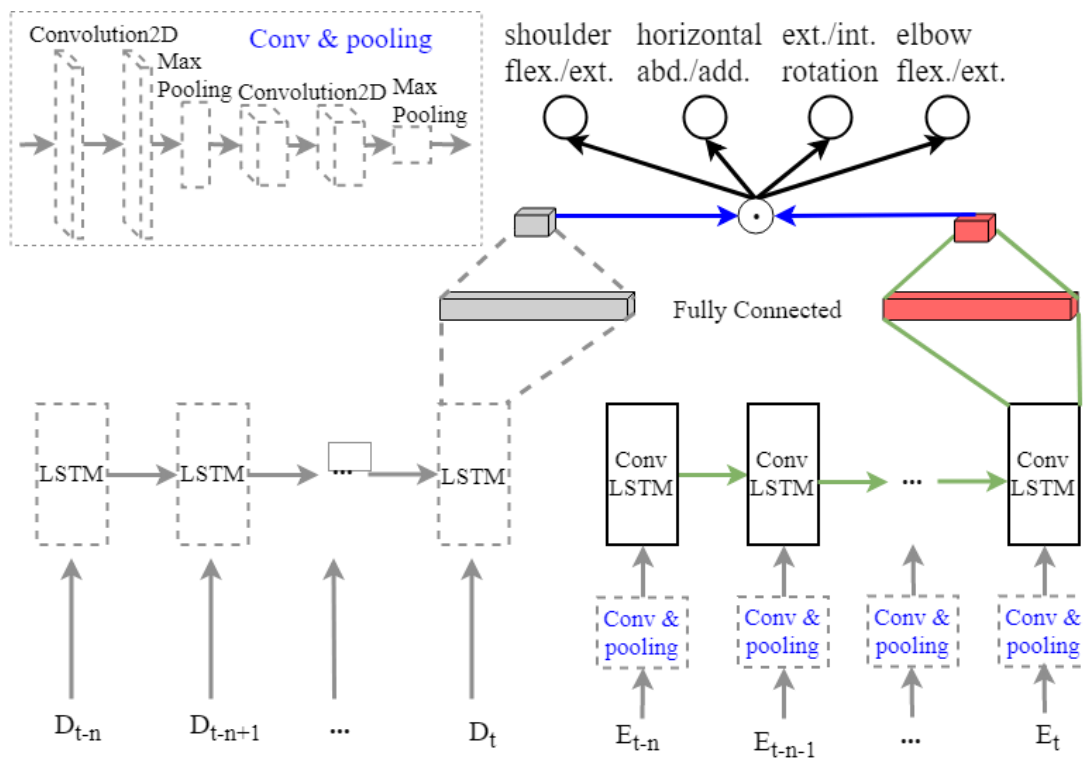


Fig. 3.12 Two Stream LSTM model fine-tune architecture.

3.4 Control of Robot System

We test the real-time performance of the proposed mode on the upper limb rehabilitation exoskeleton robot NTUH-II which we have introduced in Chapter 2.1. The PID controller we use is EPOS which is built in the motor already. Each joint has a standard PID controller. The output joint angles of our model will be set as controller's

reference trajectory and the velocity is the first order difference of the angle trajectories. We adjust the parameters of the controller so that it can track the reference trajectory as quickly as possible. Fig. 3.13 shows our whole control diagram where $\theta_{S,t}, \dot{\theta}_{S,t}$ are angles and angular velocities calculated by complementary filter regarded in the sensors' coordinate system, $\theta_{H,t}, \dot{\theta}_{H,t}$ are angles and angular velocities regarded in human arm's joint coordinate system, $\hat{\theta}_{H,t}, \hat{\dot{\theta}}_{H,t}$ are angles and angular velocities estimated after muscle compensation, $\hat{\theta}_{H,t+k}$ is the model outputs of human arm joint angles at k time steps ahead latent treated as reference trajectory to the controller, and $\hat{\dot{\theta}}_{H,t+k}$ is the first derivative of $\hat{\theta}_{H,t+k}$ represent human arm joint angular velocities. Note that τ_R is the controller's output torque to control motor motion and $\theta_R, \dot{\theta}_R$ are robot joint angles and angular velocities that will be feed back to the controller. The motion prediction regression model (MPRM) is our best designed MS-LSTM dueling model.

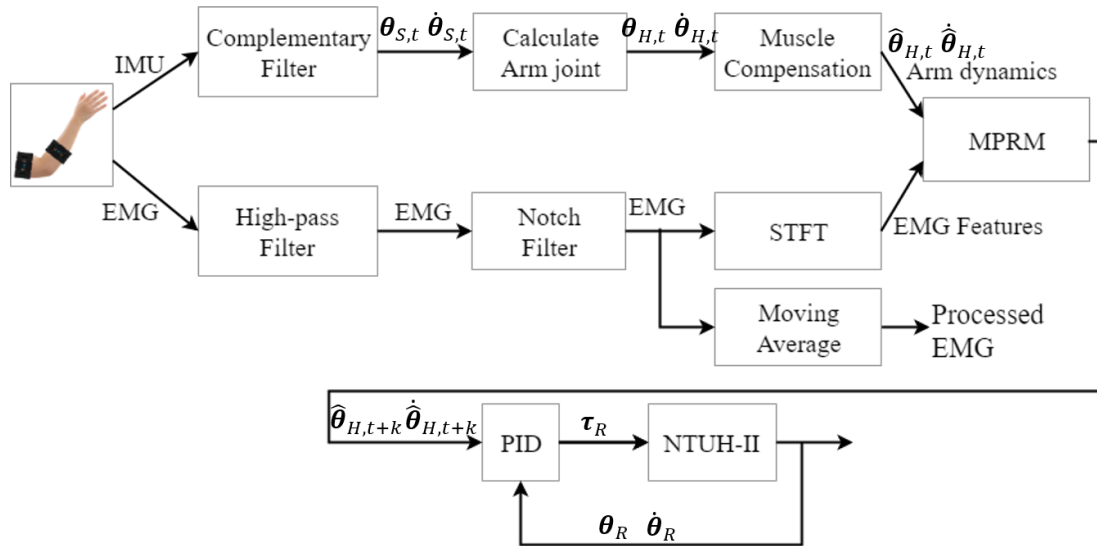


Fig. 3.13 Control diagram for guide mode and active mode.

3.4.1 Guide Mode

The guide model can be divided into bilateral mode and lead mode. In lead mode,

the therapist will wear two Myo armbands on the same side as the patient's impaired arm and perform standard treatment movement, whereby the robot arm will do the same to drive the patient's impaired arm to do rehabilitation as shown in Fig. 3.14 (A). This mode is suitable for the patient whose both limbs are affected and under low-level motor. The therapist who leads patients to do rehabilitation can let the patient quickly familiarize himself/herself with the process and strengthen his/her confidence.

Bilateral mode is suitable for patients with semi-paralysis. Two Myo armbands are worn in the patient's healthy arm which performs the specified rehabilitation movement, and the impaired arm which sits on the robot arm will be moved through mirroring motion by the robot arm as shown in Fig. 3.14 (B). With our motion prediction regression model, the robot arm will move synchronously with the therapist arm or the patient's healthy arm. Note that, we have to reverse the direction of horizontal abd./add. and ext./int. rotation angles in this mode.

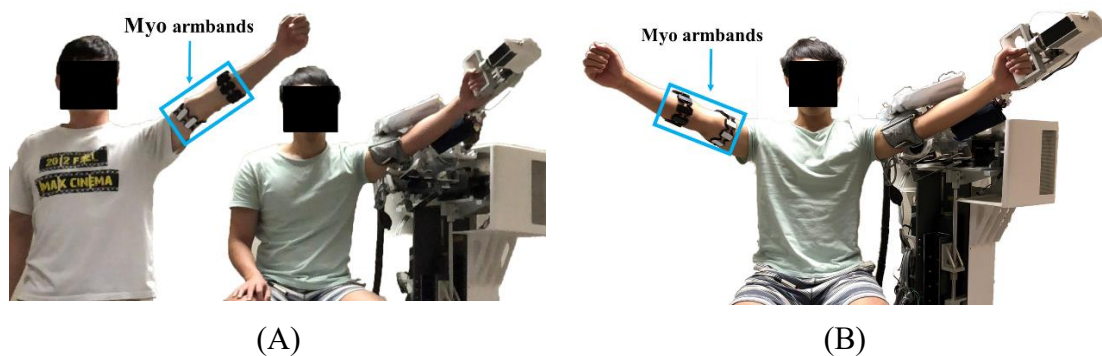


Fig. 3.14 (A) In lead mode, two Myo armbands are worn on the therapist's arm with the same side of patients impaired arm. (B) In bilateral mode, two Myo armbands are worn on the patient's healthy arm.

As a comparison, we will remove our designed model, directly using the estimate human arm's joint angles from IMU as reference trajectory to the controller. The control diagram is shown in Fig. 3.15.

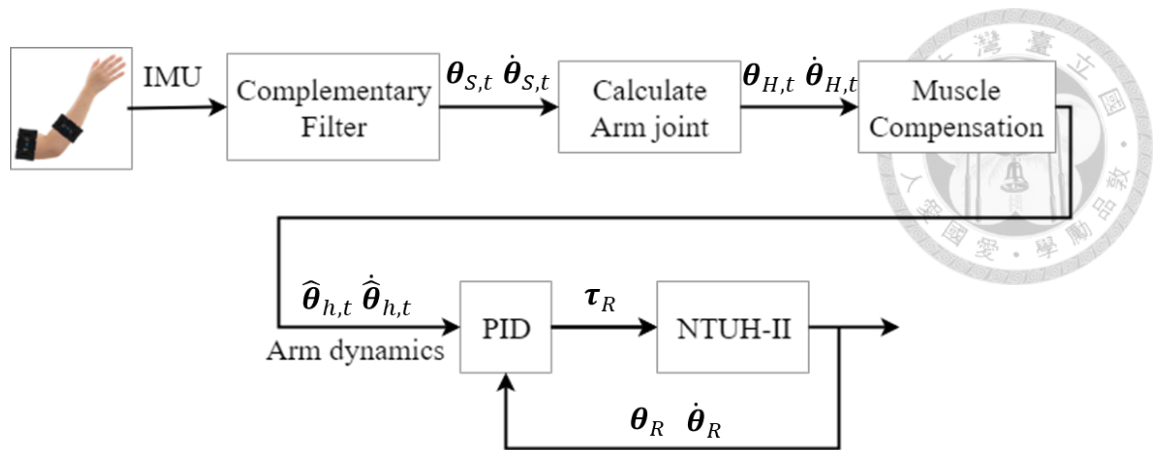


Fig. 3.15 Control diagram without MPRM for guide mode.

3.4.2 Active Mode

The patients after receiving full rehabilitation about their ROM could then undertake active mode therapy. In this therapy, two Myo armbands are worn in patients impaired arm and then the patient actively move their arm as shown in Fig. 3.16. The motion prediction regression model generates future joint angles and angular velocities to the controller making the exoskeleton robot follow the human arm's motion. The control diagram is shown in Fig. 3.13.

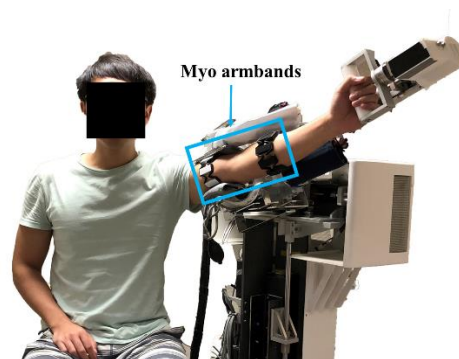
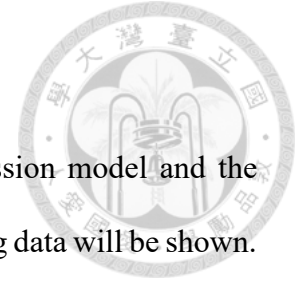


Fig. 3.16 In active mode, two Myo armbands are worn on the patient's impaired arm.

Chapter 4 Experimental and Results



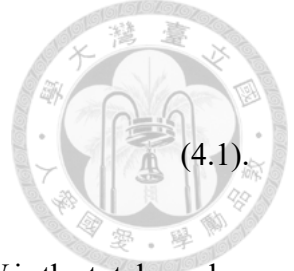
In this chapter, the training detail of motion prediction regression model and the result of comparison with different models as well as different training data will be shown. Next, the experiment protocol includes single joint task, multi-joint task, and evaluation index are introduced. In guide mode, the proposed method is compared with that only using IMU sensor based control system. In active mode, the proposed method is compared with EMG based motion intention recognition model [37]. After the experiment setup, the results are shown and discussed.

4.1 Model Training

We collect 8 health subject's left and right arm data. To test our designed model's ability, we select 5 sessions data out of 3 subjects as training data, the remaining 3 sessions data out of the 3 subjects and 8 sessions data out of other 5 subjects are all set as testing data. Note that, we use EMG features in training all deep learning model, but use processed EMG in traditional machine learning model because high dimensional features will cause curse of dimensionality [45]. To test the impact of different training data volume on the model, we train 4 different models, where one with 1 subject as training data (means his 5 session data for training, remaining as testing), one with 3 subjects as training data, one with 5 subjects as training data, and one with 7 subjects as training data. Before training models, we need to test our controller's ability to obtain the time for the future prediction of the proposed model. A predefined trajectory is given to the controller to calculate average delay time and set it as look-ahead time for model prediction. After test, our controller has 4-time steps delay. The controller sampling rate is 20Hz. All deep learning models are trained in 200 epochs with Adam [46] optimizer. The learning rates we set 0.001. Since our model is doing the regression task, we select mean squared error

as the loss function shown below

$$loss = \frac{1}{N} \sum_{i=1}^N \sum_{j=1}^4 (\theta_{ij} - \hat{\theta}_{ij})^2 \quad (4.1).$$



where θ_{ij} is the ground truth value, $\hat{\theta}_{ij}$ is the model output value, N is the total number of samples in one batch, j represents joint numbers.

4.2 Offline Evaluation Indexes and Result

In order to verify the performance and generalization ability of proposed model, we use mean absolute error (MAE) and standard deviations to evaluate the results. The equation is shown below:

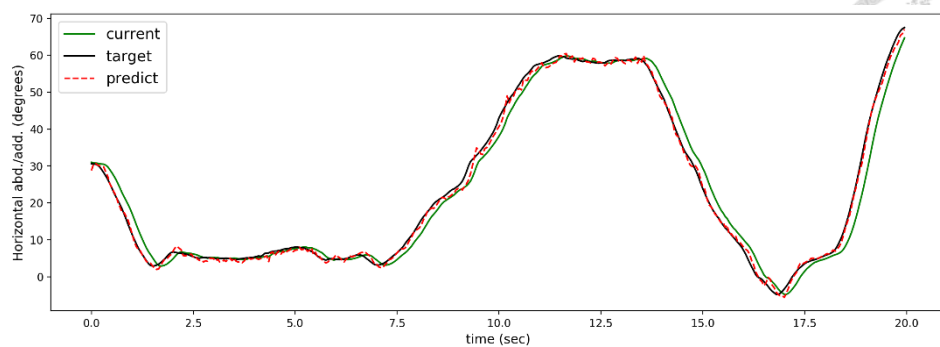
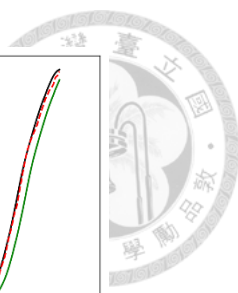
$$MAE = \frac{1}{KNJ} \sum_{k=1}^K \sum_{i=1}^N \sum_{j=1}^J |y_{ij} - \hat{y}_{ij}| \quad (4.2)$$

$$\sigma = \sqrt{\frac{1}{K} \sum_{k=1}^K (MAE_k - \mu)^2} \quad (4.3)$$

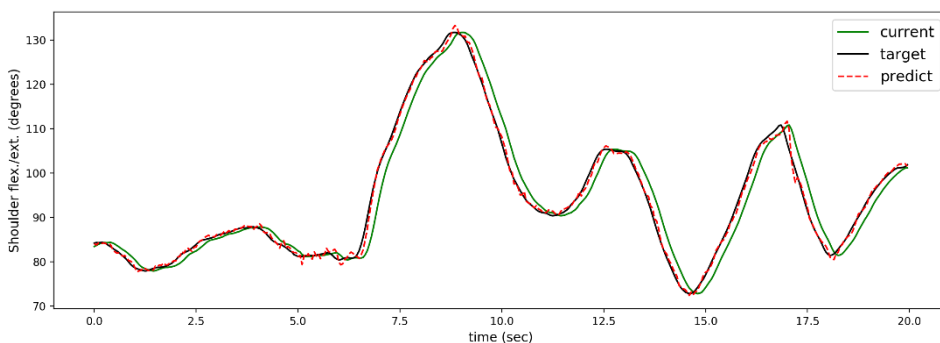
$$\mu = \frac{1}{K} \sum_{k=1}^K MAE_k$$

where K is the number of subjects, N is the number of sample from the corresponding subject and J is the number of joints. MAE_k represent k -th subject mean absolute error. In order to demonstrate the ability of our models to be used on new users, we define validation MAE is MAE calculate from remained 3 sessions' data of training subject, test MAE is MAE calculate from test subjects' all sessions data.

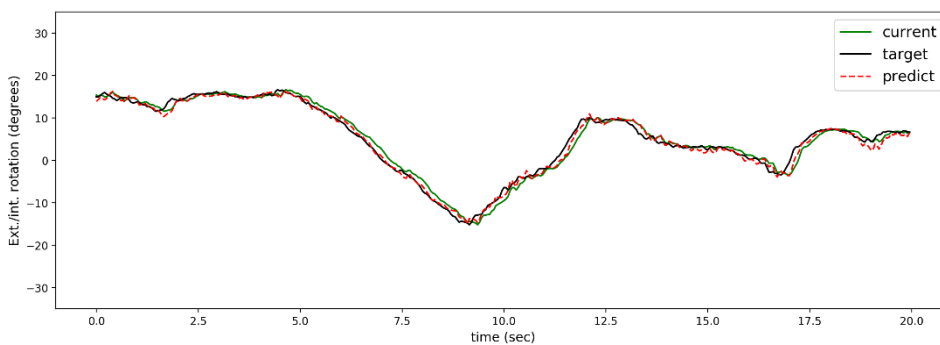
Fig. 4.1 ~ Fig. 4.2 shows the comparison between offline predict trajectory performed on two different models. Fig. 4.1 shows the predict human joint angle based on MS-LSTM Dueling model. To contrast the traditional machine learning model with the deep learning model, the results of SVR model is shown in Fig. 4.2. One subject's results are selected in the following figures for both models.



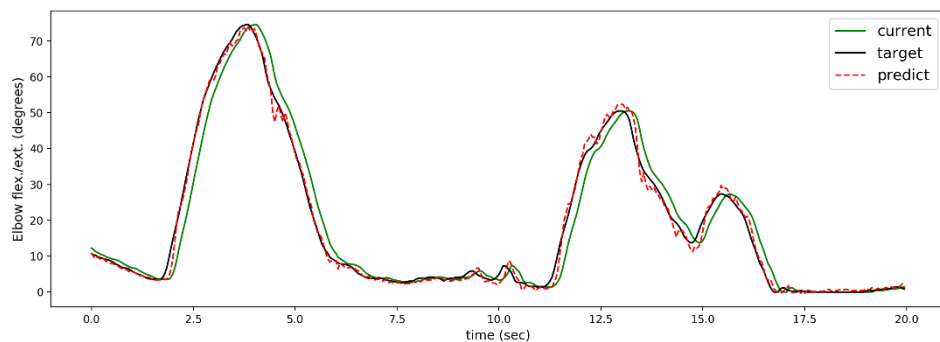
(A) Trajectory of horizontal abd./add.



(B) Trajectory of shoulder flex./ext.

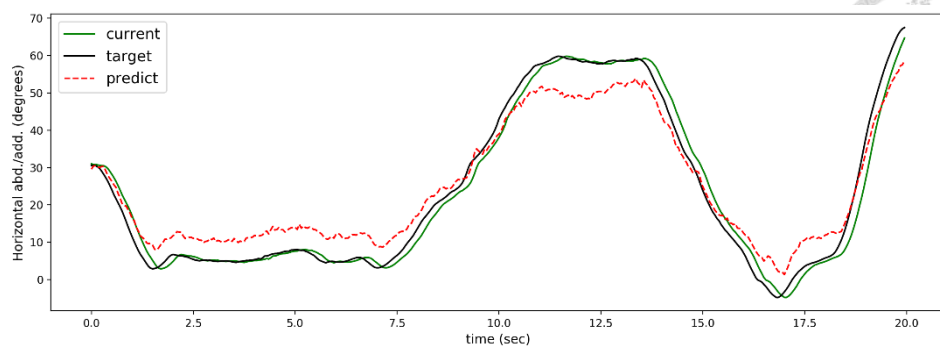
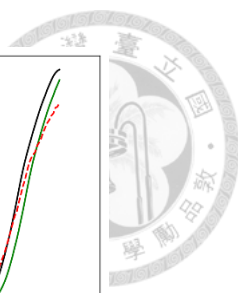


(C) Trajectory of ext./int. rotation

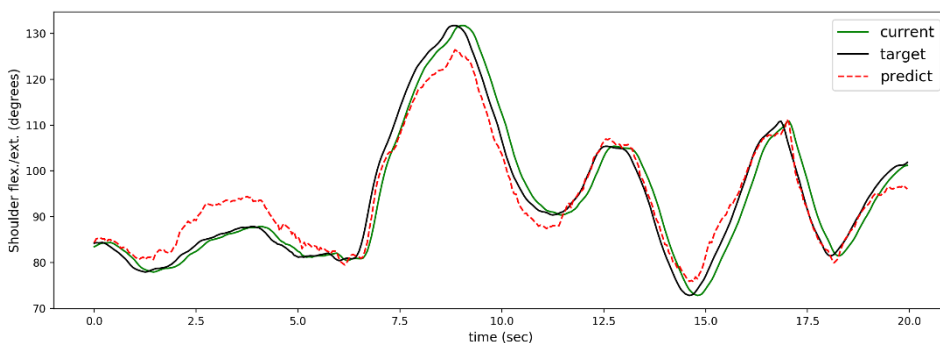


(D) Trajectory of elbow flex./ext.

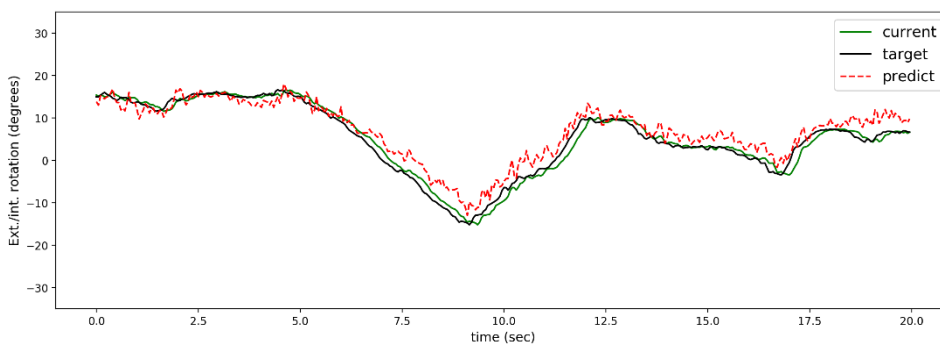
Fig. 4.1 Predict trajectory of four joint angle based on MS-LSTM Dueling model.



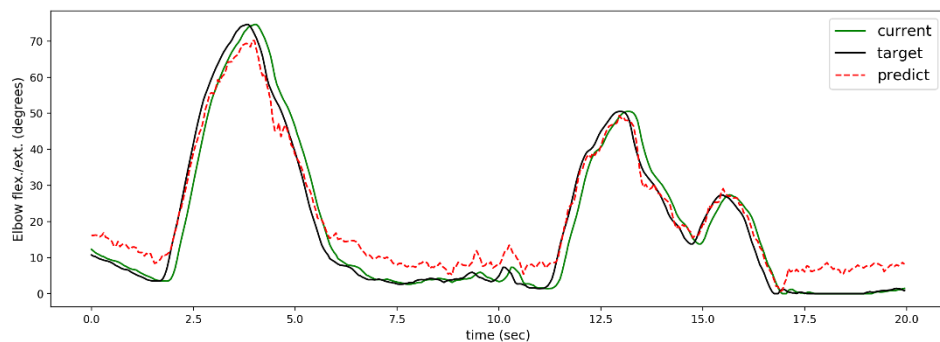
(A) Trajectory of horizontal abd./add.



(B) Trajectory of shoulder flex./ext.

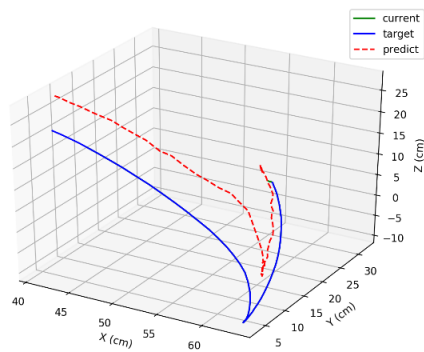


(C) Trajectory of ext./int. rotation

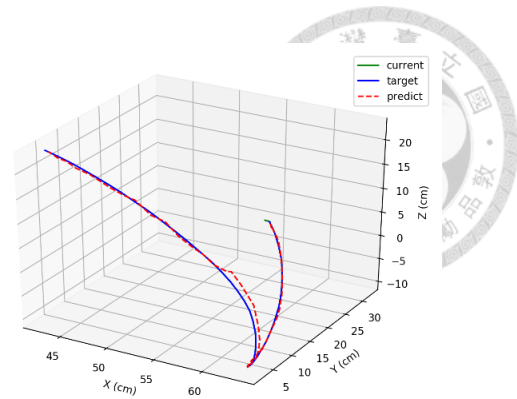


(D) Trajectory of elbow flex./ext.

Fig. 4.2 Predict trajectory of four joint angle based on SVR model.



(A) SVR model



(B) MS-LSTM Dueling model

Fig. 4.3 Compare hand position is 3D space reconstruct from arm joint angles of the different model.

The y-axis of Fig. 4.1 ~ Fig. 4.2 is joint angle (degrees) and the x-axis is time (second). The green line is the current arm joint angles, the black line is joint angles 4-time steps ahead which we can see horizontal translation between the green line and black line. The red dashed line represents joint angles of model output. The closer the red line to the black line, the higher the accuracy of the model. MS-LSTM Dueling model shows good performance to predict joint angles, whereas SVR model output has large bias and variation in each joint. In addition, the most important point is that the SVR model cannot learn to predict the start point. For instance, SVR model cannot predict when the arm starts to move in shoulder flex./ext. at 6s but MS-LSTM Dueling model can achieve that.

Fig. 4.3 shows hand position reconstructed from four joint angles for both model outputs. The green line is current position and blue time is the future position both in space, and these two lines will overlap together. Due to each joint angle output from SVR model has a bias, the predicted hand position shows large bias.

The result of different models' performance is shown in TABLE 4.1 and the MAE of single-joint for different models are shown in TABLE 4.2. The lower standard deviation, the smaller difference in the performance of different people, the better generalization ability of the model. The lower MEA, the more accuracy of the model

outputs. We can see from the table that the proposed model has the best performance and the performance improved in testing data after fine-tuning the model.



TABLE 4.1 Performance of different models.

Model type	σ	Total MAE	Validation MAE	Test MAE
SVR	0.52°	3.16°	2.69°	3.43°
KNR	0.60°	4.84°	4.18°	5.24°
LSTM	0.82°	4.17°	3.48°	4.58°
MS-LSTM	0.30°	1.44°	1.24°	1.55°
MS-LSTM Dueling	0.25°	0.97°	0.74°	1.11°
MS-LSTM Dueling Fine-tune	0.20°	0.90°	/	0.90°

TABLE 4.2 Total mean absolute error (degrees) of different models in single-joint.

Model type	Horizontal abd./add.	Shoulder flex./ext.	Ext./int. rotation	Elbow flex./ext.
SVR	3.69°	2.84°	2.35°	3.74°
KNR	5.91°	4.27°	5.01°	4.19°
LSTM	5.24°	3.04°	3.47°	4.95°
MS-LSTM	1.22°	0.86°	2.05°	1.61°
MS-LSTM Dueling	0.80°	0.73°	1.06°	1.33°
MS-LSTM Dueling Fine-tune	0.78°	0.63°	1.04°	1.17°

The performance of models which is trained on different training data volumes are shown in TABLE 4.3 and the MAE of single-joint are shown in TABLE 4.4. The results show that the more training data, the better performance of the model. But except for the first one, the difference between the other performance is very small.

TABLE 4.3 Performance of MS-LSTM Dueling model trained on different training data volumes.

Data volume	σ	Total MAE	Validation MAE	Test MAE
1train-7test	0.77°	2.33°	0.76°	2.55°
3train-5test	0.25°	0.97°	0.74°	1.11°
5train-3test	0.21°	0.96°	0.74°	1.10°
7train-1test	0.18°	0.84°	0.73°	0.99°

TABLE 4.4 Total mean absolute error (degrees) in single-joint.

Data volume	Horizontal abd./add.	Shoulder flex./ext.	Ext./int. rotation	Elbow flex./ext.
1train-7test	1.80°	1.59°	2.16°	3.78°
3train-5test	0.80°	0.73°	1.06°	1.33°
5train-3test	0.79°	0.69°	1.08°	1.27°
7train-1test	0.69°	0.60°	0.95°	1.10°

All subjects' result distribution of different models and different training data volumes are shown in Fig. 4.4. From (A) we can see that after fine-tune stage, MS-LSTM Dueling model can get minimum variance. From (B) it shows that similar distribution if the model is trained on data from multiple subjects.

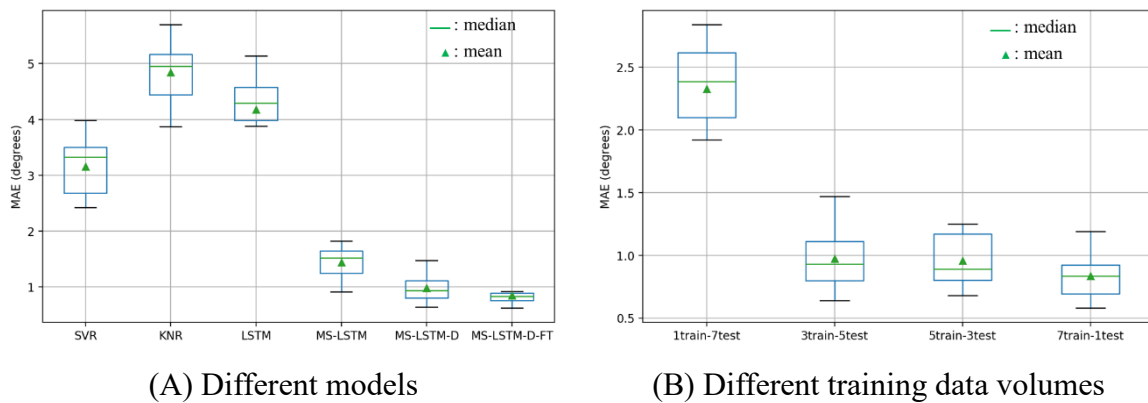


Fig. 4.4 Boxplot of (A) different models and (B) different training data volumes for all subjects' result distribution.

4.3 Real-time Experiment Protocol

In order to evaluate our system, we design two kinds of experiment, which are single-joint exercises and multi-joint exercises in each of two modes (guide mode and active mode) to verify the effectiveness of the proposed model. We select the model which trained with 5 sessions data out of 3 subjects. There are 3 healthy subjects invited for these real-time experiments. Their ages range from 22 to 25 and the gender of this group contains two males (one subject's data used as training data, another is new subject) and one female (new subject). Additionally, they don't have any upper limb impairment, to imitate the patient, the subject's left arm is set as the impaired arm and right arm is set as the healthy arm.

The experiments are designed to evaluate the single-joint and multi-joint performance of the proposed system. We choose horizontal abd./add., shoulder flex./ext., ext./int. rotation, elbow flex./ext., exercises for the single-joint evaluation, and select feeding (two joint) and greeting (four joint) exercise for multi-joint evaluation. The desired trajectories are shown in Fig. 4.5 and corresponding joint angles detail are listed in TABLE 4.5. The first four exercises in single-joint are common upper limb rehabilitation exercises for regaining motor functions in elbow and shoulder joints and the rest of two task-oriented exercises are used for improving motor control. Moreover, feeding exercise is composed of elbow flex./ext. and horizontal abd./add. in specific upper limb configuration, which can be achieved by controlling NTUH-II with the predefined pose (shoulder flexion at 65 degrees and shoulder external rotation at 20 degrees). Greeting exercise is composed of four joint with free motion. During the experiments, a monitor in front of the subject shows the current joint angles as the visual feedback to the subjects. Each task should be performed for 3 repetitions.

TABLE 4.5 Experiment protocol of single-joint and multi-joint tasks

Task	Protocol
Single joint	1. Horizontal abd./add. : S: 0° → P: 60° → E: 0° 2. Shoulder flex./ext. : S: 90° → P: 150° → E: 90° 3. Ext./int. rotation : S: 0° → P: -30° → P: 30° → E: 0° 4. Elbow flex./ext. : S: 0° → P: 60° → E: 0°
Multi-joint	5. Feeding (two joint) : Horizontal abd./add. : free Shoulder flex./ext. : 65° Ext./int. rotation : 20° Elbow flex./ext. : free 6. Greeting (four joint) : all joint free

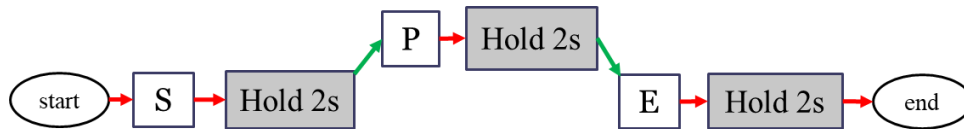


Fig. 4.5 Desired trajectory for subjects

The performance of evaluation will be separated into two parts according to rehabilitation mode. For guide mode, mean absolute error MAE and average delay time DT are used to evaluate the motion prediction regression model accuracy and effectiveness. when the MAE and DT time is small, which means the robot can meet the human arm current motion simultaneous. The MAE of tracking error is shown in (4.2) and average delay time can be calculated as

$$DT = \frac{1}{N} \sum_{i=1}^N \frac{\theta_{H,i} - \theta_{R,i}}{\omega_{R,i}}, \text{ when } \omega_R \neq 0 \quad (4.4)$$

where θ_H is human arm joint angle, θ_R is robot joint angle and ω_R is robot angular velocity. Delay time only be calculated during motion stage since there is no delay when the robot remains stationary. To evaluate motion smoothness, another index

“dimensionless-squared-jerk” (DLJ) which evaluates the square jerk magnitude with normalized unit in each joint is used. The expression is shown below:

$$DLJ = \frac{[(n_2 - n_1) \cdot \Delta t]^3}{\omega_{peak}^2} \sum_{n_1}^{n_2} \ddot{\omega}_R(t)^2 \cdot \Delta t \quad (4.5)$$

where n_1 and n_2 are start time and end time, ω_{peak} is peak value of robot angular velocity, Δt is sampling time.

In active mode, there are two indexes to evaluate the performance during static period and moving period. First, the average interactive force during moving period $avg |F_{mo}|$ is defined as

$$avg(|F_{mo}|) = \frac{\sum_{T_{mo}} |F_{mo}|}{T_{mo}} \quad (4.6)$$

where F is the interactive force on the direction of task recorded by the F/T sensor placed on the end-point and T_{mo} is the time during the moving period. This evaluation indicator shows whether the model makes robot arm to coordinate with users.

Another index is the standard deviation of joint angle during static period σ_{st} , which represents the performance when the subject tries to halt at some place. σ_{st} can be computed as

$$\sigma_{st} = \sqrt{\frac{\sum_b \sum_c \sum_{T_{st,c}} (\theta_{R,b} - \bar{\theta}_{st,c,b})^2}{b \sum_c T_{st,c}}} \quad (4.7)$$

where $T_{st,c}$ is the time during c -th static period, b is the number of joint in the task and $\bar{\theta}_{st,c}$ is the mean value of joint angle in the c -th static period.

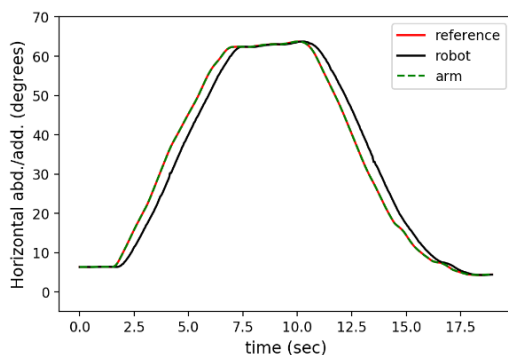


4.4 Real-time Experiment Result

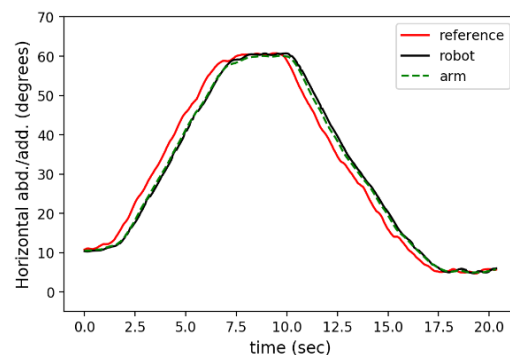
4.4.1 Performance for Bilateral Mode Exercise

Fig. 4.6-Fig. 4.8 shows the joint angle comparison between experiments performed by IMU based control in left column and MS-LSTM Dueling model based control in right column. The y-axis of the figure is joint angle (degrees) and the x-axis is time (second). Specific in Fig. 4.7(C), the trajectories of feeding (two joint) task are displayed in 2-dimensional joint space where the x-axis and y-axis are both joint angles (degrees). One subject's results are selected in the following figures for every task. TABLE 4.6-TABLE 4.8 shows average evaluation results of all subjects for each task.

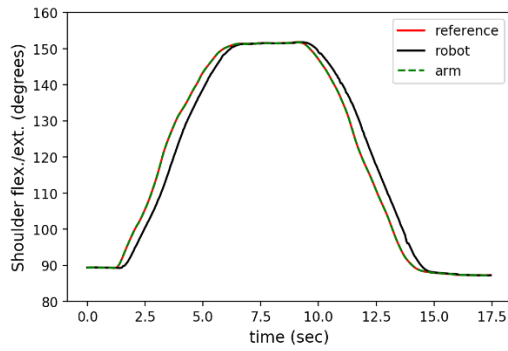
In IMU based control, the current human arm angle is used as a reference trajectory, allowing the robotic arm to track the trajectory. As shown in the figure of left column where the reference trajectory (red solid lines) overlap the human arm trajectory (green dashed line). In MS-LSTM Dueling model based control, the predict human arm angle is used as a reference trajectory. This allows the robot arm to start in advance to achieve a synchronized movement with the human arm as shown in the figure of right column where the human arm trajectory (green dashed line) is very close to the robot arm trajectory (black solid lines).



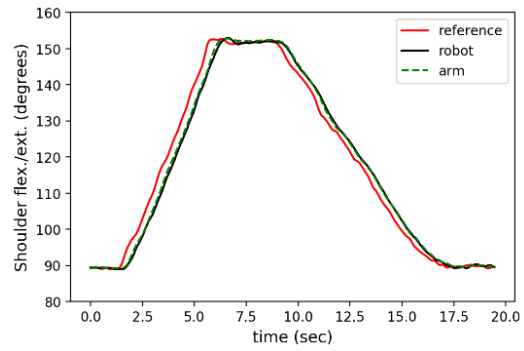
(A1) Horizontal abd./add.



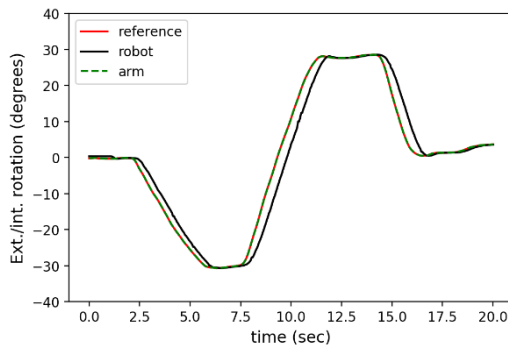
(A2) Horizontal abd./add.



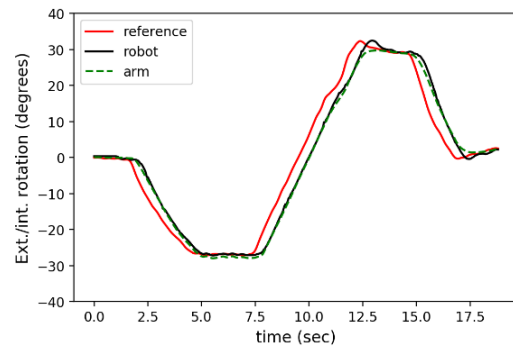
(B1) Shoulder flex./ext.



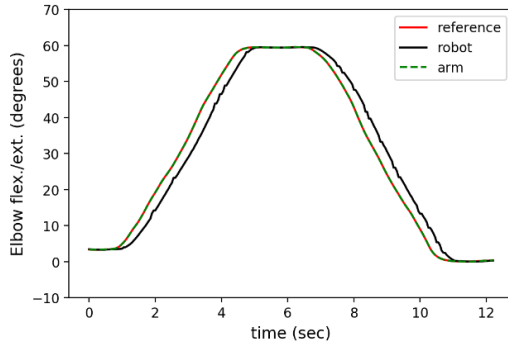
(B2) Shoulder flex./ext.



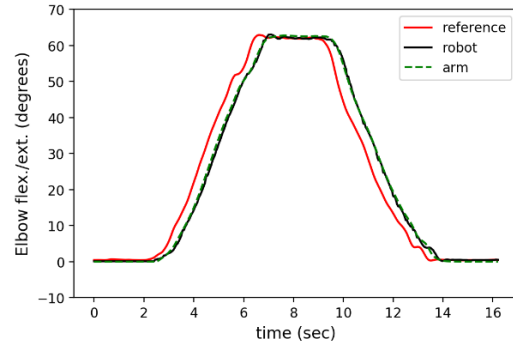
(C1) Ext./int. rotation



(C2) Ext./int. rotation



(D1) Elbow flex./ext.



(D2) Elbow flex./ext.

Fig. 4.6 Real arm angle (green dashed line), predict joint angle as reference trajectory (red solid line) joint angle and robot joint angle (black solid line) of IMU control (left column) and MS-LSTM Dueling model control (right column) for single-joint tasks in bilateral rehabilitation.

TABLE 4.6 shows average results of all subjects for single-joint tasks in bilateral mode where Normalized index represent value of Proposed index value divided by IMU index value. The smaller of normalized value, the better of our proposed method. Both MAE and DT are very small in our proposed method, reflecting the high synchronism

between the robot and the human arm. Since the output of the model has a slight oscillation, the smoothness of the movement of the robot arm is not as good as that of the IMU based control as shown in DLJ index, but the difference is small. It is difficult to feel that the smoothness of using MS-LSTM Dueling mode control is worse than using IMU signal directly control.

TABLE 4.6 Evaluation results of single-joint tasks in bilateral mode.

Single Joint	Index	IMU	Proposed	Normalized(1)
Horizontal abd./add	MAE (degrees)	3.15°	0.91°	0.29
	DT (sec)	0.44	0.075	0.17
	DLJ (10 ⁷)	5.61	6.98	1.24
Shoulder flex./ext.	MAE (degrees)	2.8°	1.33°	0.48
	DT (sec)	0.344	0.161	0.47
	DLJ (10 ⁷)	11.57	14.94	1.29
Ext./int. rotation	MAE (degrees)	2.63°	1.44°	0.55
	DT (sec)	3.182	0.414	0.13
	DLJ (10 ⁷)	27.99	25.50	0.91
Elbow flex./ext.	MAE (degrees)	3.2°	0.62°	0.19
	DT (sec)	0.325	0.016	0.05
	DLJ (10 ⁷)	3.26	4.16	1.27

Fig. 4.7 (A) and (B) illustrate each joint trajectory in feeding task. (C) shows the joint trajectories in 2D-space which can better reflect the coordination of the two-axis movement. We don't expect horizontal or vertical lines in 2D-space diagram, which means that the two joints cannot coordinate motion. Compare left column figures with right column figures, the results are basically the same as the single joint task. It can be said that the feeding (two joint) experiment is as good as the single joint tasks

performance. Both methods embody good two-joint coordination performance but IMU control shows a high delay time which makes the subject feel more difficult to control.

TABLE 4.7 shows average experiment results of all subjects for feeding task in bilateral mode. The results are as good as the single-joint tasks performance.

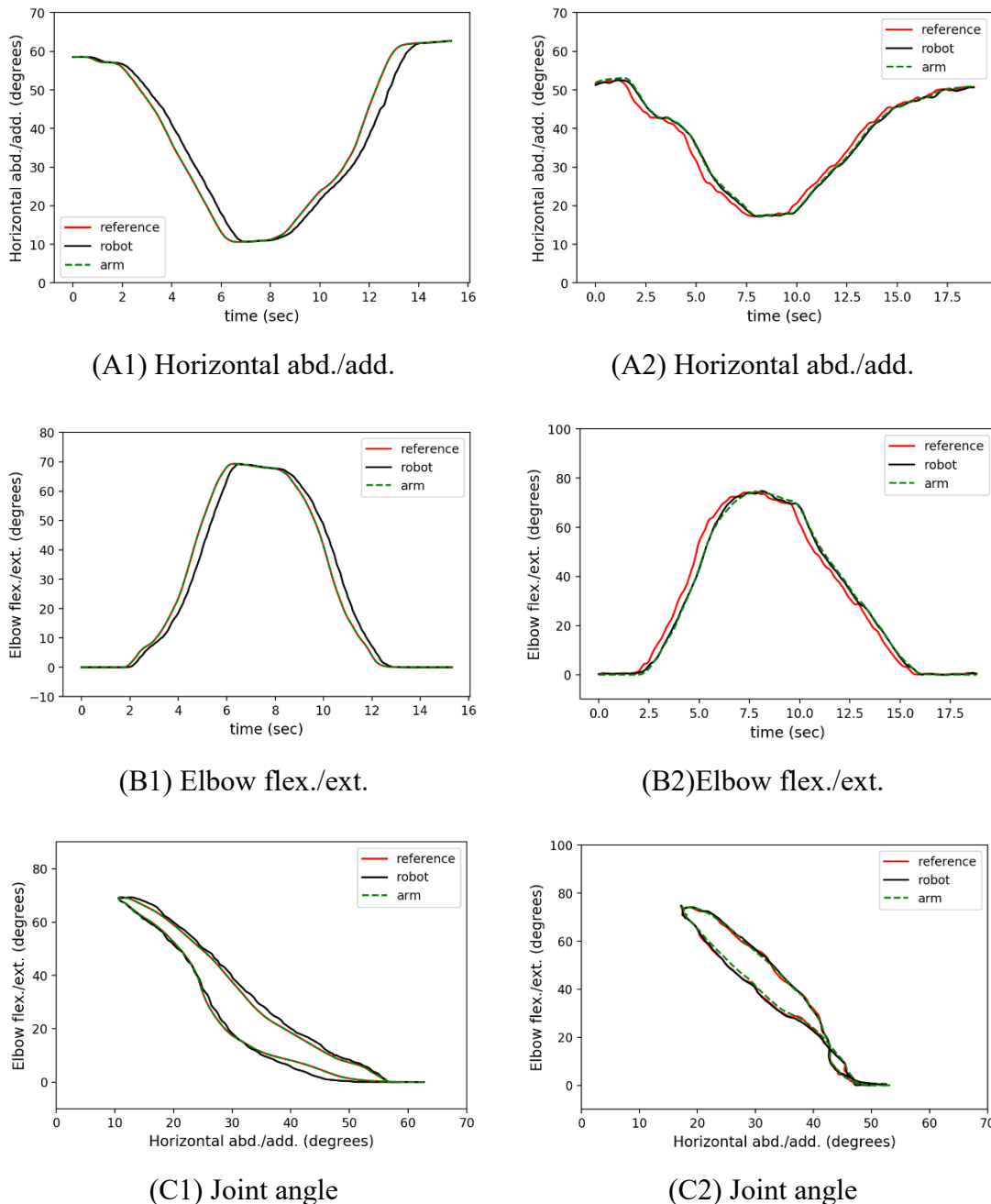


Fig. 4.7 Real arm angle (green dashed line), predict joint angle as reference trajectory (red solid line) joint angle and robot joint angle (black solid line) of IMU control (left column) and MS-LSTM Dueling model control (right column) for feeding (two-joint) task in bilateral mode.

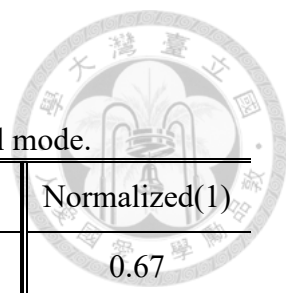
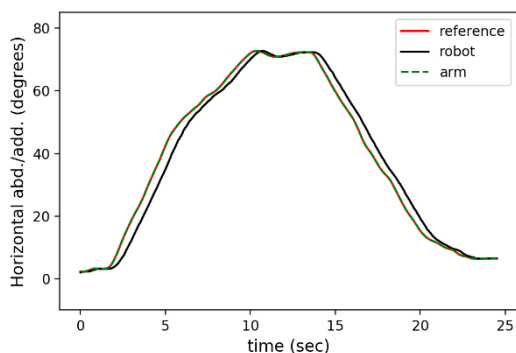


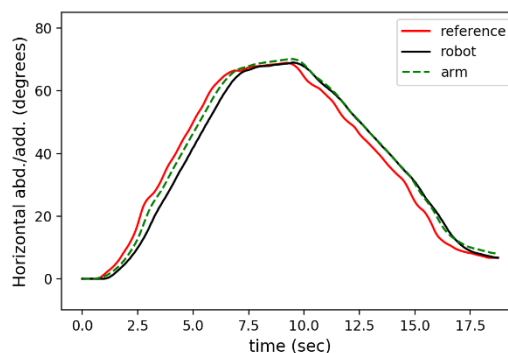
TABLE 4.7 Evaluation results of feeding task in bilateral mode.

Feeding	Index	IMU	Proposed	Normalized(1)
Horizontal abd./add.	MAE (degrees)	1.91°	1.28°	0.67
	DT (sec)	0.376	0.033	0.09
	DLJ (10 ⁷)	9.64	11.18	1.16
Elbow flex./ext.	MAE (degrees)	3.33°	1.35°	0.41
	DT (sec)	0.338	0.049	0.14
	DLJ (10 ⁷)	9.19	11.30	1.23
Average	MAE (degrees)	2.62°	1.32°	0.50
	DT (sec)	0.357	0.041	0.11
	DLJ (10 ⁷)	9.42	11.24	1.19

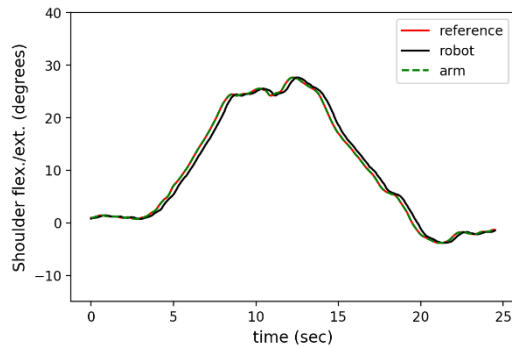
Fig. 4.8 (A)-(B) illustrate each joint trajectory in greeting task. Because we allow the subjects to move completely free, there are some deviations in the trajectory. We can see that both IMU based control and the proposed method have a slight concussion on the trajectory because there's a little bit of shake in the arm when subjects are completely free to move.



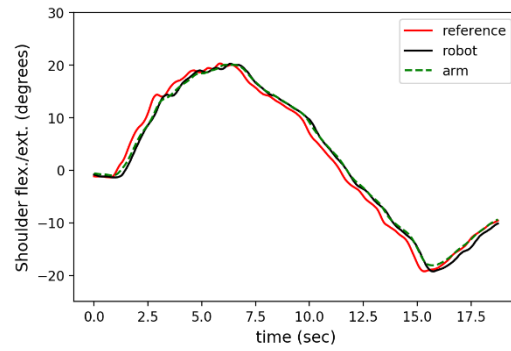
(A1) Horizontal abd./add.



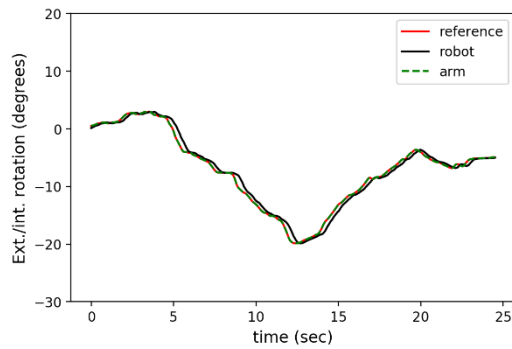
(A2) Horizontal abd./add.



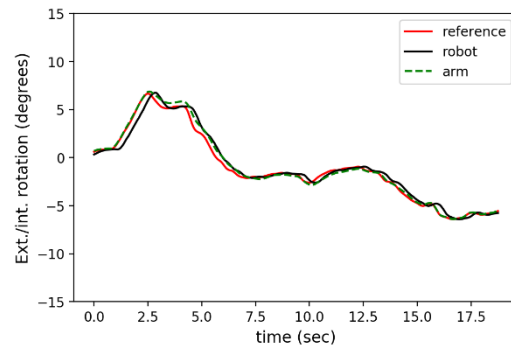
(B1) Shoulder flex./ext.



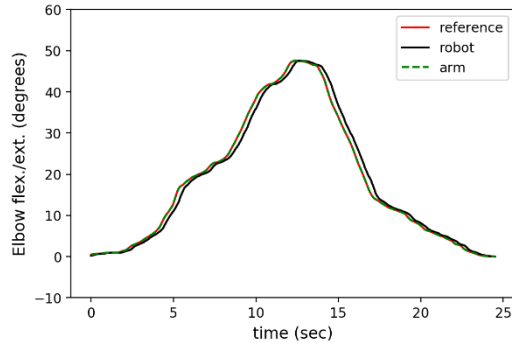
(B2) Shoulder flex./ext.



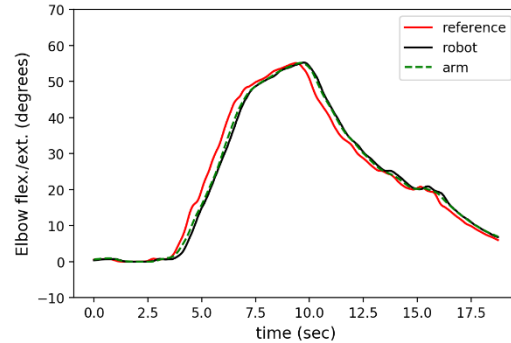
(C1) Ext./int. rotation



(C2) Ext./int. rotation



(D1) Elbow flex./ext.



(D2) Elbow flex./ext.

Fig. 4.8 Real arm angle (green dashed line), predict joint angle as reference trajectory (red solid line) joint angle and robot joint angle (black solid line) of IMU control (left column) and MS-LSTM Dueling model control (right column) for greeting (four-joint) task in bilateral mode.

TABLE 4.8 shows the average evaluation results of all subjects for greeting (four-joint) task. When all the joints are moving together, the model also shows a better output stability. This is because, in the model training stage, we do the global optimization for the four outputs at the same time. So in the real-time experiment, four joint movements

can also have a better performance.

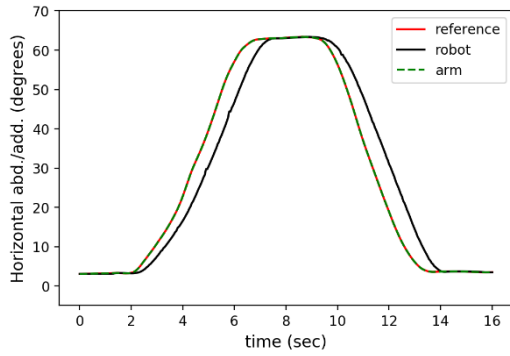
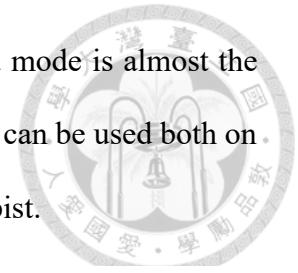
TABLE 4.8 Evaluation results of greeting task in bilateral mode.

Greeting	Index	IMU	Proposed	Normalized(1)
Horizontal abd./add.	MAE (degrees)	3.26°	1.17°	0.36
	DT (sec)	0.473°	0.132°	0.28
	DLJ (10 ⁷)	15.60	22.91	1.47
Shoulder flex./ext.	MAE (degrees)	0.8°	0.66°	0.83
	DT (sec)	0.274	0.041	0.15
	DLJ (10 ⁷)	23.04	26.44	1.15
Ext./int. rotation	MAE (degrees)	0.44°	0.42°	0.95
	DT (sec)	2.27	0.801	0.35
	DLJ (10 ⁷)	24.40	27.96	1.15
Elbow flex./ext.	MAE (degrees)	1.54°	0.98°	0.64
	DT (sec)	0.325	0.04	0.12
	DLJ (10 ⁷)	8.06	9.03	1.12
Average	MAE (degrees)	1.51°	0.8075°	0.53
	DT (sec)	0.8355	0.2535	0.30
	DLJ (10 ⁷)	17.69	21.59	1.22

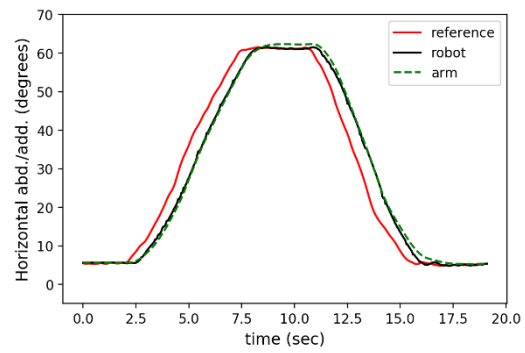
4.4.2 Performance for Lead Mode Exercise

Fig. 4.9-Fig. 4.11 shows the joint angle comparison between experiments performed by IMU based control and MS-LSTM Dueling model based control in lead mode. The layout of the figures is the same as the previous bilateral mode. In lead mode, we want to verify that MS-LSTM Dueling model can also be applied to the same side arm (left arm) of the therapist, driving the robot arm movement. The movement pattern shown in the figures are very similar to bilateral mode. The detail of evaluation index values is shown

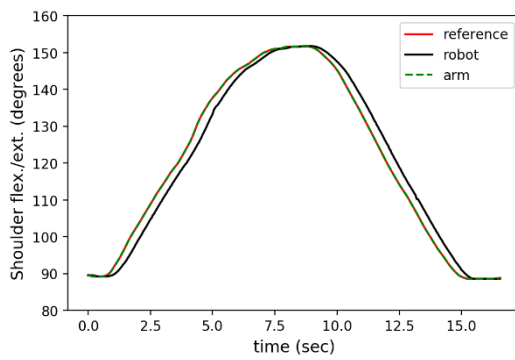
in TABLE 4.9-TABLE 4.10. It is clear that the performance in lead mode is almost the same as in bilateral mode, which proves that the method we propose can be used both on the patient's contralateral arm and on the same side arm of the therapist.



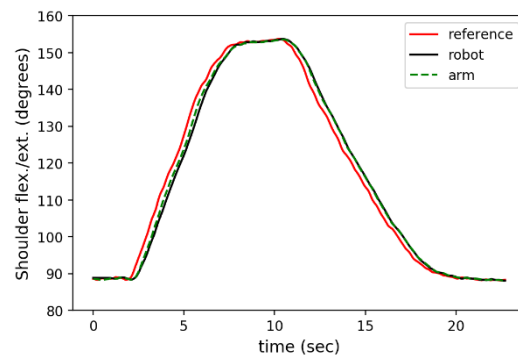
(A1) Horizontal abd./add.



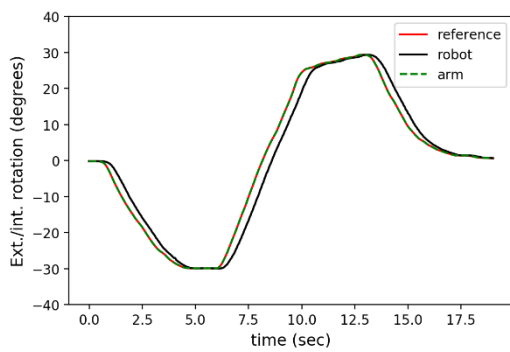
(A2) Horizontal abd./add.



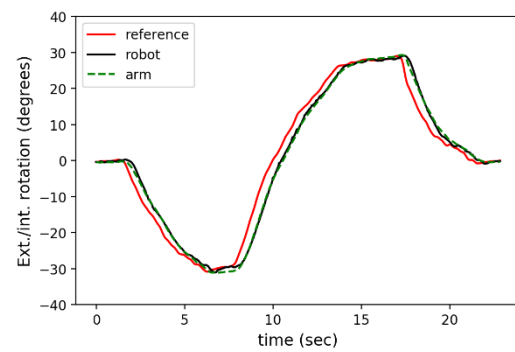
(B1) Shoulder flex./ext.



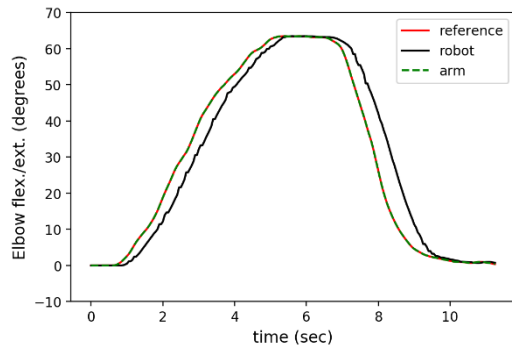
(B2) Shoulder flex./ext.



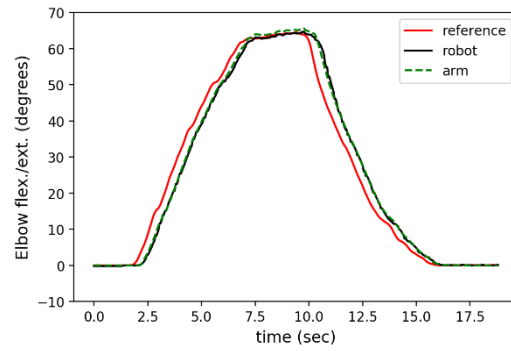
(C1) Ext./int. rotation



(C2) Ext./int. rotation



(D1) Elbow flex./ext.



(D2) Elbow flex./ext.

Fig. 4.9 Real arm angle (green dashed line), predict joint angle as reference trajectory (red solid line) joint angle and robot joint angle (black solid line) of IMU control (left column) and MS-LSTM Dueling model control (right column) for single-joint tasks in lead mode.

TABLE 4.9 shows the detail of average evaluation results of all subjects for single-joint tasks. The results are close to bilateral mode.

TABLE 4.9 Evaluation results of single-joint tasks in lead mode.

Single Joint	Index	IMU	Proposed	Normalized(1)
Horizontal abd./add.	MAE (degrees)	2.95	1.13	0.38
	DT (sec)	0.446	0.169	0.38
	DLJ (10^7)	6.04	6.73	1.11
Shoulder flex./ext.	MAE (degrees)	2.45	1.08	0.44
	DT (sec)	0.351	0.149	0.42
	DLJ (10^7)	19.09	19.81	1.04
Ext./int. rotation	MAE (degrees)	2.2	0.83	0.38
	DT (sec)	3.355	0.354	0.11
	DLJ (10^7)	22.88	24.13	1.05
Elbow flex./ext.	MAE (degrees)	3.2	1.09	0.34
	DT (sec)	0.323	0.179	0.55
	DLJ (10^7)	2.84	3.26	1.15

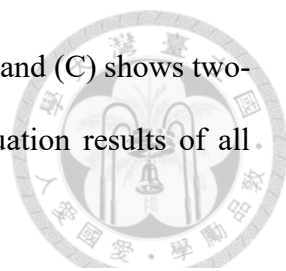
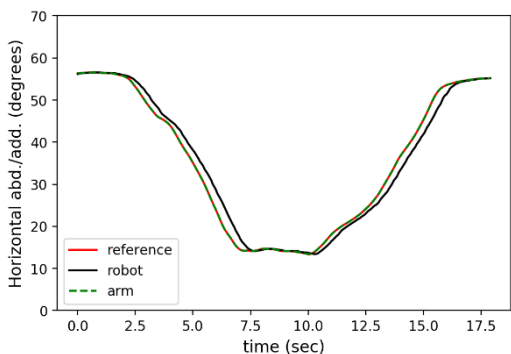
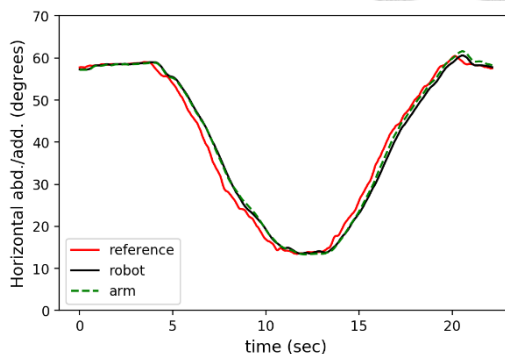


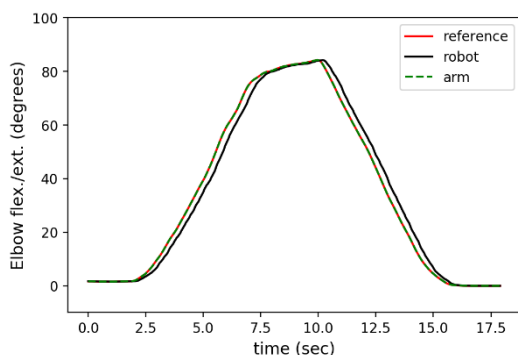
Fig. 4.10 (A) - (B) illustrate each joint trajectory in feeding task and (C) shows two-joint trajectory in 2D-space. TABLE 4.10 shows the average evaluation results of all subjects for feeding task in lead mod.



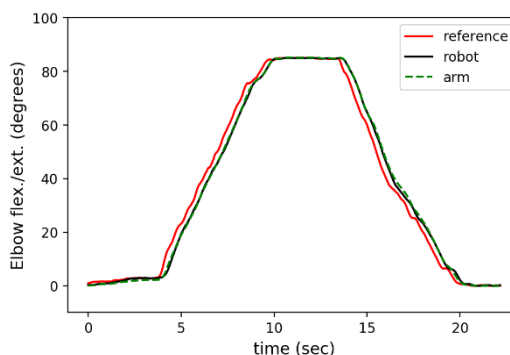
(A1) Horizontal abd./add.



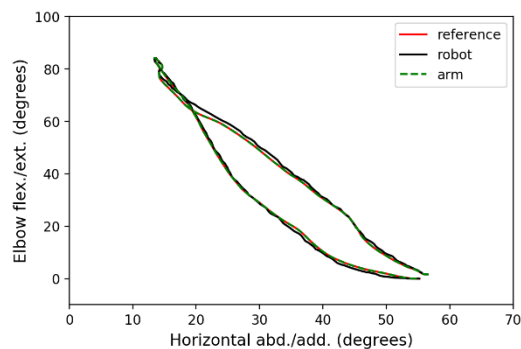
(A2) Horizontal abd./add.



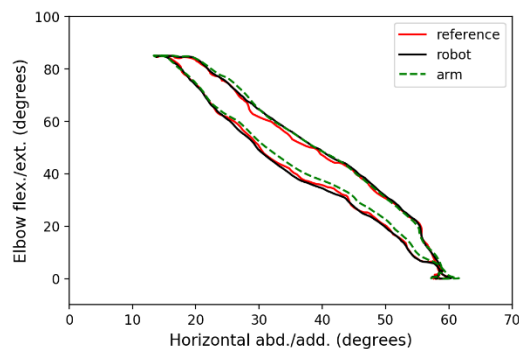
(B1) Elbow flex./ext.



(B2) Elbow flex./ext.



(C1) Joint angle



(C2) Joint angle

Fig. 4.10 Real arm angle (green dashed line), predict joint angle as reference trajectory (red solid line) joint angle and robot joint angle (black solid line) of IMU control (left column) and MS-LSTM Dueling model control (right column) for feeding (two-joint) task in lead mode.

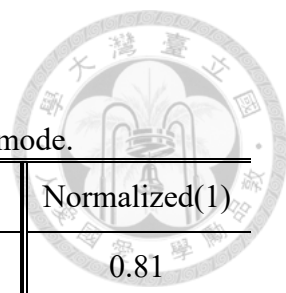
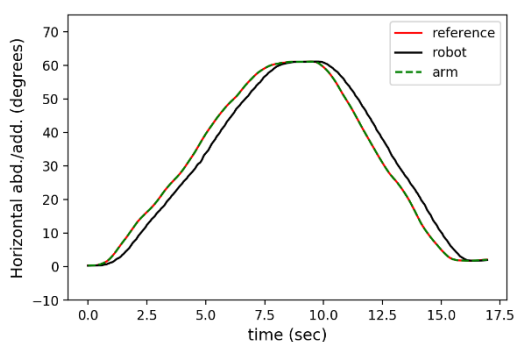


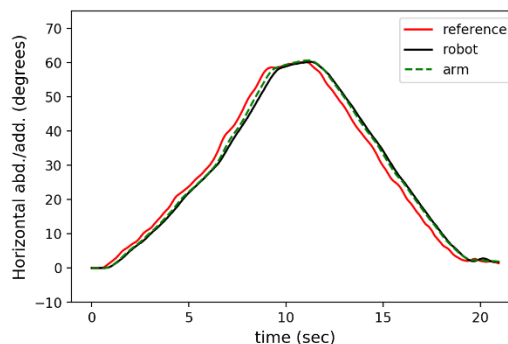
TABLE 4.10 Evaluation results of feeding task in lead mode.

Feeding	Index	IMU	Proposed	Normalized(1)
Horizontal abd./add.	MAE (degrees)	2.37	1.91	0.81
	DT (sec)	0.416	0.068	0.16
	DLJ (10^7)	14.24	15.86	1.11
Elbow flex./ext.	MAE (degrees)	3.88	1.74	0.45
	DT (sec)	0.355	0.042	0.12
	DLJ (10^7)	13.44	16.28	1.21
Average	MAE (degrees)	3.12	1.83	0.59
	DT (sec)	0.385	0.055	0.14
	DLJ (10^7)	13.84	16.07	1.16

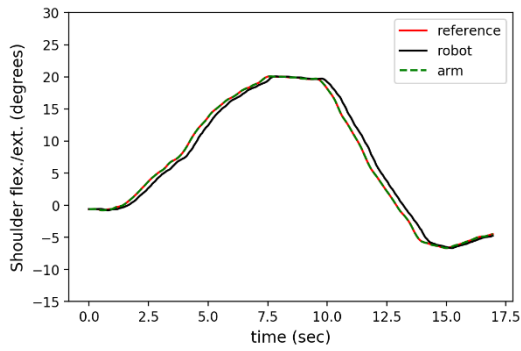
Fig. 4.11 shows each joint trajectory in greeting task. By comparing IMU based control (left column) with MS-LSTM Dueling model based control (right column), we can see that the output of the proposed model has some turbulence. As in bilateral mode, the arm itself has a little shock when it's doing completely free motion, and it's also not easy to stabilize in lead mode with a non-habitual hand. Moreover, the model has a little turbulence itself, as shown in the offline result.



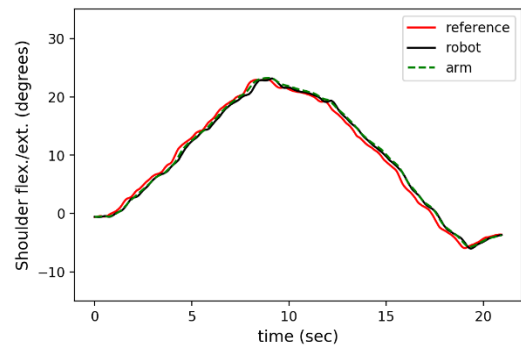
(A1) Horizontal abd./add.



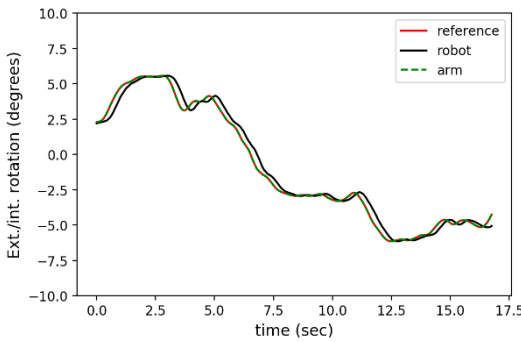
(A2) Horizontal abd./add.



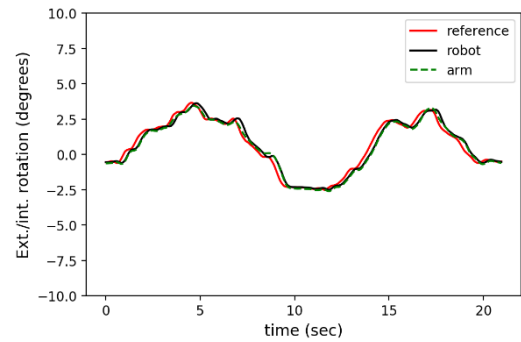
(B1) Shoulder flex./ext.



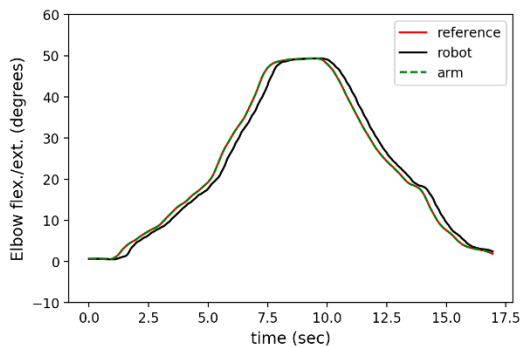
(B2) Shoulder flex./ext.



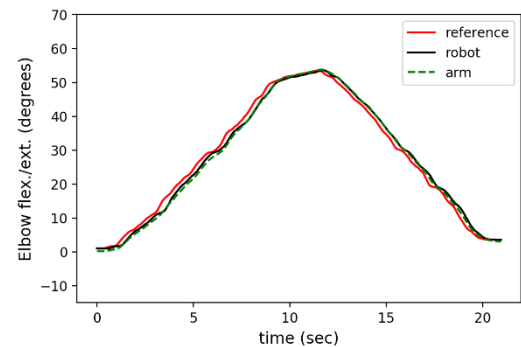
(C1) Ext./int. rotation



(C2) Ext./int. rotation



(D1) Elbow flex./ext.



(D2) Elbow flex./ext.

Fig. 4.11 Real arm angle (green dashed line), predict joint angle as reference trajectory (red solid line) joint angle and robot joint angle (black solid line) of IMU control (left column) and MS-LSTM Dueling model control (right column) for greeting (four-joint) task in lead mode.

TABLE 4.11 shows detailed result of average evaluation index values for greeting task. The MAE of the proposed method reduces at least 50% compare with IMU based control and DT of the proposed model reduces at least 70% compare with IMU based control. Additionally, the DLJ index of IMU control is only higher 24%, which is hard to

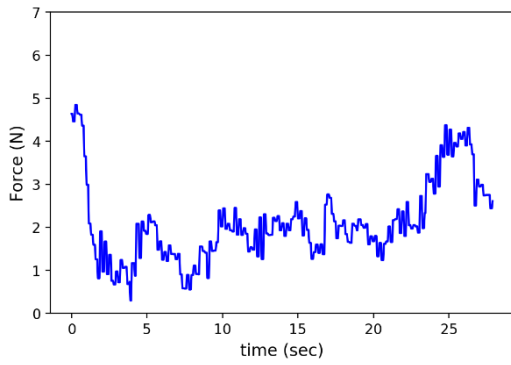
feel the difference for the subject.

TABLE 4.11 Evaluation results of greeting task in lead mode.

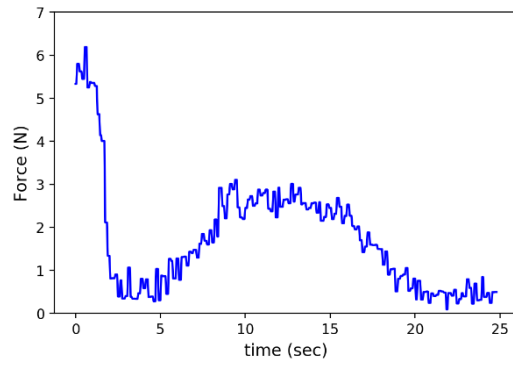
Greeting	Index	IMU	Proposed	Normalized(1)
Horizontal abd./add.	MAE (degrees)	3.31	1.43	0.43
	DT (sec)	0.473	0.161	0.34
	DLJ (10^7)	13.37	15.16	1.13
Shoulder flex./ext.	MAE (degrees)	0.725	0.345	0.48
	DT (sec)	0.271	0.02	0.07
	DLJ (10^7)	16.68	17.82	1.07
Ext./int. rotation	MAE (degrees)	0.74	0.56	0.76
	DT (sec)	2.218	0.947	0.43
	DLJ (10^7)	15.11	19.34	1.28
Elbow flex./ext.	MAE (degrees)	1.48	0.63	0.42
	DT (sec)	0.303	0.009	0.03
	DLJ (10^7)	77.59	100.24	1.29
Average	MAE (degrees)	1.56	0.74	0.47
	DT (sec)	0.82	0.28	0.35
	DLJ (10^7)	30.68	38.14	1.24

From the bilateral and lead experiment above, we can conclude that using the MS-LSTM Dueling model to control the robotic arm in guide mode can achieve higher synchronization with the human arm, and have strong stability.

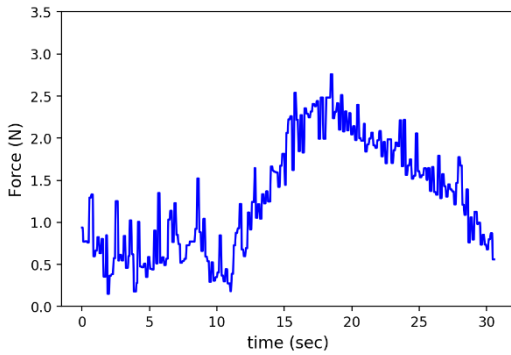
4.4.3 Performance for Active Mode Exercise



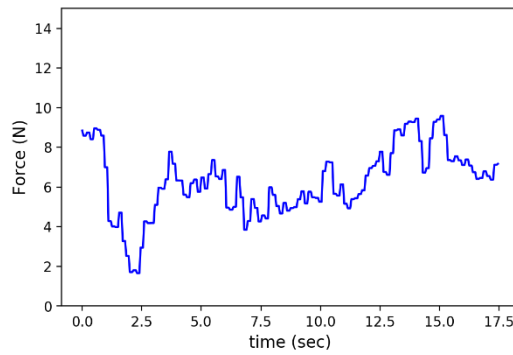
(A) Horizontal abd./add.



(B) Shoulder flex./ext.

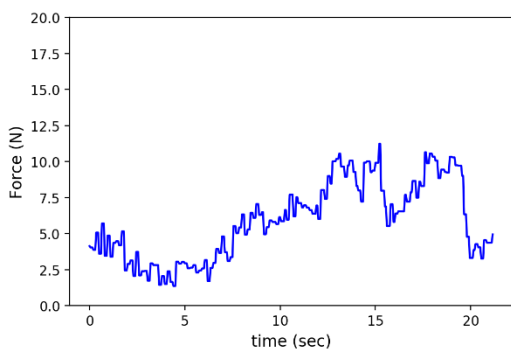


(C) Ext./int. rotation

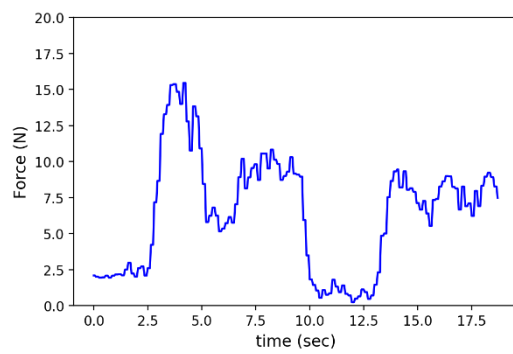


(D) Elbow flex./ext.

Fig. 4.12 End-point interactive force for single-joint tasks in active mode.

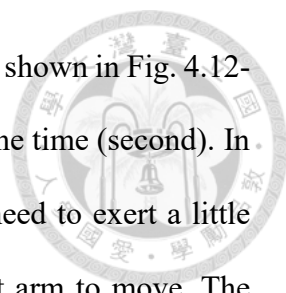


(A) Feeding



(B) Greeting

Fig. 4.13 End-point interactive force for multi-joint tasks (feeding and greeting) in active mode.



The result of single-joint and multi-joint tasks in active mode are shown in Fig. 4.12- Fig. 4.13. The y-axis is interactive force (Newton) and the x-axis is the time (second). In the single-joint task, except ext./int. rotation, the other three tasks need to exert a little force at start to make the model output future position let the robot arm to move. The force is very small during motion stage. At the end-point, the subject still needs to exert force to hold arm stop in the air, so in the figures, there is no zero force except for the greeting task. In the greeting task, the F/T sensor will be perpendicular to the hand at the end point, and will not receive the force exerted by the hand. It has shown in Fig. 4.13 (B) at time 10.5s – 12.5s. The force exerted in the multi-joint task is larger than that in the single-joint task, due to we have to exert force to multi-directions.

The average evaluation results of all subjects in active mode are shown in TABLE 4.12 and TABLE 4.13. The average interactive force during moving period has signification improvement in both single-joint and multi-joint tasks. Our proposed method reduces 40% interactive force compare with MIRM based control and reduces 60% interactive force compare with F/T sensor based control. The performance of proposed active control method outperforms the F/T sensor based active control method in either moving or static period. The standard deviation of joint angle during static period of the proposed method is a little higher than that of MIRM control. Because our method considers IMU singles which is very sensitive to small activities and there's a small amount of vibration in the arm when human moving their arm. Moreover, MIRM model only considers EMG signals, it only gives human arm moving directions. Those cause our method not as good as MIRM but the values are still in the acceptable range.

In the greeting task, the F/T control and MIMR can only control two-joint at the same time, the subject cannot control each joint independently. Our proposed method can control four-joint independently which is more fitting human's normal movement and

experiment results are even better.

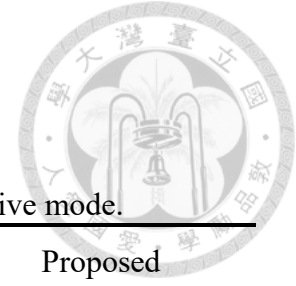


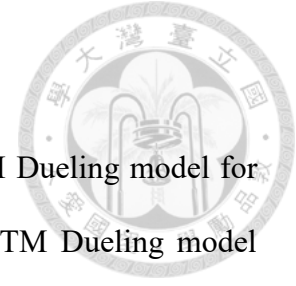
TABLE 4.12 Evaluation results of single-joint tasks in active mode.

	F/T sensor		MIRM		Proposed	
	$avg(F_{mo})$	σ_{st}	$avg(F_{mo})$	σ_{st}	$avg(F_{mo})$	σ_{st}
Single joint						
Horizontal abd./add.	3.52	1.46	3.83	0.36	1.89	0.31
Shoulder flex./ext.	12.75	1.41	6.90	0.23	2.86	0.56
Ext./int. rotation	--	--	--	--	1.86	0.78
Elbow flex./ext.	12.71	2.82	10.98	0.64	6.79	0.63
Average	9.66	1.89	7.24	0.41	3.85	0.50
Normalized	0.40	0.26	0.53	1.22	1	1

TABLE 4.13 Evaluation results of multi-joint tasks in active mode.

	F/T sensor (two-joint)		MIRM (two-joint)		Proposed	
	$avg(F_{mo})$	σ_{st}	$avg(F_{mo})$	σ_{st}	$avg(F_{mo})$	σ_{st}
Feeding	6.87	2.62	12.03	0.40	7.85	0.50
Normalized	1.14	0.19	0.65	1.25	1	1
Greeting	14.61	1.78	8.89	0.46	5.54	0.65
Normalized	0.38	0.37	0.62	1.41	1	1

Chapter 5 Conclusion



In this thesis, motion prediction based on Multi-Stream LSTM Dueling model for control of upper limb exoskeleton robot is proposed. The MS-LSTM Dueling model predicts the joint angles of the human arm to be moving, and then input the angles to the controller such that the moving of the robotic arm is synchronized with the arm. To achieve accurate predictions, our model requires two inputs, human arm dynamics and EMG features. In particular, we have designed the internal architecture of the model to deal with the different characteristics of the input signal. In order to get accurate angles of the current human arm joints, we design a complementary filter and the compensation for muscle deformation. EMG features are obtained by short-time Fourier transform and the preprocessed EMG signals are recorded, which can be used for subsequent evaluation and analysis. We provide active mode and guide mode which include bilateral and lead mode control on exoskeleton robot.

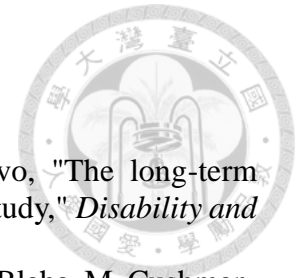
In the related work, the EMG based control methods have large estimation errors, long setup time and is only available for single-joint or coupled multi-joint (two joints moves dependently) movement. Besides, the F/T sensor based control methods have disadvantages that the user should exert the interactive force on the location of the sensor and it is difficult to model the mapping from sensor values to human intention. The proposed model can predict the user's intended motion position. Because we use convolutional layer to extract higher EMG patterns, LSTM layer to extract time-sequence features and do end-to-end global optimization, our model can predict the four joint angles of the human arm at the same time and can conquer some changes in EMG patterns. We use a large number of data from different subjects to train the model. So, even when a new subject came in, they could still directly do the rehabilitation therapy. As a result,

the setup time before the therapy can be reduced to less than 5 minutes.

The results of offline experiments show that the MS-LSTM Dueling model has the better accuracy than other deep learning models and traditional regression models. The more different subjects' data in training data, the better result we get. But after more than 3 different subjects' data, the accuracy only improves a little. In the real-time experiment, the subjects were asked to perform the single- and multi-joint tasks in both guide mode and active mode. In the guide mode which contains bilateral and lead mode, we compare IMU control with MS-LSTM Dueling mode control. The results show that the proposed method reduces 50% of mean average error of joint angle between human arm and robot arm, and reduces 70% of delay time to allow users to feel better human-robot coordination. The performance of standard deviation of joint angle during static period has slightly decreased but still in the acceptable range. In active mode, compared to the MIRM based control, the proposed method 40% of interactive force is decreased. We improve 60% of stability during static period compared with F/T sensor based control but slightly decreased compare with MIRM based control. Nevertheless, it is still in an acceptable range and subjects rarely to feel the difference.

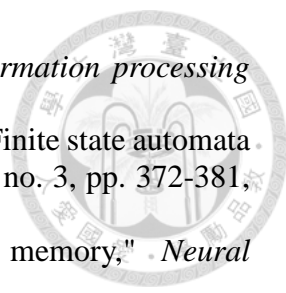
This work allows patients to take voluntary exercises in either bilateral mode or active mode rehabilitation therapy. It also allows therapists to assist in training patients in lead mode. In addition, it is easy to extend to different types of rehabilitation tasks in the future. The proposed method can be integrated not only in NTUH-II but also in any robot arm that achieves multi-joint movement independently.

REFERENCE



- [1] J. G. BROEKS, G. Lankhorst, K. Rumping, and A. Prevo, "The long-term outcome of arm function after stroke: results of a follow-up study," *Disability and rehabilitation*, vol. 21, no. 8, pp. 357-364, 1999.
- [2] D. Mozaffarian, E. J. Benjamin, A. S. Go, D. K. Arnett, M. J. Blaha, M. Cushman, S. R. Das, S. De Ferranti, J. P. Després, and H. J. Fullerton, "Executive summary: heart disease and stroke statistics-2016 update: a report from the American Heart Association," *Circulation*, vol. 133, no. 4, pp. 447-454, 2016.
- [3] H. Veeger and F. Van Der Helm, "Shoulder function: the perfect compromise between mobility and stability," *Journal of biomechanics*, vol. 40, no. 10, pp. 2119-2129, 2007.
- [4] J. McCabe, M. Monkiewicz, J. Holcomb, S. Pundik, and J. J. Daly, "Comparison of robotics, functional electrical stimulation, and motor learning methods for treatment of persistent upper extremity dysfunction after stroke: a randomized controlled trial," *Archives of physical medicine and rehabilitation*, vol. 96, no. 6, pp. 981-990, 2015.
- [5] J. H. Cauraugh and J. J. Summers, "Neural plasticity and bilateral movements: a rehabilitation approach for chronic stroke," *Progress in neurobiology*, vol. 75, no. 5, pp. 309-320, 2005.
- [6] K. C. Stewart, J. H. Cauraugh, and J. J. Summers, "Bilateral movement training and stroke rehabilitation: a systematic review and meta-analysis," *Journal of the neurological sciences*, vol. 244, no. 1, pp. 89-95, 2006.
- [7] J. Manyika, M. Chui, B. Brown, J. Bughin, R. Dobbs, C. Roxburgh, and A. H. Byers, "Big data: The next frontier for innovation, competition, and productivity," 2011.
- [8] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, p. 436, 2015.
- [9] S. P. Swinnen, "Intermanual coordination: from behavioural principles to neural-network interactions," *Nature Reviews Neuroscience*, vol. 3, no. 5, p. 348, 2002.
- [10] S. M. Waller, M. Harris-Love, W. Liu, and J. Whittall, "Temporal coordination of the arms during bilateral simultaneous and sequential movements in patients with chronic hemiparesis," *Experimental brain research*, vol. 168, no. 3, pp. 450-454, 2006.
- [11] S. M. Waller and J. Whittall, "Bilateral arm training: why and who benefits?," *NeuroRehabilitation*, vol. 23, no. 1, pp. 29-41, 2008.
- [12] R. Gopura, D. Bandara, K. Kiguchi, and G. K. Mann, "Developments in hardware systems of active upper-limb exoskeleton robots: A review," *Robotics and Autonomous Systems*, vol. 75, pp. 203-220, 2016.
- [13] M. Rahman, M. Rahman, M. Saad, C. Ochoa-Luna, S. Ferrer, and P. Archambault, "Control of an upper extremity exoskeleton robot to provide active assistive therapy," in *Modelling, Identification & Control (ICMIC), 2013 Proceedings of International Conference on*, 2013, pp. 105-110: IEEE.
- [14] A. M. Khan, D.-w. Yun, M. A. Ali, J. Han, K. Shin, and C. Han, "Adaptive impedance control for upper limb assist exoskeleton," in *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, 2015, pp. 4359-4366: IEEE.
- [15] H.-Y. Li, L.-Y. Chien, H.-Y. Hong, S.-H. Pan, C.-L. Chiao, H.-W. Chen, L.-C. Fu, and J.-S. Lai, "Active control with force sensor and shoulder circumduction

- implemented on exoskeleton robot NTUH-II," in *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on*, 2016, pp. 2406-2411: IEEE.
- [16] D. Leonardis, M. Barsotti, C. Loconsole, M. Solazzi, M. Troncossi, C. Mazzotti, V. P. Castelli, C. Procopio, G. Lamola, and C. Chisari, "An EMG-controlled robotic hand exoskeleton for bilateral rehabilitation," *IEEE transactions on haptics*, vol. 8, no. 2, pp. 140-151, 2015.
- [17] Z. O. Khokhar, Z. G. Xiao, and C. Menon, "Surface EMG pattern recognition for real-time control of a wrist exoskeleton," *Biomedical engineering online*, vol. 9, no. 1, p. 41, 2010.
- [18] T. Lenzi, S. M. M. De Rossi, N. Vitiello, and M. C. Carrozza, "Intention-based EMG control for powered exoskeletons," *IEEE transactions on biomedical engineering*, vol. 59, no. 8, pp. 2180-2190, 2012.
- [19] J. Vogel, C. Castellini, and P. van der Smagt, "EMG-based teleoperation and manipulation with the DLR LWR-III," in *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, 2011, pp. 672-678: IEEE.
- [20] T. Kawase, T. Sakurada, Y. Koike, and K. Kansaku, "Estimating joint angles from biological signals for multi-joint exoskeletons," in *Systems, Man and Cybernetics (SMC), 2014 IEEE International Conference on*, 2014, pp. 1470-1474: IEEE.
- [21] W.-W. Wang and L.-C. Fu, "Mirror therapy with an exoskeleton upper-limb robot based on IMU measurement system," in *Medical Measurements and Applications Proceedings (MeMeA), 2011 IEEE International Workshop on*, 2011, pp. 370-375: IEEE.
- [22] D. Blana, T. Kyriacou, J. M. Lambrecht, and E. K. Chadwick, "Feasibility of using combined EMG and kinematic signals for prosthesis control: a simulation study using a virtual reality environment," *Journal of Electromyography and Kinesiology*, vol. 29, pp. 21-27, 2016.
- [23] C. H. Lin, W. M. Lien, W. W. Wang, S. H. Chen, C. H. Lo, S. Y. Lin, L. C. Fu, and J. S. Lai, "NTUH-II robot arm with dynamic torque gain adjustment method for frozen shoulder rehabilitation," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2014, pp. 3555-3560.
- [24] C. A. Doorenbosch, A. J. Mourits, D. H. Veeger, J. Harlaar, and F. C. van der Helm, "Determination of functional rotation axes during elevation of the shoulder complex," *Journal of Orthopaedic & Sports Physical Therapy*, vol. 31, no. 3, pp. 133-137, 2001.
- [25] P. W. McClure, L. A. Michener, B. J. Sennett, and A. R. Karduna, "Direct 3-dimensional measurement of scapular kinematics during dynamic movements in vivo," *Journal of Shoulder and Elbow Surgery*, vol. 10, no. 3, pp. 269-277, 2001.
- [26] M. J. Bey, S. K. Kline, R. Zauel, T. R. Lock, and P. A. Kolowich, "Measuring dynamic in-vivo glenohumeral joint kinematics: technique and preliminary results," *Journal of biomechanics*, vol. 41, no. 3, pp. 711-714, 2008.
- [27] I. a. A. Lab. (2012). "Normal ROM,". Available: <https://assessmentandinterventiongroup8.wordpress.com/rom/normal-rom/>.
- [28] W. T. Higgins, "A comparison of complementary and Kalman filtering," *IEEE Transactions on Aerospace and Electronic Systems*, no. 3, pp. 321-325, 1975.
- [29] S. R. Gunn, "Support vector machines for classification and regression," *ISIS technical report*, vol. 14, no. 1, pp. 5-16, 1998.
- [30] N. S. Altman, "An introduction to kernel and nearest-neighbor nonparametric regression," *The American Statistician*, vol. 46, no. 3, pp. 175-185, 1992.
- [31] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep

- 
- convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097-1105.
- [32] A. Cleeremans, D. Servan-Schreiber, and J. L. McClelland, "Finite state automata and simple recurrent networks," *Neural computation*, vol. 1, no. 3, pp. 372-381, 1989.
- [33] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735-1780, 1997.
- [34] A. Graves, "Generating sequences with recurrent neural networks," *arXiv preprint arXiv:1308.0850*, 2013.
- [35] P. J. Werbos, "Backpropagation through time: what it does and how to do it," *Proceedings of the IEEE*, vol. 78, no. 10, pp. 1550-1560, 1990.
- [36] S. Xingjian, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W.-c. Woo, "Convolutional LSTM network: A machine learning approach for precipitation nowcasting," in *Advances in neural information processing systems*, 2015, pp. 802-810.
- [37] L.-K. Liu, L.-Y. Chien, S.-H. Pan, J.-L. Ren, C.-L. Chiao, W.-H. Chen, L.-C. Fu, and J.-S. Lai, "Interactive torque controller with electromyography intention prediction implemented on exoskeleton robot NTUH-II," in *Robotics and Biomimetics (ROBIO), 2017 IEEE International Conference on*, 2017, pp. 1485-1490: IEEE.
- [38] R. H. Chowdhury, M. B. Reaz, M. A. B. M. Ali, A. A. Bakar, K. Chellappan, and T. G. Chang, "Surface electromyography signal processing and classification techniques," *Sensors*, vol. 13, no. 9, pp. 12431-12466, 2013.
- [39] M. Solomonow, R. Baratta, M. Bernardi, B. Zhou, Y. Lu, M. Zhu, and S. Acierno, "Surface and wire EMG crosstalk in neighbouring muscles," *Journal of Electromyography and Kinesiology*, vol. 4, no. 3, pp. 131-142, 1994.
- [40] P. Xia, J. Hu, and Y. Peng, "EMG-Based Estimation of Limb Movement Using Deep Learning With Recurrent Convolutional Neural Networks," *Artificial organs*, vol. 42, no. 5, pp. E67-E77, 2018.
- [41] I. Kuzborskij, A. Gijsberts, and B. Caputo, "On the challenge of classifying 52 hand movements from surface electromyography," in *Engineering in Medicine and Biology Society (EMBC), 2012 Annual International Conference of the IEEE*, 2012, pp. 4931-4937: IEEE.
- [42] E. Arisoy and M. Saraçlar, "Multi-stream long short-term memory neural network language model," in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.
- [43] B. Singh, T. K. Marks, M. Jones, O. Tuzel, and M. Shao, "A multi-stream bi-directional recurrent neural network for fine-grained action detection," in *Computer Vision and Pattern Recognition (CVPR), 2016 IEEE Conference on*, 2016, pp. 1961-1970: IEEE.
- [44] Z. Wang, T. Schaul, M. Hessel, H. Van Hasselt, M. Lanctot, and N. De Freitas, "Dueling network architectures for deep reinforcement learning," *arXiv preprint arXiv:1511.06581*, 2015.
- [45] P. Indyk and R. Motwani, "Approximate nearest neighbors: towards removing the curse of dimensionality," in *Proceedings of the thirtieth annual ACM symposium on Theory of computing*, 1998, pp. 604-613: ACM.
- [46] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.