R2Gmotion: an IMU-based Approach for Controlling Transradial Prostheses by Learning the Kinematics of Reach-to-Grasp Movement

Shunit Polinsky

R2Gmotion:
an IMU-based Approach for Controlling Transradial Prostheses by Learning the Kinematics of Reach-to-Grasp Movement

Research Thesis

Submitted in partial fulfillment of the requirements

for the degree of Master of Science

Shunit Polinsky

Submitted to the Senate
of the Technion — Israel Institute of Technology
Nisan 5781 Haifa March 2021

This research was carried out under the supervision of Prof. Alon Wolf, in the Faculty of Mechanical Engineering.

Some results in this thesis have been published as articles by the author and research collaborators in conferences and journals during the course of the author's master research period, the most up-to-date versions of which being:

Presented in Conferences:

Shunit Polinsky, Yair Herbst, Yoav Medan and Alon Wolf. User Interface Design for a Low-Cost 3d Printed Electro-Mechanical Prosthetic Hand. a podium presentation at the *European Society of Biomechanics* annual conference in Vienna. July 2019.

Submitted for Conference Proceeding:

Yair Herbst, Shunit Polinsky, Anath Fischer, Yoav Medan, Ronit Schneor, Joshua Kahn, and Alon Wolf. Scan-Driven Fully-Automated Pipeline for a Personalized, 3D Printed Low-Cost Prosthetic Hand. *IEEE International Conference on Automation Science and Engineering (CASE)*

In preparation for Manuscript Publication:

Shunit Polinsky, Arielle Fischer, and Alon Wolf. R2Gmotion: an IMU-based Approach for Controlling Transradial Prostheses by Learning the Kinematics of Reach-to-Grasp Movement. *sensors*

The generous financial help of the Technion is gratefully acknowledged.

Table of Contents

List of Figures

List of Tables

Abstract

The human body is a very efficient system comprised of endless different components that are working together in inspiring coordination and synergy. Recruiting this principle for the natural reach-to-grasp (R2G) movement is the cornerstone of this work towards the development of an intelligent prosthetic hand that can potentially learn the body language of R2G patterns and applies a smoother and more dynamic way of action. In comparison to previous studies, here we acquired and decoded the time-dependent R2G movement, rather than classify several discrete need-to-learn user gestures as most control strategies are based on. Our hypothesis is that grasp intention can be predicted and can be distinguished from other hand movements in people with transradial amputation by analyzing the kinematics of the upper-body segments, thanks to the complexity and synergy of the R2G movement. Since the end-goal is to control transradial prostheses we assumed we do not have any kinematic data from below the elbow. For this purpose, we developed a low-cost wearable IMU-based system that acquires the upper limb and torso kinematics, not including pronation-supination of the palm. Ten healthy participants were recorded on two different days by our system, as well as by an optoelectronic system for validation. Each participant accomplished different R2G tasks of different objects from different and random places at random times according to visual and vocal instructions managed by the application we built, while sitting and standing (total of ~3.5k R2G movements). Also, the participants were recorded while walking and talking to mimic other hand movements to validate the robustness of our algorithm when differentiating R2G movement from other tasks. The data analysis was done by comparing different types of machine learning architectures and different input data types such as varied time-window samples and kinematics characteristics. The selected machine learning model is a combination of a classifier that detects if an R2G movement is currently occurring (median results: 89.25% sensitivity and 85.28% specificity) and a regression estimator that estimates the current stage of the R2G movement process (median results: 0.031 mean-square error, for an output range of [0,1]). According to the model outputs and customized parameters, the system identifies the grasp moment (for selected parameters: 1.5% false positive, and 92.8% true positive from which 23.8% considered as early-detection) and triggers the low-cost 3d printed prosthetic hand we built to accomplish the action. In conclusion, we present here a novel concept for controlling prosthetic hands that takes advantage of the arm reaching movement that naturally occurs before almost any grasp. This concept leads to a better starting point for prosthetic hand control and potentially enables a low control burden for sensors-fused-systems. Our *R2Gmotion*, a wearable, real-time system based on low-cost and accessible components, ready to be implemented, can be easily

reproduced and be used as a stand-alone control system or as a platform that other methods can be implemented on top of it to construct a robust fused control system for prosthetic hands.

Abbreviations and Notations

ADL	:	Activity of Daily Living
BLE	:	Bluetooth Low Energy
CS	:	Coordinates System
FPR	:	False Positive Rate
GRU	:	Gated Recurrent Unit
GUI	:	Graphic User Interface
IK	:	Inverse Kinematics
IMU	:	Inertia Motion Unit
ISB	:	International Society of Biomechanics
ISB	:	International Society of Biomechanics
LSTM	:	Long-Short Term Memory
MSE	:	Mean Square Error
NI	:	National Instrument
R2G	:	Reach-to-Grasp movement
RF	:	Random Forest
RNN	:	Recurrent Neural Network
ROM	:	Range of Motion
RUL	:	Right Upper Limb model
SVM	:	Support Vector Machine
TNR	:	True Negative Rate
TPR	:	True Positive Rate
w.r.t	:	with respect to

Chapter 1

Introduction

Almost any task of the human body involves several muscles to produce coupled movements of the human body segments. Recruiting this natural synergy and coupling, we can develop much smarter prosthetic hands that can potentially predict reach-to-grasp (R2G) patterns and achieve a smoother and more dynamic way of action.

In contrast with the well-investigated gait analysis, upper body kinematics has less defined protocols due to the non-cyclic motion and the uncertainty of the starting point of the action. For instance, most stroke evaluation protocols are subjective and are not based on clear quantity measurements [1]. However, some studies have attempted to establish new protocols for different needs, such as Butler et al who established a 'Reach and Grasp Cycle' protocol which focused on evaluating the development of children with cerebral palsy [2]. Yet, as of today, there is no wide agreement of R2G movement and therefore, R2G patterns are much more difficult to normalize and test between a wide range of subjects and grasp types.

Still, our hands have a very important role in our life. It is our basic tool to interact and communicate with our environment, either with an object or with other people. Although the prosthetic hand is a well-researched field, there are still open issues from a theoretical and practical point-of-view. One of the main issues reported at Cordella et al for electric prosthetic hand is the need to increase the reliability of the hand and to reduce the dependency on visual feedback [3]. Nowadays, most commercial prosthetic hands are EMG-based [4] which is considered a high cognitive burden for many users and required relatively demanding training [5]. Moreover, for some people, the activation of the residual limb muscle provokes phantom pain or it is not stable and robust enough for EMG analysis due to muscle atrophy and muscle control loss [5], [6]. Nevertheless, the main challenge is the decoding of the user's needs and desires into commands to the prosthetic hand. Using current methods for controlling high-end prosthetic hand the user needs to: (i) approach an object, (ii) stop next to the object, (iii) adjust the hand orientation if needed, (iv) and then produce a specific signal to apply a specific grasp type. This cumbersome process is necessary if we want to separate between the human and the machine, but here we propose a completely different approach – harness the reach-to-

grasp movement, the first step in the aforementioned process, as a continuous and analog user input to result in a grasp motion at the end of this movement.

Here throughout this Thesis, we show that R2G intention can be predicted based on the kinematics of the upper limb and can be detected and distinguished from other upper limb movements. Our solution is *R2Gmotion*, a wearable, real-time system based on low-cost and accessible components, which is open source and ready to be implemented. Potentially, the proposed control approach can reduce the cognitive burden and the adaptation time while controlling prosthetic hands.

1.1 Related Work

There are several suggested mathematical models for the upper limb biomechanics of human-like motions. Some are based on optimization of the acceleration or the jerk, some on minimum energy, and some on the correlation between two different dynamic parameters such as the speed-curvature power low established by Flash et al[7][8]. There are also several studies about human intent prediction, mostly as part of improving the interface and the compatibility of a robot's behavior to its user. These studies are mostly based on an external camera that analyzes the human motion and distance from the robot as done by Hu et al[9]. They deduced the most reachable object from the user perspective and design an algorithm to assure the robot's workspace and user's workspace will not be intersecting. In contrast, on the prosthetic hand field, decoding the user's body language and in general, usage of motion sensors is rarely used. There are some studies where the motion sensors are used together with other sensors similar to Gardner et al work on a prosthetic hand which is controlled by motion sensors, electromyography sensor (EMG), and camera[10].

1.1.1 Reach-to-Grasp Movements

In contrast, with the well-investigated gait analysis field, studies of the R2G movement have gathered fewer data and have much fewer cross-disciplinary conclusions. One of the reasons is the lack of conventions regarding the starting and ending point of the movement. It is not cyclic as gait and does not have a specific way of normalization process hence makes it more difficult to apply a comparison between different studies. In general, many studies regarding stroke are investigating the R2G movement, since the rehabilitation tasks are focused on upper body movements[1][11].

From a biomechanical point of view, there are several mathematical models of general human-arm movements, such as the speed-curvature power low established by Flash et al [8]. The speed-curvature power low is a mathematical model that shows a relationship between the angular velocity and the curvature of the trajectory which is also dependent on the trajectory shape and the medium viscosity. Similarly, several studies support the theory of Listing's law constraint for reaching hand

movements which implies that during movement the axis of rotation is relatively fixed in space[12]. Well-known cost functions for the upper body motion are based on minimum jerk which implies maximum smoothness, acceleration, and other kinematic parameters. There also dynamic models based on minimum energy and rate of torque change [7].

In general, the motion capture field mostly utilizes optoelectronic markers-based systems while in biomechanics research the gold standard system is VICON™. There are many studies from different areas that are looking for different solutions to overcome the main disadvantages of this type of systems such as – cost, laboratory environment only, markers attachment and more [11], [13],[14]. For instance, Yabuki et al [13] used two force plates for motion recognition. Their system is low-cost, suit for small places and does not lead to privacy issues, as using cameras. Naturally, to validate these solutions, a comparison between a state-of-the-art system and the new solution is conducted. Bonnechère et al [11], compared their system based on Kinect™ and VICON™'s. They selected a different type of parameters than we chose and evaluated the correlation of the two systems by Pearson coefficient value. Here we perform a similar evaluation between our IMU-based system and VICON™'s and compare our results with [11] outcomes.

To enhance the development of marker-less motion capture systems, few databases were constructed and are published as open-source. One of these databases is a combination of three modalities – IMUs (XSENS™ suit), RGB camera (GoPro™), and Depth camera (SoftKinetic™). This database was collected while conducting an unstructured experiment (e.g. cooking) and includes annotation of the grasp timing and type [15]. In another study, a database of grasping from a specific position but with different orientation including IMU, EMG and a paired of video images represent the participant eye-view (camera mounted on the head) and the hand view (mounted on the grasping hand) was collected and published [16].

1.1.2 Predicting Human Motion

One of the more used methods for predicting human motion is Recurrent Neural Network (RNN) which potentially can represent the time-dependent features of human motion such as, movement synergy, kinematic derivatives and pattern characteristics [17]. There are more methods for learning the temporal characteristics of human motions such as hidden Markov Model, random forest (RF), conditional restricted Boltzman machine, Encoder-Recurrent-Decoder and more [17]. Other studies were focusing on human motion prediction for human-robot tasks, such as Hu et al [9]. According to a camera with a visual on most of the user environments, they deduce which object is the most reachable to predict the user intention. The deduction about the reachability of an object was based on intrinsic kinematic constraints of a human body, such as Range of Motion (ROM) and critical angles, and combined into the basic Inverse Kinematics (IK) for robotic. Other camera-based studies for human-robot tasks used Markov Random Fields (MRFs) model to estimate the probability of a person

to perform a specific task according to his/her body pose, distance to an object and according to the prior person's actions[18].

1.1.3 Joint Angles Estimation using IMU

IMU sensor has been used in a variety of applications, such as sport and rehabilitation, for many years. Its main advantage is that it is wearable and is not restricted to the lab environment, however, its main drawback is the need for calibration and alignment [19]. There are many different reported approaches to solve the IMU calibration and alignment problem. For gait analysis, [20] tried to solve the IMU alignment and placement-dependency problem and tested their algorithm on gait among other methods on lower limb joints. Based on their method we developed our sensor-to-body-segment calibration for the upper body joints. Although they assumed a specific orientation of the sensor w.r.t the floor (or the gravity direction) while we do not assume anything about the sensor orientation. For the upper body, an investigation of the elbow kinematic is very common due to its relatively low degrees of freedom. For instance, Muller et al suggested a calibration process that is independent of the sensor position, similar to what we developed. They estimated the two degrees of freedom of the elbow flexion and pronation-supination by forming an optimization problem and solving using gradient descent. But this solution is suitable only for joints with two degrees of freedom[19].

As part of the above solutions, a validation process between the IMU-based system and a state-of-the-art system, a marker-based system, was developed. [21] Investigated several methods to register and align IMU system and marker-based motion capture system. They have shown that angular velocity transformation and quaternion representation produce better results than the other suggested methods. [22] Reported a comparison between IMU and optoelectronic system (marker-based) for monitor surgical movement during a minimally invasive procedure. They manually aligned the IMU to the upper body segments and tested the torso, shoulder, and elbow joints with good accuracy w.r.t to the marker-based system (max error of the elbow flexion of 8.2+-2.8) and reported an underestimation of all joints angle for the IMU system.

1.1.4 Prosthetic Hands and IMU-based Systems

One of the most used interfaces for controlling prosthetic hands is the surface-EMG interface[4]. Thanks to the rapid advance in the computer vision field, many research groups worked on integrating a camera into prosthetic hands in different ways. For instance, Gardner et al [10] used an MMG - microphone-based mechanomyogram sensors for grasp activation, a webcam for object classification. So that each class was associated with a set of possible grasp patterns, and a specific grasp was selected for the relevant set according to the user's hand trajectory based on IMU data from the wrist and biceps, by applying K-Nearest-Neighbors (KNN). In this study, the wrist motion sensor was more informative than the biceps, but it is important to note that for transradial amputation this information

does not exist. Besides, [23] used a fused system of EMG and IMU to control prosthetic hands where the IMU signal was for switching between the hand modes (by recognizing arm across shoulder pose), which is very different from our concept to use the natural hand and arm movement. Not only upper body prostheses but also powered lower limb prothesis can benefit from the IMU-based control method. For example, [24] combined EMG and IMU input to improve the intention recognition of change mode for low limb prosthetic control (powered knee or ankle). For human-robot applications, [25] developed a sleeve incorporated with an EMG matrix to estimate the hand and fingers gesture type and one IMU for tracking the arm.

As mentioned before, motion prediction combined with the usage of IMU is a well-familiar regime with many applications. One of them is stroke and stroke rehabilitation which has given rise to a variety of studies. [14] Developed a wearable system, low cost of monitoring stroke rehabilitation and more specifically monitor tasks such as grasping. The system includes IMU and pressure sensor on the forearm. One of the IMU axes is aligned with the forearm longitudinal axis and is responsible to estimate the distance when the subjects held an object (from grasping to placing). Utilizing different fused-system [26] developed a wearable system including IMU and EMG to monitor and evaluate the rehabilitation process of patients after a stroke. They show the important effect of the smoothness of the movement among others to estimate the Action Research Arm Test (ARAT) clinical for upper body movement rehabilitation. Moreover, they investigated the torso flexion compensation during reaching movement and its different behavior among stroke survivors and healthy subjects. On this study as well as on many other studies, the R2G movement is only one step (and mostly the first step) in a much longer task such as pick-and-place, and the focus it not necessary on this section. [26] normalized the task by the hand trajectory, in contrast to the R2G time which is the method we used in this Thesis. Another similar work was done by [27], but in contrast to our work, they used wireless IMU and placed the IMU on the pelvis instead of the torso. They pointed out the difficulties to generalize the net results of human gesture recognition and even generalization between subjects was not trivial due to significant differences between each subject's velocities.

1.2 Research Objectives

Here, we propose our *R2Gmotion* approach – a novel control approach for transradial prostheses. By measuring the kinematics of R2G movement using IMUs attached to the upper body segments, we can predict the timing of the grasp by learning the R2G patterns. The proposed approach takes advantage of the movement that happens either way when healthy people, as well as people with transradial amputation, reach their hand to an object. Hopefully by utilizing the nature, sure-to-happen reach-to-grasp movement we would potentially reduce the cognitive burden while using prosthetic hand and improve the acceptance rate of prostheses in general.

This approach is one of the three building blocks to develop an autonomous prosthetic hand and is the cornerstone of our ongoing research on that matter. The idea is to utilize the proposed approach into a sensors-fused system consists of several different sensors to predict the grasp type and the object interaction type based on our grasp timing prediction and R2G patterns [see Figure 1].

Potentially the end-product would lead to a more natural and simple-to-learn control interface for a robotic prosthetic hand that could: (i) dynamically adjusts the orientation, (ii) selects the most suitable grasp type for the situation according to the object size, shape, and location (iii) decodes the desired interaction with the object such as pick-and-place or pick-and-drink.

In summary, the main hypothesis of this Thesis is that Reach-to-Grasp movement can be distinguished from other hand movements such as walking and talking, and its pattern can be learned so that a learning algorithm has the potential to identify the intention for grasping, to estimate the Reach-to-Grasp stage and to predict the grasp moment itself.

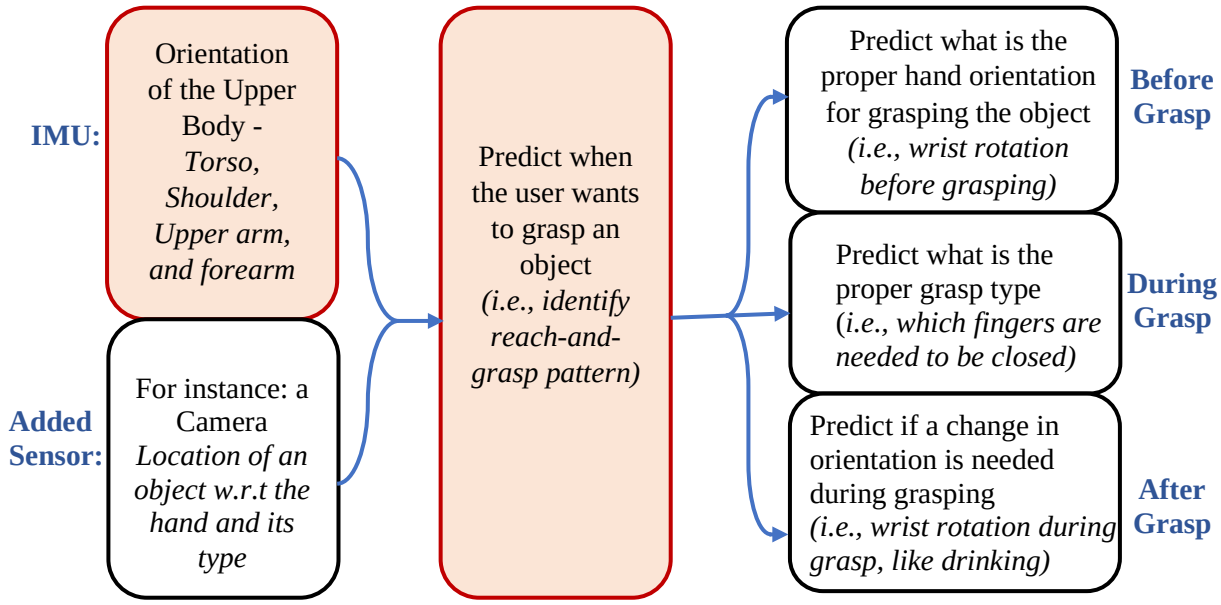


Figure 1: **The big picture.** This research is part of a broader idea and is focused on predicting the grasp moment based on the upper body kinematics (in red).

1.3 Background

1.3.1 Quaternion

There are few options to represent the orientation of an object in space. Here, since our proposed system is based on IMU, the calculation of the 3d orientation is represented by a quaternion. Same as rotation matrix, quaternion represents the orientation of an object in space, but it benefits from more compact and stable representation which results in a better real-time implementation without the need to deal with singularity as occurs when working with Euler angles. Moreover, according to [21] it is the recommended representation to use. Another common representation is the exponential map as was used by Julieta Martinez et. al. [17] to represent each joint.

Quaternion is a 4-elements representation that can be considered as a generalization of the complex numbers with three imaginary axes or a combination between a scalar and a vector. Here we defined that the 1st element is the scalar, and the other three elements are the vector.

The quaternion's four elements can be expressed as the following combination of axis-angle representation [28]:

$$q = \begin{bmatrix} \cos\left(\frac{\theta}{2}\right) \\ \sin\left(\frac{\theta}{2}\right) \cdot \hat{n} \end{bmatrix} \tag{1}$$

So that, θ is the rotation angle and n is the rotation axis.

It has been shown that the optimal estimator for the mean quaternion which minimizes the MSE of the angle between each quaternion and the mean estimator, can be obtained accordingly [29]:

$$\hat{q}_{avg} = \sum_i q \,/\, \left| \sum_i q \right| \tag{2}$$

The transformation for quaternion to the angular velocity can be calculated based on [28]:

$$(\omega_I^B)_I = 2 \cdot \dot{q}_I^B \cdot q_I^{B*} \tag{3}$$

This means that eq. (3) results the angular velocity of system B, the rotated system w.r.t system I, the fixed system, in I coordinates system (CS). According to ISB recommendation[30], a transformation from system I to system B is applied to represent the angular velocity of the distal segment (i.e. system B) w.r.t the proximal segment (i.e. system I) in the distal segment CS, according to eq. (4) :

$$v_B = q_B^I \cdot v_I \cdot q_B^{I*} \tag{4}$$

So that v is a general vector and eq. (4) represents the operation of vector rotation by a quaternion.

All other kinematic derivates such as angular acceleration and jerk can be developed accordingly to obtain the following relations:

$$\begin{cases} \omega = 2\dot{q} \cdot q^* \\ \alpha = (2\ddot{q} - \omega \cdot \dot{q}) \cdot q^* \\ \dot{\alpha} = 2\dddot{q} \cdot q^* - 2\alpha \cdot \dot{q} \cdot q^* - \omega \cdot \ddot{q} \cdot q^* \end{cases} \tag{5}$$

In eq. (5) we eliminated the I, B notations for better visualization. Since all of the above derivatives are calculated as a quaternion, the final results can be obtained by extracting only the vector part from the 4-element result [see Appendix A].

Similarly, for Euler angles and rotation matrix which are the default representation of VICON™'s system, the angular velocity can be calculated by using the following equations [28]:

$$\begin{cases} [(\omega_I^B)_I]_\times = \dot{R}_I^B \cdot (R_I^B)^T \\ [\omega]_\times = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix} \Rightarrow \omega = \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \end{cases} \tag{6}$$

All other kinematics derivatives equations can be seen in Appendix A.

1.3.2 Motion Sensors

The IMUs in our research are Bosch's chip with a fused algorithm to extract the absolute orientation of the sensor w.r.t the world, where the y-axis is facing toward the magnetic north and the z-axis toward the inverse direction of gravity. Notice not all IMUs have magnetometers and outputs the absolute orientation of the IMU w.r.t the world. The advantage of using a magnetometer that defines the 'world' CS, is that it can be assumed that the orientations of all sensors are w.r.t the same 'world' CS, while no magnetic field is present in between the sensors. Hence, we can obtain the orientation of each sensor w.r.t the other at the same CS. We also used here the conventional equations to calculate the angular velocity, acceleration, and jerk from the quaternions as explained in section 1.3.1.

1.3.3 Numerical Differentiation

Numerical differentiation is a rich field of its own. There are many theories, studies, and rules of thumb of which numerical differentiation method is best for each data type. Here we decide to estimate all derivatives according to the following equation for higher derivatives based on Newton's difference quotient:

$$f^{(n)}(x) = \frac{1}{\mathrm{h}^n} \sum_{k=0}^{n} (-1)^{k+n} \binom{n}{k} f(x + k\mathrm{h}) \tag{7}$$

So that n is the derivative order and k is the window width.

1.3.4 Motion Capture

Optoelectronic Stereophotogrammetry is a method for non-invasive motion measurements. One of the goals in optoelectronic motion measurements is to estimate a rigid motion of a specific body part. For this purpose, markers are attached to the body part surface. The kinematics of the arm and hand was measured using a VICON™ motion capture system (VICON™, Oxford Metrics Ltd., Oxford, UK). The system in our lab is comprised of 20 cameras collecting frames at 120 Hz. The system was also used to sample analog data for synchronization between the motion capture system and the IMU-based system we developed. The analog data were sampled at 960 Hz using the internal analog to digital converter.

1.3.5 Machine Learning

Human motion is to some degree random and characterized as a high dimensionality problem. Even though many of our movements are pattern-based there is significant variance in human motion due to different decision making and the inherent large degrees of freedom (DOF) of the human motor system. In this research, we are using some of the common classifiers for biomechanical classification problems and in general - KNN, RF, and SVM, and some of the common RNN such as LSTM and GRU [17], [31]. It is common to replace the LSTM architecture with the GRU to reduce training time, presumably the results from the new architecture are similar [17].

To validate between our system and VICON™'s we are using the Pearson coefficient. Pearson coefficient represents the linear correlation between two values and is defined as:

$$\rho_{X,Y} = \frac{cov(X,Y)}{\sigma_X \cdot \sigma_Y} \tag{8}$$

So that $\rho_{X,Y}$ is the Pearson coefficient between signal X and Y, $cov(X,Y)$ is the covariance of the two signals, and σ is the standard deviation of each signal.

Chapter 2

The *R2Gmotion* System

2.1 System Overview

Here presented a wireless, wearable system, we developed with a focus on easy-to-use and easy-to-modify, including a coherent GUI for the basic developer and common users with no strong background on the system hardware and software. We named it the *R2Gmotion* system, inspired by Reach-to-Grasp Movement.

The system we developed can be separated into two different versions. One version is for the experiment itself where we added another IMU sensor on the wrist – a total of 5 IMU sensors, and a logging procedure to save the raw data on the controller. Any other data analysis was unnecessary since all the analysis was done off-line and not in real-time. In contrast, the second version is the actual *R2Gmotion* system which consists of only 4 IMU sensors, real-time data analysis without the need to save the data on the controller memory, communication with Python application for the machine learning algorithms which in turn sends commands via Bluetooth Low Energy (BLE) to the hand mobile application we developed as well as the low-cost 3d-printed robotic hand that we used to demonstrate the system abilities.

In general, the system components are 4 or 5 low-cost motion sensors, a controller, a GUI application (i.e., MAIN app) for the experiment and the demo, a python desktop application (i.e., ML app), the Hand mobile application, the Hand itself (see Figure 2). All system components were developed during this research except for the motion sensor and controller, but including all software and the Hand itself.

In the future version, the controller, which is myRIO, National Instrument, should be replaced with a basic, low-cost controller. We chose this controller in favor of the experiment and demo which needed a GUI. All the machine learning algorithms are implemented on Python, a free, open-source platform.

One of the main purposes of this research was to develop a practical, accessible, and reproducible idea to achieve a result that is as close as possible to real-life application. Thus, the described requirements were necessary and inherent in this research and system architecture.

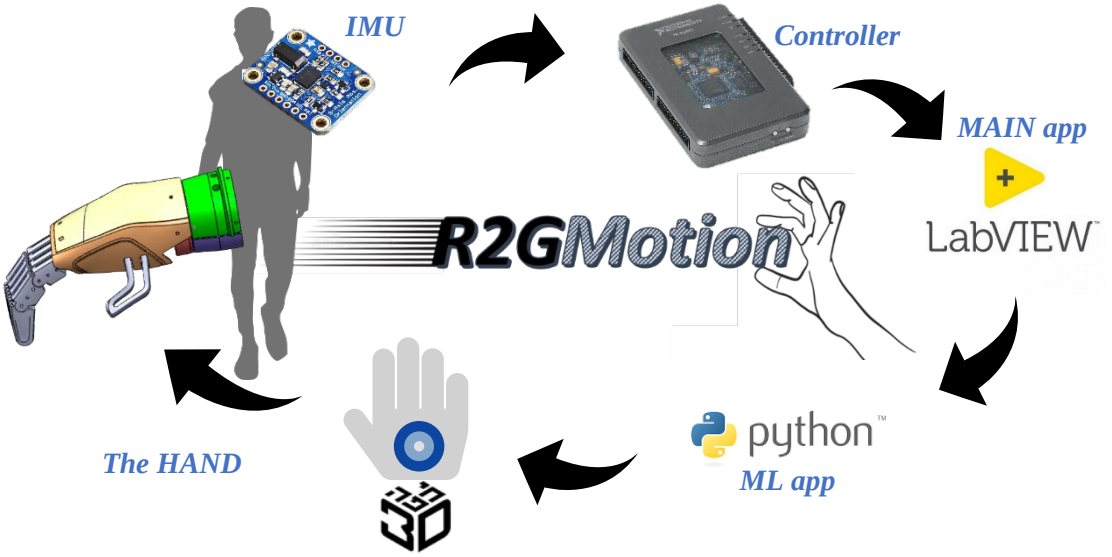


Figure 2. ***R2Gmotion.*** Main components overview, including IMU sensors, myRIO controller, MAIN app developed on LabVIEW, ML app developed on Python, and the HAND we developed in collaboration with Haifa3D

A link for a video demonstration of the *R2Gmotio* system can be found in Appendix B. The demonstration includes the *R2Gmotion* wearable and wireless system when a user reaches her hand to grab an object using the prosthetic hand and the hand's fingers are closing autonomously when a grasp moment was detected.

2.1.1 Motion sensors

BNO055 9-DOF sensor of Bosch's embedded on Adafruit breakout board is a motion sensor including accelerometer, gyroscope, and magnetometer which also has a processor that can compute and processed a fuse data such as the absolute orientation of the sensor relative to earth [see Figure 3]. There are several optional configurations of this sensor and for our system, we selected the following: external crystal oscillator: ON, chip placement: P0 and operation mode: NDOF, (for more details, see Bosch's datasheet [32]). After turning on the sensor, there is a need to obtain offset parameters that calibrate the sensor w.r.t the world (z-axis is pointing opposite to the gravity direction, and y-axis toward the magnetic north). When the system environment changed significantly, and in our case, before any experiment trial, we calibrated the sensor manually by keeping the sensor still on a flat surface parallel to the ground (for the gyroscope), moving the sensor in space as a figure-8 shape (for the magnetometer) and finally placing the sensor in 6 different orientations by slowing transform from one orientation to the other (for the accelerometer). If the system environment is similar to the last time the sensor was activated, or if the accuracy is not very important, we uploaded the previous

calibration parameters, i.e., the offset parameter to the sensor registers. The sensor has two optional addresses for I2C communication. Since we used more than two sensors we added an I2C multiplexer component (TCA9548A) to toggle between the sensors.

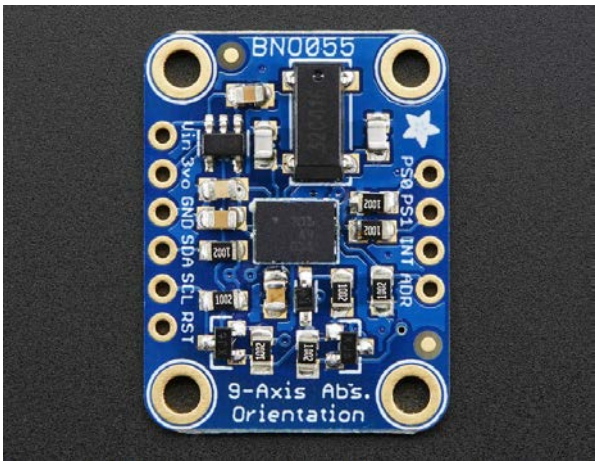


Figure 3. **IMU sensor.** Adafruit breakout board for Bosch's BNO055, 9-DOF motion sensor

2.1.2 Controller

myRIO is National Instrument's (NI) controller which samples the data from the motion sensor via an I2C communication and is also used to synchronize between the motion sensors data and the VICON™'s data. In addition, the instructions for the participants are based on a GUI implemented in LabView, National Instrument's Software and the communication between the computer and myRIO is via Wi-Fi. The myRIO was powered by two Li-ion batteries and contains a memory card where all the experimental data was saved [see Figure 4]. This evaluation board offers relatively simple and easy-to-use real-time features such as FPGA and clock configurations. The constant sample rate of this device and its real-time capabilities are very important for many data analysis applications. In our application, the controller does not sample the data but read it via I2C communication which is limited by 100 kilobits per second in the standard mode in addition to the sensors' embedded sample rate of 100 [Hz], which means the real-time features of myRIO is not essential for our purposes. Moreover, biomechanical movements are characterized by a low-frequency rate, thus a low-rate data acquisition can be suitable for our application as well. Due to the low-frequency rate, the synchronization process between our system and VICON™ does not have to be based on clock sync, but only based on a cue that triggers both systems. Here we chose a push-button connected to myRIO I/O to be considered as the trigger.

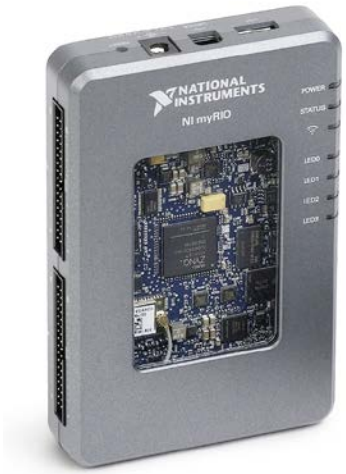


Figure 4. **Controller**. myRio is National Instrument's controller

2.1.3 MAIN App – LabVIEW GUI Application

The MAIN app of the *R2Gmotion* system was developed on LabVIEW, NI with the dependency installation of NI's myRIO bundle and NI's Robotics toolkit. This platform allows compiling code into myRIO controller for wireless application and to design a GUI. The GUI is important mostly for the experiment since the grasping task was managed by the MAIN app. The app is designed to randomly arrange 8 objects in different locations and randomly set the timing (between 1-5 seconds) of vocal and visual instructions. During the experiment, the MAIN app also logs the raw data from the sensors.

The design of the MAIN app is very modular and based on a state-machine model. Thus, both the experiment app version and the real-time app version are based on the same app structure with few different modules (i.e., functions). Furthermore, although this app was designed to acquire and analyze the right upper body movement, any kinematic chain such as left arm, right leg, fingers, and even a serial robot can be used by only changing the Setting in the app GUI with no need to change the code itself.

The Setting module we developed is divided into five different types which help the user to configure many parameters, including data logging, the number of sensors, sensors calibration and registers, communication parameter (I2C and Wi-Fi), additional connection to the controller, data analysis parameters, the 'Grasp Task' objects and timing for the experiment, the Phantom (3D model) design and parameters, which mimics the user's action for the real-time demo and more.

As expected in LabVIEW programming and most parallel computing systems, the app structure is a state-machine with several modules that are managed by the event holder and the command holder [see Figure 5]. The data analysis modules (green blocks in Figure 5) convert the acquired data from

the sensors to comprehensible values and apply the mathematical algorithms described in section 4. During the data acquisition (DAQ) the compiled code on the myRIO controller is based on a 'Real-Time loop' of 40 Hz. The sample rate was defined for robust DAQ from all 5 sensors including saving the data in parallel to acquiring it on the controller itself. The communication between the desktop app and the compiled myRIO app was via Wi-Fi and the DAQ module was responsible to transfer indications from one app to another.

Each module is controlled by the command holder in a queued manner when the processed data flow from one module to another as a queue (save all data-points in a First-In-First-Out mode) or as a notification (save only the last data-point). For the 'Experiment version', the data were analyzed offline, hence it was important (and possible) to save each sample and the data flow for the 'Log' module was a queue. However, for the 'Real-Time version', the data is analyzed in real-time and a lag in the data is not excepted, hence the data flow was based on notification and a data loss, i.e. skipped samples, could have happened. Another difference between the two versions is the main display – for the experiment version, the Grasp Task was projected [see Figure 6 and Figure 9 for ongoing 'Grasp Task'], and for the real-time version a phantom, 3D model, of the estimated joints angles is projected for visual validation [see Figure 7]. Also, for the Real-Time version a 'Predict' model was implemented which is responsible for the communication with the ML app written in Python.

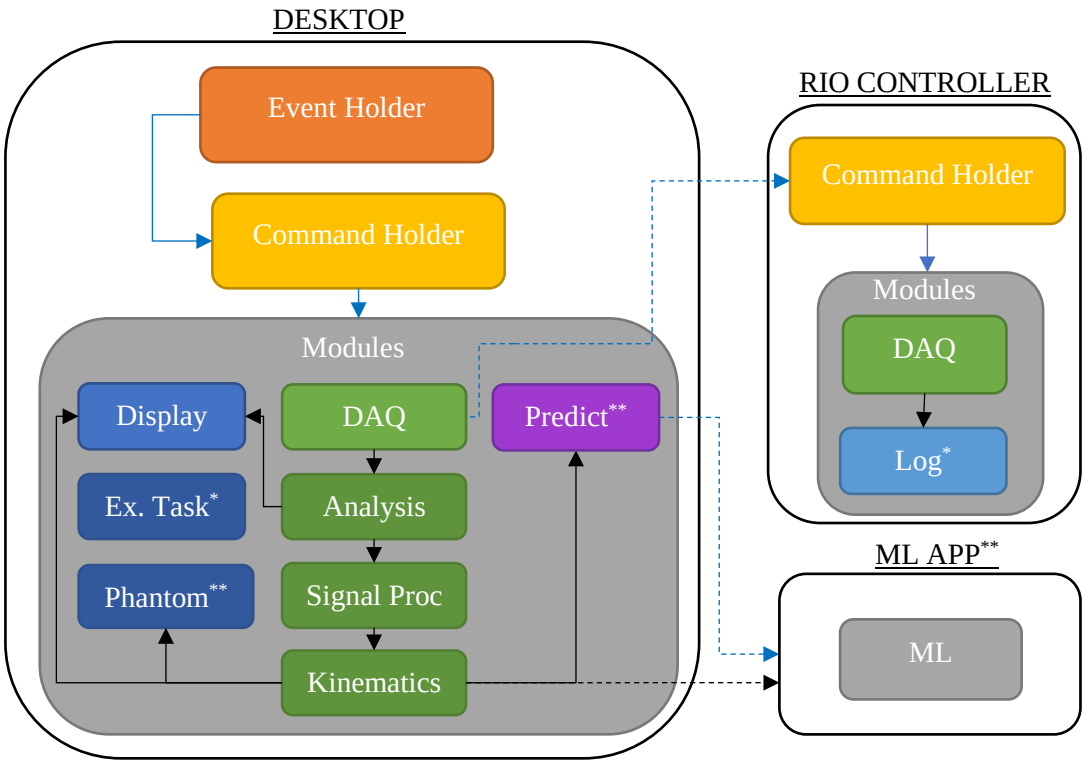


Figure 5. **MAIN app.** Block diagram of the *R2Gmotion* system MAIN app. The command communication is denoted by blue arrows and data streaming by black arrows. The broken line indicates wireless communication. Modules with one asterisk are for the experiment version only, while modules with two asterisks are for the real-time version only.

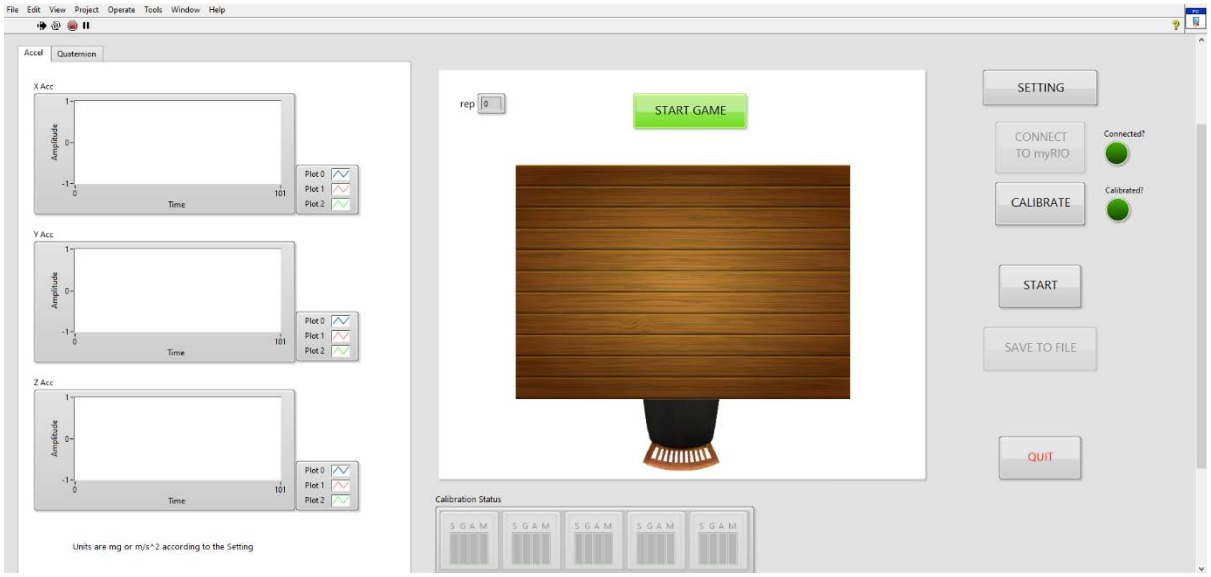


Figure 6. **Experiment GUI.** The main GUI for the Experiment version of the MAIN App we designed. On the right panel - the user can run different operations, on the left - raw data graphs are displayed, and on the middle - the experiment's Grasp Task is administered by the app which produces a vocal and visual queue for grasping a random object, at a random place on a random time.

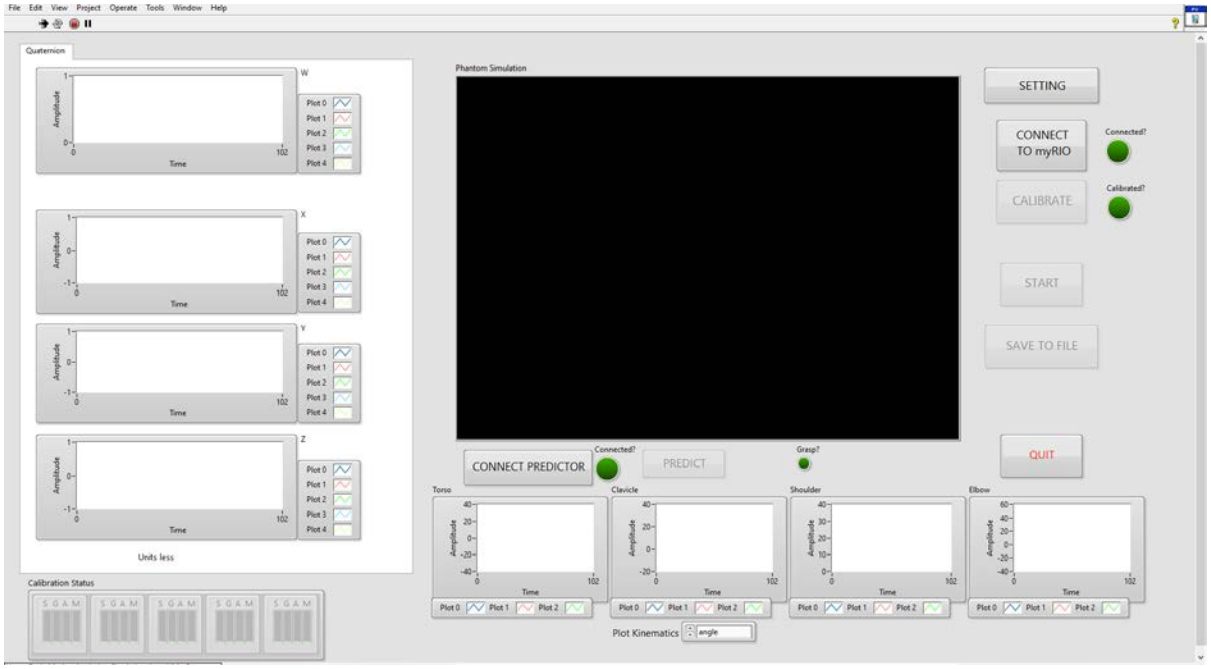


Figure 7. **Real-Time GUI.** The main GUI for the Real-Time Python-connected *R2Gmotion* system. On the right panel - the user can run different operations, on the left - raw data graphs are displayed, on the bottom - the estimated joints kinematics is shown, and on the middle - the 3D phantom model of the user is plotted.

2.1.4 ML App – Machine Learning Platform

As mentioned in the previous section, the data analysis is done on NI's platform, LabVIEW, (i.e., the 'MAIN app') so that the MAIN app outputs the kinematic data of the investigated joints - the joint's angles, angular velocity, angular acceleration, and jerk. Based on the kinematic data, learning algorithms were developed to predict the R2G stage and more specifically the grasp moment. This part was developed in Python due to the ready-to-use libraries of machine learning, including tools for fast learning by using the computer resources more efficiently, and since it is open-source which is important to this research goals. Using TensorFlow capabilities in Python we trained many different net architectures on offline data until we selected one specific architecture [see Section 6 for more details]. Then we developed a simple python application for real-time prediction and embedded it into the *R2Gmotion* system (i.e., the 'ML app'). See Appendix C for the full code including the trained model.

The implemented communication between the MAIN app and the ML app is Transmission Control Protocol, Internet Protocol suit (TCP/IP). We created two different communication sockets – one for commands and the other for data streaming. The data streaming is one-directional communication for sending estimated kinematic data by the MAIN app to the ML app. The ML app was developed as a state-machine model, consequently, the MAIN app manages the state of the ML app by utilizing the commands sockets such as when to start listening to the data stream, when to stop, quit and so on. In contrast to the data-streaming socket, the commands socket was designated for bidirectional communication, so when an event occurs (e.g., a grasp moment was detected), the ML app can alert the MAIN app. In addition, the ML app established a BLE communication (client) with the Hand controller (master) that we have developed, according to the BLE server address. If a grasp moment was detected the ML app sends a commend to the Hand as well using the BLE protocol, we developed [Appendix C- link for our open-source app].

When the MAIN app sends a commend to start processing the incoming data for R2G prediction, the ML app reads each sample according to predefined order, then normalizes each incoming sample by applying the selected normalization method with the calculated parameters based on the train session and finally applies the learning algorithm architecture that was carefully chosen [see Section 6 for more details].

It is important to make sure the ML app parameters are according to the MAIN app, such as the number of bytes per datapoint (8 bytes), the window size (10 samples which are 250 milliseconds for our system). The threshold parameters as described in Figure 16 can be modified to fit the relevant need as investigated in Figure 24.

2.2 The Hand Project

The Hand Project aims to provide a low-cost prosthetic solution to overcome the limitations of the current commercial prosthetic hands. The main objective of our model is to have a prosthetic hand design for people with transradial amputation that their needs are yet to be met by the current prosthetic solution on the market. According to feedbacks we collected from prospective users and according to Cordella et al. paper[3], the main design considerations include (1) electronic control that does not require a muscular activity of the residual limb; (2) lightweight; (3) reliable and durable; (4) intuitive and easy to use; (5) rotatable; (6) variable grip force pattern; (7) low-cost.

The solution presented here is an electronic hand that can perform a variety of user-defined hand configurations and grips by actively control flexion and extension of four fingers and rotation of the wrist with two different levels of grip torques. The mechanical design consists of four fingers with four different tendon-driven mechanisms actuated by four different micro-DC motors, a passive thumb with two discrete positions using two magnets, and wrist rotation driven by gears and a DC motor. All five independent micro-DC motors are driven by a microcontroller system that measures the motors' current. The adaptive control mechanism allows autonomous grip adaptation to different objects' sizes and shapes, with different torque levels [Figure 8]. The user has full control over the hand configurations using a mobile application we developed. The mobile application and more specifically the BLE protocol we developed allows fast and simple integration with different sensors and technology that can be used to control a prosthetic hand.

The design is focused on low-cost, easy to reproduce and assemble for any person around the world with only basic technical capabilities and is published as an open-source project including the mechanical design, electronic and software [Appendix C].

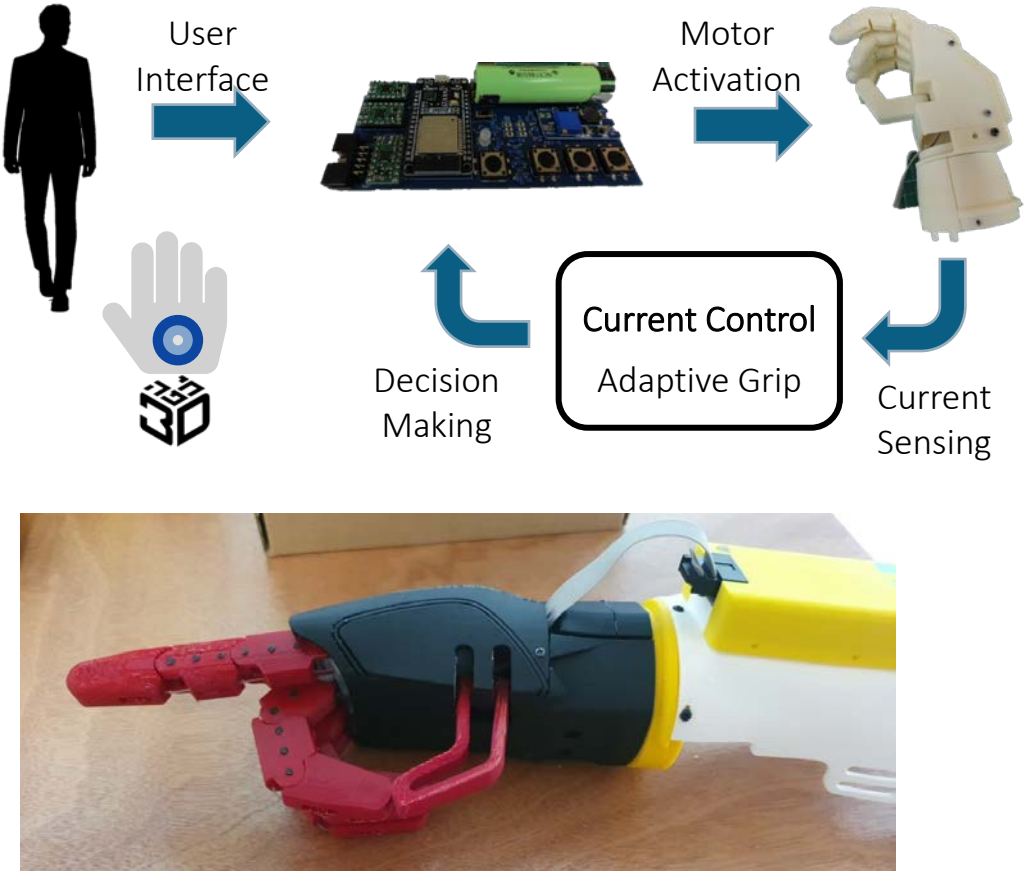


Figure 8. **The Hand Project.** System functionality overview (top) and the printed hand (bottom)

Chapter 3

Experiment

|The purpose of the experiment was to gather an enriched dataset of R2G movements including the starting and ending point of the movement. 10 healthy right-handed subjects participated in the experiment during two different days, to reduce the sensors placement effect. Each trial was started by preparing the subject according to VICON^{TM} motion capture system's Right Upper Limb (RUL) model ─14 markers at specific landmarks on the right upper body segments; and according to the proposed system *R2Gmotion* which consists of 5 IMU sensors on the torso, shoulder, upper arm, below the elbow and palm. We took the measurements of each subject's upper body segments [see Table 1] for normalization purposes and then the participants were asked to complete the calibration routine which involves the following poses: standing straight, lean forward and standing in the T-Pose. According to these three poses, our system calculates the sensor-to-body-segment calibration as further discussed in the next sections. Then, during each trial, the participants were asked to accomplish the Grasping Task, then walk freely around the motion capture room (3m x 8m approx.), carry out the Grasping Task again and at the end, had a conversation about the trial itself. The 'walking' and 'talking' parts were measured and considered as differentiative movements (i.e., not R2G movements) for the R2G classification. The participants were not aware of this purpose and were told the 'walking' part is to rest between the two grasping tasks and the conversation at the end was for improving the experiment protocol [see Figure 11].

3.1 Experiment Objective

To learn more about the R2G movement with the focus of predicting the R2G stage, where 0% is the Reaching moment and 100% is the Grasp moment, we needed a dataset including the time stamps indication of the starting and ending point of the R2G movement. The purpose of the *R2Gmotion* system was to be able to estimate the R2G stage and predict the grasp moment for ADLs, thus the dataset we collected represents a variety of R2G movements and more specifically (i) different type of objects, (ii) grasps angles, and (iii) different initial states before the R2G movement (e.g., rest, during series of movements, etc.). Since the system we developed was yet to be validated the experiment protocol includes synchronization with a state-of-the-art motion capture system to evaluate the

estimated anatomical joints of our system. Moreover, an added-value of the *R2Gmotion* system is the easy-to-use and simple-to-wear system thanks to the calibration process we developed which reduces drastically the dependency of the sensors placements on the users. Thus, each participant was required to accomplish the experiment twice, on two different days. Lastly, the dataset also includes other movements that are not R2G movements to demonstrate the system's ability to differentiate between R2G and non-R2G movements. Here we decided to use the natural hand movement during walking and communication such as a conversation between the participant and the experiment conductor.

3.2 Experiment Setup

The research population includes 10 subjects according to inclusion criteria of right-handed, 18-40 years old subjects while excluding any subjects with major hand orthopedic and neurologic conditions. The subjects volunteered to participate in the experiment and sign a consent form after the experimental procedure was thoroughly explained to them. Each subject participated in the experiment on two different days, and the experiment of each day took less than an hour to complete.

The experiment was conducted in the Biorobotics and Biomechanics Lab (BRML) at the faculty of mechanical engineering, Technion, Haifa, Israel. The experiment was at a closed room (3m x 8m approx.) equipped with 24 VICON™ cameras and the auxiliary required hardware for the VICON™ system and our *R2Gmotion* system (see full details on *R2Gmotion* hardware in section 2.1). One of the supplementary hardware that worth mentioned is the ZeroWire™, third-party sensors that enable the VICON™ system to acquire analog signals so that the signals are integrated and synchronized with the markers' position and joints angle estimation data. To such a sensor we connected a button that was also connected to the input of the *R2Gmotion* controller and served as a trigger signal for synchronization of the two motion capture systems. Both systems were calibrated according to the manufacturer’s instruction, the optoelectronic VICON™ system and the BNO055 Bosch’s IMU sensors.

The main task during the experiment was a pick-and-place task of different objects (i.e., the ‘Grasping Task’). The objects’ positions in space were acquired by placing one marker per object. All objects are meant to be grasped and moved by (i) only the right hand - tennis ball, coin, mug, cup, mouse (for pc), and pen, (ii) by using both hands (big square box). One of the objects (fabric strip) was only for the subject to touch. The objects were selected to represent different types of daily-living objects [Figure 9].

All objects were put on a table with a fixed height. Each subject seat on an adjustable office chair next to the table, when the chair height was changed in a way there was a little more than 90 degrees angle between the subject’s upper-arm and forearm while seating straight and until it considered as a comfortable sitting posture according to the subject. Next, the subjects held chalk and while seating

with a straight back and arm they draw a half-circle around them (the table surface was a blackboard). The marked half-circle indicates the objects' possible distance from the subject (i.e., the effective radius) and the experiment GUI application we developed (see Figure 9c) defined the specific location of each object along the half-circle (i.e., the effective angle) from the following options – Left-Down (LD), Left-Middle (LM), Left-Up (LU), Middle-Left-Up (MLU), Middle-Right-Up (MRU), Right - Up (RU), Right -Middle (RM), and Right -Down (RD) (see Figure 9b).

During the 'Grasping Task', on the first pick-and-place trials, the participants arranged the objects outside the half-circle (far from them, approximately at 70% ROM) and their task was to place the objects inside the circle (close to them), on the last pick-and-place trials, the objects were arranged inside the half-circle (close to them, approximately at 50% ROM) and the participants picked each object and placed it outside the circuit. The described method was done to normalize the planetary location of the objects w.r.t the arm length of each subject to obtain two different reaching distances of 50% and 70% w.r.t the participant's total arm length. The following subject's measurements were taken to evaluate the objects' distance normalization – height, pelvis height, hand size – from pinky to thumb when the hand is completely open, and from middle finger to wrist, the forearm length, as well as the upper-arm length (see Table 1 for all subjects' measurements).

Table 1. The subjects' measurements of all relevant body segments in [cm]

Subject ID	**Height**	**Pelvis height**	**Pinky to thumb**	**Middle finger to wrist**	**Wrist to elbow**	**Elbow to shoulder**
1	158.75	95	19.55	18.4	24.85	31.15
2	161	88.5	17.95	18.2	23.4	30.5
3	187	100	19.95	21.55	25.75	37.9
4	183	96.75	21.05	20.7	23.45	35.75
5	180	92	22.85	20.4	27.5	36
6	164.5	97	19.1	19.55	25	34.5
7	162	91	18.85	19	23.65	35.5
8	165	97.5	21	18.5	25.15	34.75
9	171	94.5	23.15	20	27	34.75
10	156	88.5	18	17.75	23.5	32.5

All measurements were taken twice and averaged to reduce human-error

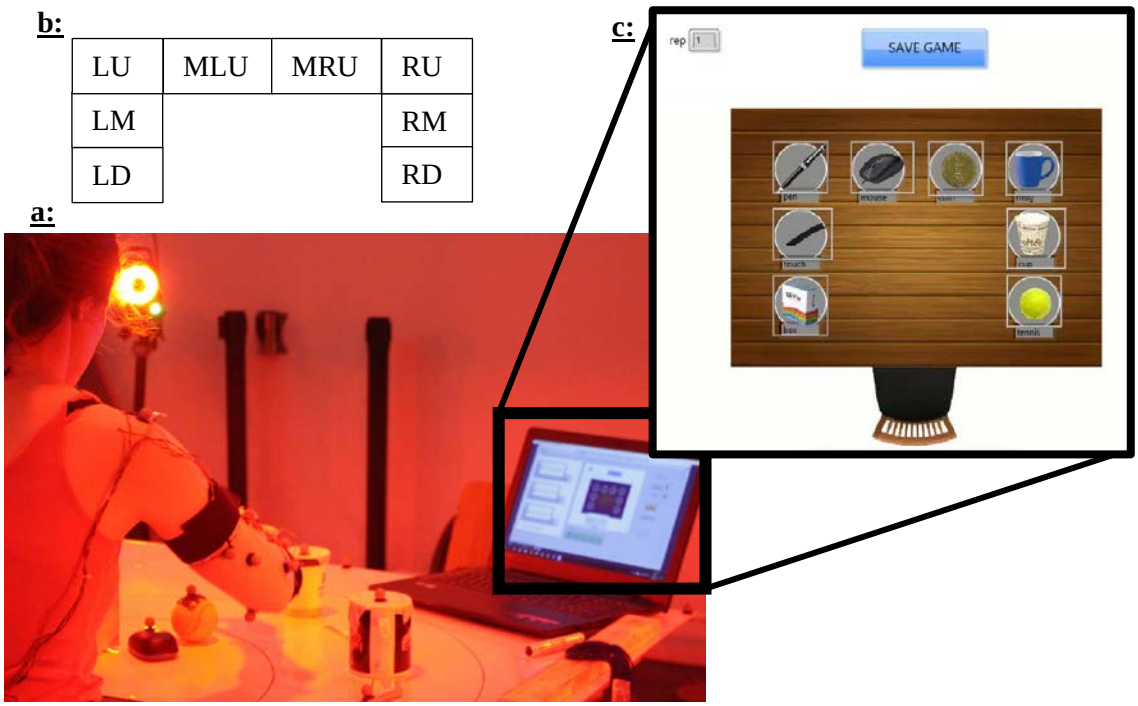


Figure 9. **Experiment's Grasping Task**. Example of the Grasping Task setup (a). Before each task, the objects' initial location was selected randomly from all optional locations (b) and during the task, the GUI (c) instructed the subjects on when and which object to grasp.

3.3 Experiment Protocol

The full experiment protocol is summarized in Figure 11 block diagram and described here in detail.

Step 0 – Subject Preparation: First, 14 markers were placed on the subject according to the predefined RUL model of Nexus™, VICON™ software[33] (see Figure 10). The markers were placed using hypoallergenic two-sided tape. Then, the *R2Gmotion* suit was placed on the subject including 5 IMU sensors – torso, right shoulder, right arm, right forearm, and right palm, using hypoallergenic two-sided tape and rubber straps (see Figure 12)

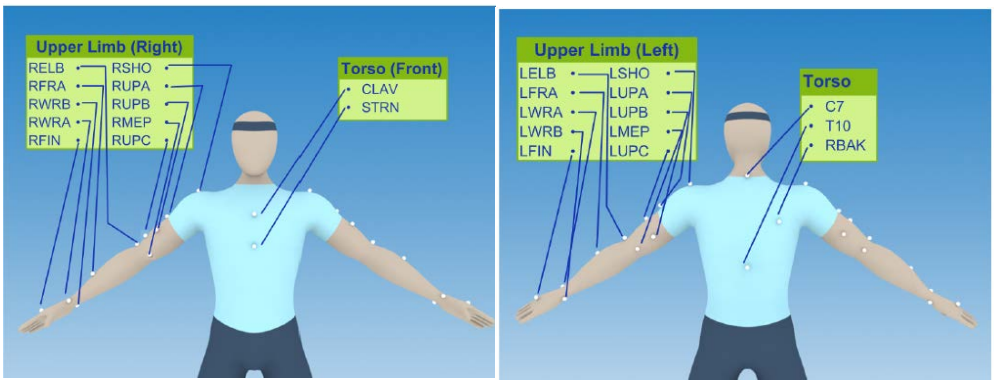


Figure 10. **VICON™'s RUL model**. Upper Limb Model marker placement from [33], front (right), and rear (left) view

Step 1 – Initial Measurements: Each subject was measured, the chair height with respect to the table was adjusted, and the normalized objects' positions on the table were marked (see section 3.2 for further details).

Step 2 – Sensor-to-Body-Segment Calibration: At the beginning of each trial the subjects were instructed to follow the next steps: clicking on the sync button and tapping three times with a flat hand on the table (an additional process for synching the two motion capture system), standing straight for 5 seconds, leaning forward for 5 seconds, then stand straight again for 5 seconds, and finally standing in a T-pose after bending the elbow few times parallel the ground to ensure the elbow flexion axis is nearly parallel to the gravity direction (see appendix B for a link to the sensor-to-body-segment calibration process video).

Step 3 – Main Experiment: The core of the experiment started with a 'Grasping Task' while sitting in front of a table. The subjects arranged the objects on the table according to the GUI instructions (see Figure 9). Then according to visual and vocal cues, the subjects were instructed which object is needed to be grasped using their right hand. After the objects were picked-and-placed, the above two steps were repeated for 12 repetitions – 6 repetitions when the distance of the object is approx. 70% of the maximal ROM and 6 repetitions when the distance of the object is approx. 50% of the maximal ROM (see section 3.2 for more details about the object distance from the participants' initial position). When all 12 repetitions were completed, the subjects were asked to walk around the room at a random pace and random directions for 3 min (i.e., the 'Walking Task'). Then, another 'Grasping Task' was carried out but this time the subjects were standing next to the table, instead of seating, while performing 12 repetitions of the 8-objects-pick-and-place as before. The experiment was concluded with elaborating conversation about the experiment, the participants' thoughts, and opinions about the protocol (i.e., the 'Talking Task'). The participants were not informed about the main purpose of the 'Walking Task' and 'Talking Task', which was to acquire hand movements that are not related to grasping and to R2G movement, to prevent un-nature hand movements.

As mentioned, the above protocol was repeated one more time on a different day.

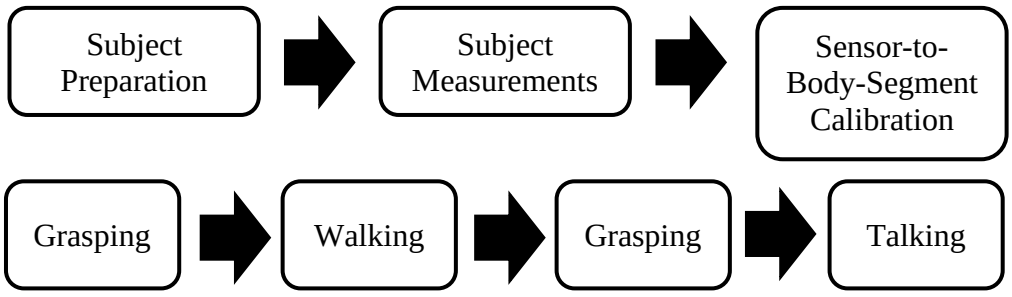


Figure 11. **Block diagram of the experiment protocol**

Chapter 4

Estimation of the Joints Biomechanical Angles

Based on the data acquired during the experiments, the biomechanical joints angles were estimated. As mentioned, the experiment includes both VICON™'s system and our *R2Gmotion* system, hence different methods were applied per each system.

The raw data from VICON™'s system was the positions of the markers in space – 14 markers on the subject according to the RUL model, 8 markers on each object, and 3 markers on the experiment table. By running the RUL predefined model, on Nexus software, the following joint angles estimation were produced – right-shoulder, right-elbow, and right-wrist. The VICON™'s system sample rate of the marker position and in turn the joints angles estimation was defined to be 120 [Hz].

The raw data from the *R2Gmotion* system was the IMU data including time stamps, calibration status of each sensor, acceleration, and quaternion (the absolute orientation of each sensor). Moreover, the details of each trial during the 'Grasping Task' including the instructions time when to pick up an object, and the objects' time and location were logged as well. The IMU data from all trials were sampled by the controller at 40 Hz, then cleaned and filtered. The sensor-to-body-segment calibration was calculated according to the calibration session each participant has preform, and the joints' relative angles were calculated [Figure 12].

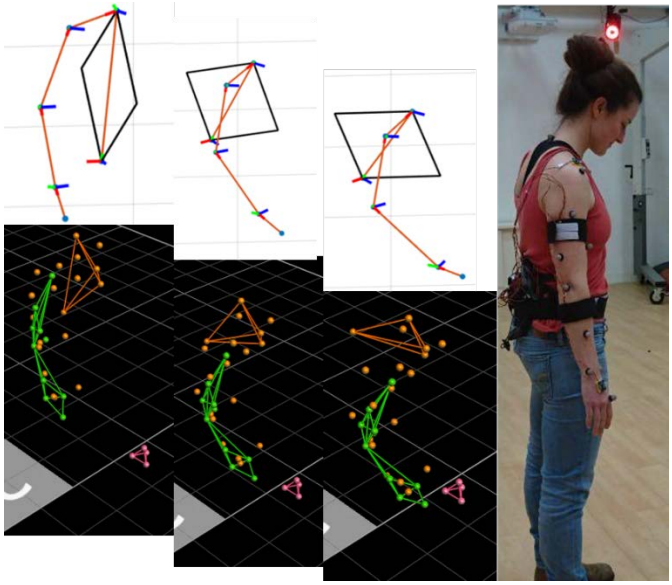


Figure 12. **Experiment's outputs example.** According to both modalities, IMU-based and marker-based systems, the joints angles are calculated, and a reconstruction of the subject's body-segments is estimated including the torso, shoulder, arm, forearm and palm (markers-based bottom, IMU-based top).

4.1 Pre-Processing

4.1.1 The unit quaternion

One of the advantages of using quaternion representation of rotation in space is the continuous outputs, in contrast to other representations such as Euler angles. As a result, it is much easier to apply several mathematical operations such as numeric differentiation. Nevertheless, since a unit quaternion of an opposite sign represents the same rotation, meaning $q = -q$, it might cause a seemingly inconsistence. Hence as a first step, we check for a significant change between two successive samples. By assuming biomechanical actions are slow and characterized by the low-frequency rate (<15 Hz), and since each quaternion's element is bounded between ± 1, we defined that any difference above 0.4 between one of the elements of two successive quaternions is defined as a 'jump' in the data.

Anytime a 'jump' was detected, we multiplied by (-1) the next quaternion samples and searching again for the next first 'jump' in the data. We repeated this process until all data-samples were examined. Notice that $q = -q$, hence a multiplication by (-1) does not modify the raw data in any way. If the change in the quaternion sign did not improve the smoothness of the data, it means that some other disturbance occurred during the acquired trial.

4.1.2 Missing data-points

As part of the *R2Gmotoin* system, we used the myRIO controller which allows a real-time sampling operation. The sampling rate was defined to be 40 [Hz] and was compiled according to LabView's

'timed-loop' feature. If for any reason a sample was not acquired, the controller skips the current sample and sample again after the defined period, meaning a constant sample rate can be assumed w.r.t the low-frequency rate of the biomechanics motion. To identify the missing data-points, we checked if the timestamps of the IMU measurements has an uneven sample rate. Since in our case all detected gaps were considered small gaps, smaller than 250 milliseconds (10 samples), we replaced the missing data-points with the average data of the two gap's ends. In total, less than 100 samples from the whole dataset were missing and needed to be treated accordingly.

4.1.3 Filtering

Since human-movements are characterized by relatively slow motion, a low-pass-filter was applied to the data. Butterworth 4th order with 15[Hz] cut-off frequency was selected and filtered the data by using zero-phasing digital filtering in MATLAB.

4.2 Sensor-to-Body-Segment Calibration

The sensor-to-body-segment calibration calculation is based on a simple and quick process that does not depend on the sensor placement and requires only two main movements: standing straight to leaning forward and the T-pose (see Figure 13).

4.2.1 Joints Coordinates System Definition

First, let us define the joints coordinates system (CS) of *R2Gmotion* – the central coordinates system is defined as z-axis along the negative direction of gravity (from leg to head during standing), x-axis pointing toward the right side of the body, and the y-axis pointing from the user's back to front. Then the other serial CSs are defined as described in Figure 13. Since we assume the user can travel around and grasp objects from different angles w.r.t the initial calibration position, we used only the thorax flexion angle. All CSs in this research are right-handed, hence the y-axis of the thorax for instance is pointing from the back to the chest. According to the International Society of Biomechanics (ISB) recommendation [30], the joints angles are defined as intrinsic rotation (i.e., 'frame rotation') of the distal segment w.r.t the proximal segment corresponding to the following order: first the flexion-extension, then the abduction-adduction and finally the axis rotation. The initial position we defined for our *R2Gmotion* system is the T-pose as described in Figure 13, where in this position all CSs are parallel, and all joints' angles are considered zeros. Although the definition of the *R2Gmotion* CS is different in comparison to the ISB recommendation [30] the order of the anatomical angles is the same except for the clavicle joint. The joints' CS and the anatomical meaning of each angle are

described in Table 2. More details about the different definition of VICON™'s system and *R2Gmotion* system is explained in section 5.2.

Table 2. The joints coordinate system definition of the *R2Gmotion* system

Joints	Intrinsic Angles Order	Anatomical Angles Description
Thorax	$xy'z''$	e_1: flexion (-) and extension (+) e_2: lateral flexion rotation: right (+) and left (-) e_3: axial rotation to the left (+) and right (-)
Clavicle	$xy'z''$	e_1: axial rotation: backward (+) and forwards (-) e_2: elevation (-) and depression (+) e_3: retraction (-) and protraction (+)
Shoulder	$zy'x''$	e_1: flexion (+) and extension (-) e_2: abduction (-) and adduction (+) e_3: axial rotation: internal (-) and external (+)
Elbow	$zy'x''$	e_1: flexion (+) and hyperextension (-) e_2: the carrying angle* e_3: axial rotation**: pronation (-) and supination (+)
Wrist	$yz'x''$	e_1: flexion (+) and extension (-) e_2: radial (+) and ulnar (-) deviation e_3: axial rotation: pronation (-) and supination (+)

** The carrying angle is a passive response of the elbow to the flexion-extension and occurs due to the ulna angulation and the humeral axis tilt.*
*** Since we place the IMU sensor a few centimeters below the elbow to mimic transradial amputation, the axial rotation is negligible*

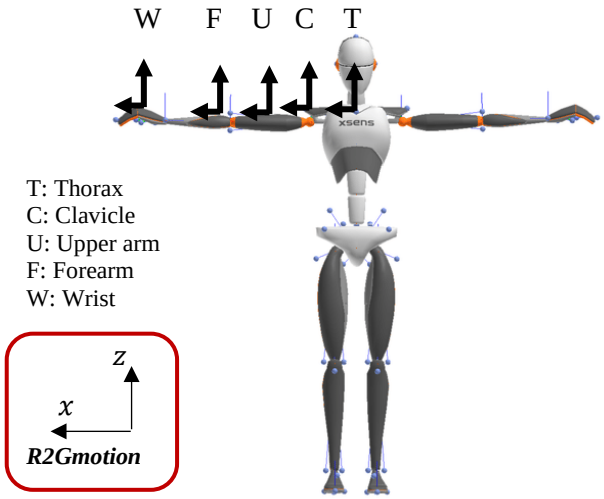


Figure 13. **Sensor-to-Body-Segment**. The defined initial position of *R2Gmotion* where all angles are zero and all joints' coordinates systems are aligned. The T-pose initial position was inspired by XSENS™ and the illustration was taken from their whitepaper [34], while the *R2Gmotion* system definition was added.

4.2.2 Calculating Sensor-to-Body-Segment Orientation

Each IMU sensor acquires its orientation w.r.t the world, denoted as q_w^s. The sensor's world CS is defined as the z-axis toward the negative direction of gravity and the y-axis toward the magnetic north (right-handed system).

According to the leaning forward and standing straight movements, we can extract the orientation of our pre-defined anatomical coordination (z-axis from legs to head, x-axis from left to right and y-axis from back to front). It can be done by averaging the acquired quaternions from the IMU attached to the thorax during leaning forward and during standing straight (see eq. (2) of quaternion averaging) and calculate the relative quaternion of the thorax orientation during standing w.r.t leaning forward:

$$q_{lean}^{stand} = q_w^{stand} \cdot \left(q_w^{lean}\right)^* \tag{9}$$

When q_w^{stand} is the thorax sensor average outputs during standing and q_w^{lean} is the thorax sensor average outputs during leaning forward. According to our definition, the change in the thorax orientation from leaning to standing is a rotation along the positive direction of the x-axis. Hence if the sensor's world CS definition and our *R2Gmotion* definition were the same the normalized q_{lean}^{stand} vector part was supposed to be the x-axis, $[1,0,0]$. Since naturally there is a difference (unless the user is faced exactly toward the magnetic north during the calibration process), we need to find the transformation between the sensor's world and our definition. By assuming the z-axis of both definitions is aligned (the user is standing straight, and the gravity vector is perpendicular to the floor), the transformation between the two systems is a rotation along the z-axis by an unknown angle. To calculate this angle, the basic projection theory can be utilized:

$$\begin{cases} \theta_z = \cos^{-1}\left(v_{ref} \cdot v_{cal,xy}\right) \\ n_z = v_{ref} \times v_{cal,xy} = \sim[0,0,\pm 1] \end{cases} \rightarrow q_w^{R2Gmotion}: (n_z, \theta_z) \tag{10}$$

When $v_{cal,xy}$ is the vector part of q_{lean}^{stand}, projected into the x-y plane and then normalized, v_{ref} is the x-axis and the resultant quaternion $q_w^{R2Gmotion}$ (based on axis-angle to quaternion conversion, eq. (1)) is the relative orientation of the anatomical *R2Gmotion* CS w.r.t the sensor's world CS.

Finally, the transformation from the sensor orientation to the body-segment orientation at the defined anatomical CS can be obtained based on the acquired orientations of each sensor during the T-pose. As mentioned, during the T-pose the user is standing straight, the humerus and torso are perpendiculars, and the elbow flexion axis is parallel to the z-axis. This pose is defined as the initial orientation of all body segments CSs which means all systems are aligned (see Figure 13, hence the following calculation can be applied:

$$\begin{cases} q_{s_1}^{b_1} = \left(q_w^{s_1}\right)^* \cdot q_w^{R2Gmotion} \cdot q_{R2Gmotion}^{b_1}, & i = 1 \\ q_{s_i}^{b_i} = \left(q_w^{s_i}\right)^* \cdot q_w^{s_{i-1}} \cdot q_{s_{i-1}}^{b_{i-1}} \cdot q_{b_{i-1}}^{b_i}, & i = 2,3,\dots \end{cases} \tag{11}$$

When $i = 1$ is the first kinematic chain, in our case the thorax, $q_{b_{i-1}}^{b_i}$ is the relative orientation of the distal body-segment w.r.t the proximal body segment, which equals to [1,0,0,0] during the T-pose, and $q_w^{s_i}$ is the raw output of the sensor. Notice that eq. (11) is a recursive equation and $q_{s_1}^{b_1}$ is the relative orientation of the thorax IMU sensor w.r.t the thorax coordinates system, which is aligned to the *R2Gmotion* defined CS, meaning $q_{R2Gmotion}^{b_1} = [1,0,0,0]$. Although the above procedure was done only on the calibration poses, since we assume the orientation of each sensor w.r.t the body-segment it is placed on is constant as along as the user does not remove the system from him or her, we can use the produced transformation to estimate the upcoming anatomical joints angles.

4.3 Relative Joints Angles Calculation

After we obtained the relative orientation of each sensor to the body-segment it is attached to, and by assuming the sensors are not moving w.r.t the segment (neglecting small movements of the sensors' straps along the arm and soft tissue artifacts), the joints angles can be calculated. According to the ISB recommendations [30], the upper-body joints angles are defined as the distal segment orientation w.r.t the proximal segment orientation according to the distal segment CS (intrinsic rotation). The anatomical joints' angles for the kinematic chain of the right upper body can be calculated according to eq. (12)

$$\begin{cases} q_{R2Gmotion}^{b_1} = \left(q_w^{R2Gmotion}\right)^* \cdot q_w^{s_1} \cdot q_{s1}^{b1}, & i = 1 \\ q_{b_{i-1}}^{b_i} = \left(q_{s_{i-1}}^{b_{i-1}}\right)^* \cdot \left(q_w^{s_{i-1}}\right)^* \cdot q_w^{s_i} \cdot q_{s_i}^{b_i} \cdot q_{b_{i-1}}^{b_i}, & i = 2,3,\dots \end{cases} \quad (12)$$

Same concepts as for eq. (11), $i = 1$ represent the thorax segment which is the first kinematic chain, and $i = 5$ is the wrist. $q_w^{s_i}$ is the raw output of the sensor, and $q_{s_i}^{b_i}$ was calculated during the Sensor-to-Body-Segment Calibration (see eq. (11)). Here, $q_{b_{i-1}}^{b_i}$ is the joints angles estimation in the *R2Gmotion* coordinates system, which is the desired result of this whole data analysis chapter.

Chapter 5

Systems Comparison

5.1 Time Synchronization

To synchronize the two systems, VICON™'s and ours, we used two methods. The first one was by acquiring the data of a push-button by both systems. The subjects were asked to press a button at the beginning of each experiment. This 'sync button' was connected (by wire) to our wearable *R2Gmotion* system as a trigger to start the data acquisition and to ZeroWire, a wireless device that can be integrated with VICON™ system. The time synchronization was done by detecting the rising edge of the button pressing acquired by VICON™ system as an analog value.

Due to some connectivity issues with ZeroWire, the participants were asked to tap three times on the table with their hands open. This way a marker and the fifth IMU that were placed on the back of the participants' palm were sensing similar wave-like movements and the 'dynamic synchronization process' was apply.

5.1.1 Dynamic Synchronization Process

The dynamic synchronization process is based on a cross-correlation between the normalized linear acceleration norm of both marker and IMU placed on the back of the subjects' palm during tapping. The IMU sensor already outputs the linear acceleration, hence the required operations were only to calculate the vector norm of each relevant timestamp and normalize it according to the min-max method [see section 6.2.1], so the data is mapped to $[0,1]$. The market data, i.e., VICON™'s output is the marker position in space, hence, first, we applied numerical differentiation of 2^{nd} order with a window width of 4 samples according to eq. (7). Then we down-sample the estimated marker acceleration to match the *R2Gmotion* system sample rate (4-times slower) by using linear interpolation. Same as we did for the IMU data, we calculated the vector norm and normalize it according to the min-max method. At this stage, we calculated the cross-correlation of the two datasets which represent roughly the same movement and as a result, we obtained the time-lag between the two datasets by finding the index of the maximal cross-correlation value.

5.2 Transformation of Joints Coordinates

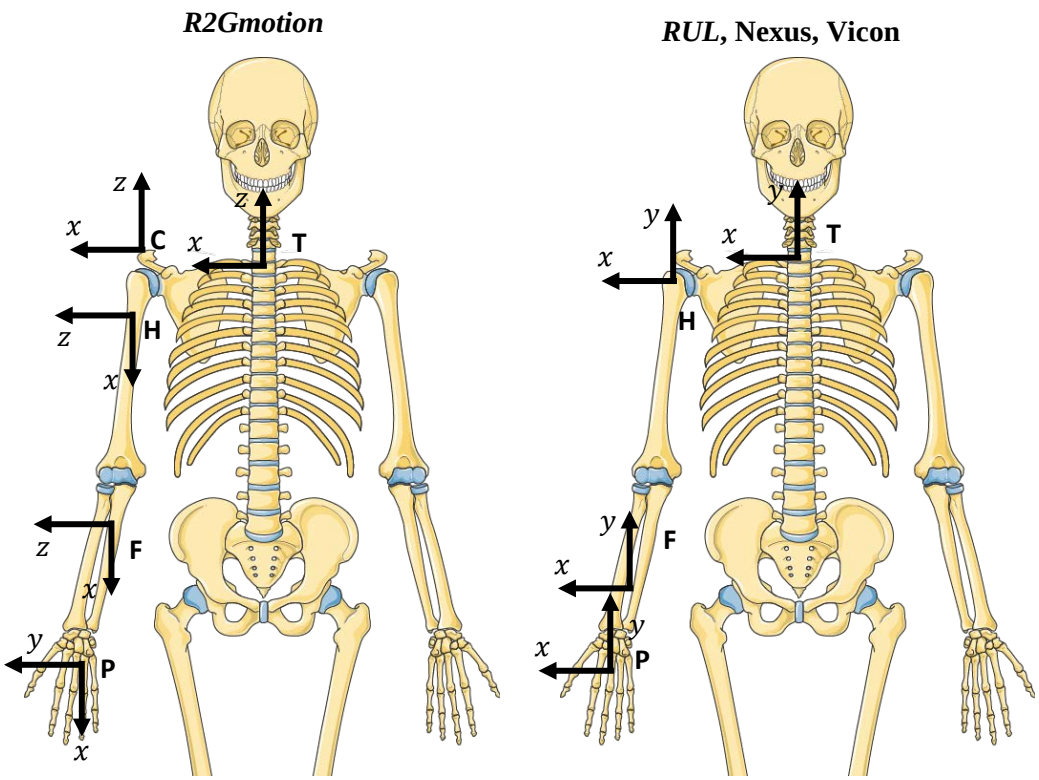
When a kinematic comparison between two different data-acquisition systems is required, there is a need to establish a coordinates transformation. Since the same orientation in space can be expressed completely differently depended on the coordinates system definition, a decision of several conventions is necessary. Section 4.2.1 describes in detail the *R2Gmotion* coordinates system but lacks the comparison with VICON™'s definition. The biomechanical initial position of VICON™'s RUL model is in accordance with ISB recommendation [30] – standing straight while both arms rest to the body sides and the palms are facing forward (see Figure 14). For better understanding Figure 14 presents both VICON™'s and our system at the described initial position. Remember, that *R2Gmotion* initial position is the T-pose, hence a transformation that rotates each of the suggested *R2Gmotion* body-segment coordinates system into the defined VICON™'s RUL model coordinates systems was developed. Eq. (13) can be obtained by observing Figure 14 and finding the relative rotation between the IMU-based system w.r.t VICON™'s system:

$$\begin{cases} q_{Tv}^{Ti} = \{R_x(-90)\} \\ q_{Hv}^{Hi} = \{R_z(-90) \cdot R_x(-90)\} \\ q_{Fv}^{Fi} = \{R_z(-90) \cdot R_x(-90)\} \\ q_{Pv}^{Pi} = \{R_z(-90)\} \end{cases} \tag{13}$$

When T, H, F, P indicates the body-segments at both models – Thorax, Humerus, Forearm, Palm and the i and v annotations are for IMU-based model and VICON™'s model, respectively. Finally, to find the conversation between the estimated joint angles by the system we developed and by the RUL VICON™'s model, we used similar tools and relations presented in section 4.3, in the following equation:

$$\begin{cases} q_{\text{sho},v} = q_{Tv}^{Hv} = q_{Tv}^{Ti} \cdot q_{Ti}^{Ci} \cdot q_{Ci}^{Hi} \cdot \left(q_{Hv}^{Hi}\right)^* = q_{Tv}^{Ti} \cdot q_{clav,i} \cdot q_{\text{sho},i} \cdot \left(q_{Hv}^{Hi}\right)^* \\ q_{elb,v} = q_{Hv}^{Fv} = q_{Hv}^{Hi} \cdot q_{Hi}^{Fi} \cdot \left(q_{Fv}^{Fi}\right)^* = q_{Hv}^{Hi} \cdot q_{elb,i} \cdot \left(q_{Fv}^{Fi}\right)^* \\ q_{wri,v} = q_{Fv}^{Pv} = q_{Fv}^{Fi} \cdot q_{Fi}^{Pi} \cdot \left(q_{Pv}^{Pi}\right)^* = q_{Fv}^{Fi} \cdot q_{wri,i} \cdot \left(q_{Pv}^{Pi}\right)^* \end{cases} \tag{14}$$

Since the RUL model excludes the clavicle (indicates as C) as a body-segment the shoulder angle transformation includes the humerus orientation w.r.t the thorax. By applying eq. (14) on the estimated *R2Gmotion* system joints angles, the corresponding RUL model joints' angles can be produced. This method assists to validate our system to the gold-standard motion capture system.



https://smart.servier.com/smart_image/skeleton-3/

Figure 14. **The joints coordinates system definitions** of our system (*R2Gmotion*) and VICON™'s at the initial position of the RUL VICON™'s model according to ISB recommendations

Chapter 6

Machine Learning

6.1 Problem Formulation

Training a net requires a quality database that exemplifies best the problem or the phenomena we want to investigate. Consequently, R2G movements were gathered during the experiment and processed to estimate the joint's angles. Since during the experiment we acquired continuous data, meaning not only the R2G but also the pick-and-place movements, there was a need to extract only the relevant data-points to the purpose of the machine learning algorithm – estimating the R2G stage and predicting the grasp moment. Therefore, we established a clear definition of the reaching moment and the grasping moment. For this reason, the two systems the VICON™'s system and our system were synchronized. The starting point of the R2G movement, the reaching moment, was defined as the instruction time managed by our app, and the endpoint of the R2G movement, the grasp moment, was defined according to the objects' markers movements. The high-level process is shown in Figure 15.

6.1.1 Grasp Moment Estimation

The grasp moment definition may change from one study to another, and here we decide to calculate it as a combination of four decision parameters: the hand distance from the object and the object's kinematics – position, linear velocity, and acceleration. Per each grasp, a segment of 1 second before the instruction until 5 seconds after the instruction was examined based on VICON™'s markers data. These four decision parameters were normalized according to the min-max method [see section 6.2.1], where the kinematics parameters were normalized by removing the baseline instead of the minimum value. Then, thresholds values were selected by trial-and-error for each parameter, so that the suggested grasp moment is when the distance from the subject's palm to the object is minimized (below specific threshold) and the kinematics parameters started to rise (e.g., the velocity of an object increases when the subject starts to move it). Due to noises at the kinematics values which were amplified after the numeric differentiation, we first found the peak and evaluated the samples before that. At this point, we have 4 different suggested grasp moments according to each parameter that needed to be combined. The final grasp moment was defined as the average of all suggested grasp

moments after excluding results that were significantly different (more than half a second) from the major of the group.

6.1.2 Features Extraction

The final step was to split the data from the 'Grasping Task' into different R2G sections, according to the instructions time and the estimated grasp moments. Then we extracted random sections with similar lengths from the 'Walking' and 'Talking' trials to construct non-R2G movement data.

From these data sections the kinematics of each anatomical joint – thorax, clavicle, shoulder, and elbow, was estimated by implementing eq. (5) that calculates the angular velocity, acceleration and jerk from the quaternions which represent the estimated joints' anatomical angles. The kinematics outcomes result in the following features in three dimensions: the angles as a vector at the direction of the rotation axis and its norm equal to the angle value, the angular velocity, acceleration, and jerk according to the anatomical CS and at the distal segment (i.e., rotating system or intrinsic rotation). The thorax was the only exception, since the only relevant angle was the torso flexion, while the lateral and axis rotations were irrelevant due to the calibration method we developed [see section 4.2 for more details]. The data was further split into windows to simulate a real-time manner, so when each sample arrives a calculation based on only past time samples can be done. Each window is considered as one input of a learning-based algorithm whose purpose is to predict the R2G stage of the whole process – from 0, the reaching moment toward 1, the grasp moment, and in turn the grasp moment.

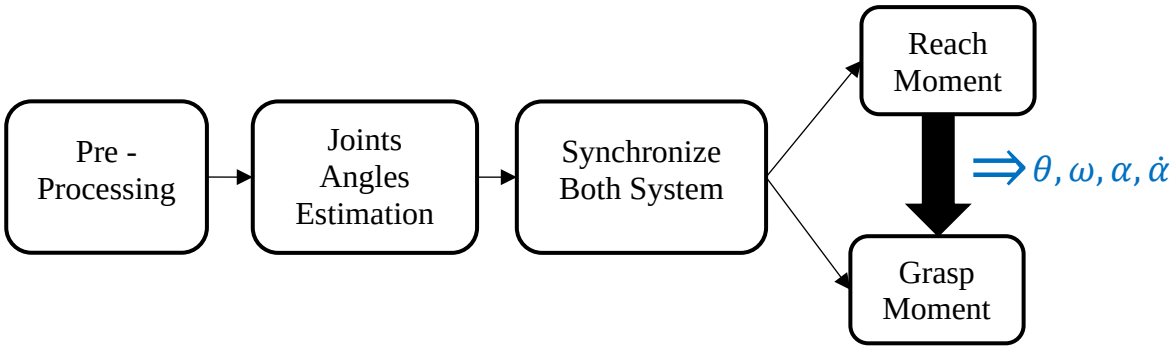


Figure 15. **The preprocessing** applied on the raw data to extract R2G kinematics sections

6.1.3 Testing Methods

The machine learning field is a stochastic discipline and subsequently, there are many ways to evaluate the learning algorithm results. In this research, we are testing our results according to three methods. The first is the basic method which is splitting the dataset into train and test datasets. The

net is trained on the training dataset, then tested on the test dataset by applying only feed-forward, i.e. the prediction process, and the net's prediction is compared with the true-label of the specific sample. Although we conducted 20 different experiments, since each subject came on two different days we wanted to test the system on completely new users. Hence, in this method, we randomly selected 3 out of 10 subjects to be the test group and all other samples were used to train the net. From the moment the test group was randomly selected all suggested architectures were tested on the same test group.

The second method is called leave-one-out and in this method, the dataset is split into *k* different groups and each time the net is trained on *k-1* groups and tested on the k^{th} group. This circuit is done *k* times and the result is a distribution of results.

The third method has tested the algorithm running time and more specifically the predicting time ('forward propagation') to emphasize the importance of the calculation time for a real-time application.

6.2 Suggested Architectures

Initially, we tried to solve the problem using basic, well-known learning methods. We tried to use classifiers such as KNN – k nearest neighbors, SVM – support vector machine, and RF – Random Forest, to predict the grasp moment, where all the data was labeled zeros and only the windows at the end of the R2G (the grasp moments) were labeled ones. The classification results corresponding with the above problem formulation are described under the Results section in Table 5.

Furthermore, we run RNN – recurrent neural network method as regression estimator to predict the R2G stage, utilizing LSTM – long short-term memory and GRU – gated recurrent unit. The data from the non-grasping section ('Walking'/ 'Talking') was considered as 0, and the R2G stage of each window from the R2G section was calculated according to the starting and ending points, by a simple uniform time normalization. Meaning, the R2G stage was considered as a continuous value from 0 to 1, so each sample of the R2G section was labeled according to the segment length. For instance, if an R2G section was 100 samples, from the reaching moment to the grasp moment, then the 72^{nd} sample was labeled as 0.72 R2G stage. The regression results corresponding with the above problem formulation are described under the Results section in Table 4.

To further improve the results, a different approach was suggested. The main idea was to combine the two architectures presented before. In the new design, the purpose of the classifier is to distinguish between R2G movements and all other movements, hence we trained the classifier on all the data where any window from R2G movement was label TRUE and any window from the 'Walking'/ 'Talking' section was labeled FALSE. Then assuming an R2G movement was detected, the RNN task

is to estimate the R2G stage. Meaning during the new design we trained the RNN on only the R2G dataset according to the grasp stage of each window. Thanks, to this method the accuracy of the results improved as described in Table 6 and Table 7.

Regardless of the two different problem approaches, there are many parameters of each machine learning algorithm that can be modified to achieve better results. For the suggested classifier we change the number of neighbors (KNN), the number of trees (RF), the loss function of SVM between 'hinge' and 'log', and in general change the data balance of the classification groups. For the suggested RNN, we tried a different number of gated units, different layer architectures such as the number of layers, the last layer type (GRU or Dense), and the activation function of the last layer (linear or sigmoid). While investigating the regression problem we tried different inclusion criteria for the input data such that only samples with R2G stage larger than X% (e.g., 50%) are part of the training dataset. The logic was that the reaching stages (i.e., smaller R2G stage) have more variance than the grasping stages (i.e., larger R2G stages) since the reaching moment was defined as the instruction time. Moreover, predicting the end staging of the R2Gmovement, meaning the grasp moment, is more important for prosthetic hand control hence the focus of the algorithm should be at these stages.

6.2.1 Normalization Methods

For both classifier and RNN, we investigated different linear normalization methods. The first is the min-max normalization method which maps the data between 0 to 1, by finding the minimum and maximum values of the features data and can be calculating according to eq. (15):

$$x_{norm} = \frac{x - \min}{\max - \min} \tag{15}$$

The second method is z-score which maps the data to a normally distributed data $\sim N(0,1)$. By calculating the mean (i.e., μ) and standard deviation (i.e., σ) of the features data, the normalized data can be generated according to eq. (16) :

$$x_{norm} = \frac{x - \mu}{\sigma} \tag{16}$$

For both methods, the normalization process was per each feature and the constants required for the normalization were calculated based on the training dataset only (and not recalculate for the test dataset).

6.2.2 Decision Making

The *R2Gmotion* system functionalities block diagram with the emphasis on the decision-making junctions is described in Figure 16. The system flow is as follows – a new raw quaternion-based sample from all IMU sensors is acquired, then a preprocess is applied on the new data-point sample combined with a buffer of the last x data-points to calculate the kinematic data (i.e., the features). Then the system classifier predicts according to the new kinematic data the probability of the new sample to exemplify an R2G movement. According to two thresholds – one for the sample itself and one for the whole buffer of samples (i.e., a window of x samples), the system decides if the current data window is part of the R2G movement. For instance, if the first threshold thr_{class} is 0.7 and the classifier outputs a probability of 0.8, which means that the current sample is considered as R2G. Next, the system checks the current window. If for instance a window is defined as 10 samples, and if the second threshold, thr_{win} is 0.7, then if 7 or more out of the last 10 samples are labeled as R2G, the system decides the current window is an R2G window. If indeed the current window is considered as R2G window, then the system's estimator (i.e., RNN architecture) predicts the grasp stage according to the same kinematic data. If the grasp stage is above a certain threshold command is sent to the hand to close all fingers.

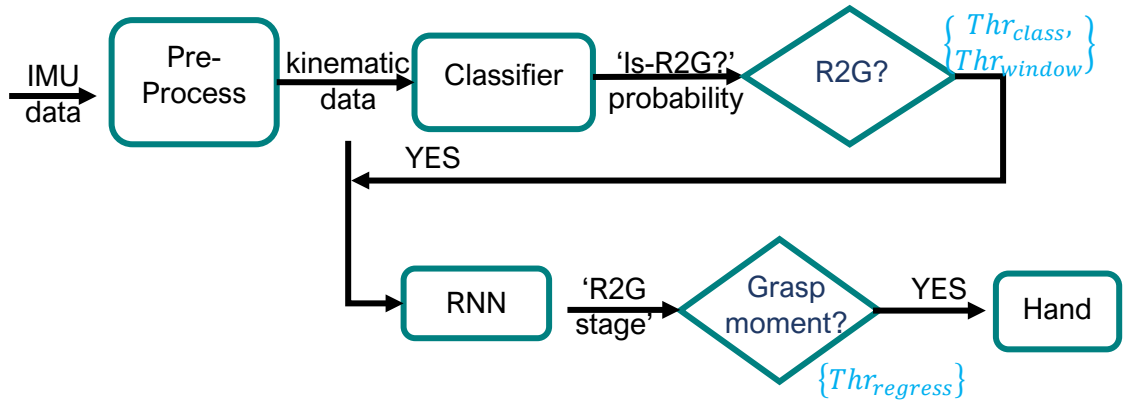


Figure 16. **The *R2Gmotion* system block diagram overview**

A link for a video demonstration of the *R2Gmotion* system can be found in Appendix B. The demonstration includes the *R2Gmotion* wearable and wireless system, and the action of the prosthetic hand in correlation with the moment an object was grabbed including a demonstration when other movements occur, but the hand is kept still.

6.3 The Selected Architecture

According to the results presented in the Results section, the following architecture was selected:

- Random Forest with 10 trees, and z-score normalization, classify the data into two groups - R2G movement and non-R2G movement (i.e., other movements).
- 1st GRU layer of 2 units with ReLu activation function, 2nd GRU layer of 1 unit with Sigmoid activation function, after applying z-score normalization, estimating the R2G stage of a 250-millisecond window (10 samples window for 40 Hz sample rate) of classified R2G samples.
- Three threshold values for the system's decision-making can be customized according to the application and user's needs.

[see model and normalization constants in Appendix C]

Chapter 7

Results

This section presents the Thesis results by first describing the data on which the research is based, then validating the mathematical models of our *R2Gmotion* system by comparing it to a well-established motion capture system, and finally describing the main results of this research – the R2G stage estimation and the grasp moment predicting. In the end, an extra quality result about the generalization of the developed architecture on a different dataset is presented.

7.1 Database Overview

During the experiment we gathered 3,540 R2G movements and extracted 7,080 other movements from random sections of ‘walking’ and ‘talking’ sessions, added up to a total of 10,620 sections. Each section length is different, and the total acquired time-points by our *R2Gmotion* system of 40 [Hz] sample rate is 475,923 time-points, when 158,641 are samples from R2G movements and 317,282 are samples from other movements. See the full experiment description in section 3.3.

The data characteristics distribution is important to understand which tasks can be represented according to the gathered information and which tasks are excluded from this scope. Nevertheless, in the Generalization section, we present the capabilities of the system to handle different types of data which potentially can prove a wider scope for our system.

First, the time duration of each R2G movement, from the reaching moment until the grasp moment, was measured and presented in Figure 17. Naturally, most R2G movement durations are longer than 0.5 seconds (>99%) and the mean duration is 1.1 ± 0.3 seconds. Hence, we can assume each R2G pattern can be exhibited as a time-scaled pattern between 0.5-2.5 seconds, according to the gathered data distribution (>99% of the data).

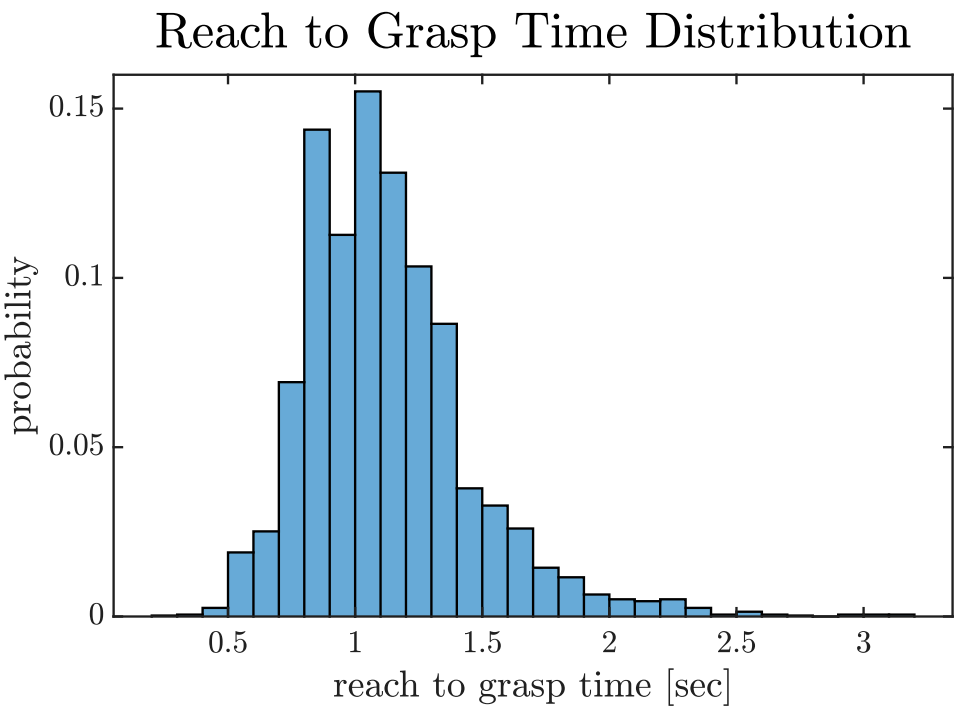


Figure 17. **R2G time variety**. The duration distribution of all R2G movements data

Second, we validated the distribution of each object's location to eliminate the possibility that the experiment conclusions are based on biased input data. As can be seen from Figure 18, there is a uniform distribution of all objects' locations at all possible locations, as required. According to the experiment protocol, there are 8 possible locations each object can be randomly assigned to during two different pick-and-place tasks. The optional locations are defined to be 8 locations along a half-circle around the subject as described in Figure 9 when the circle radius has two optional sizes, so the objects are located either close or far from the participant. The two different radius sizes were defined according to the participant total arm length to enrich the R2G data in another dimension. When validating the normalized initial distance between the participant's torso location and the objects' location at the XY plane (not including the height differences between the torso and the table, since the chair was adjusted according to the participant height), the measured two radius sizes in Figure 19 are 0.5 ± 0.2 and 0.7 ± 0.2 object distance per total arm length, as mentioned in the experiment protocol (see 3.33.2).

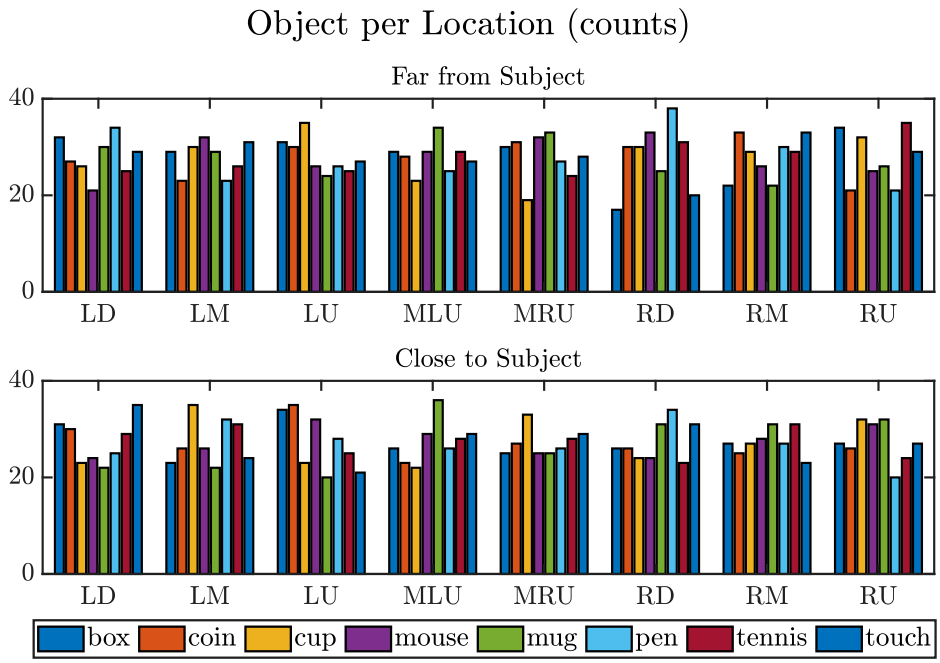


Figure 18. **R2G objects variety**. The object's location distribution of all R2G movements data

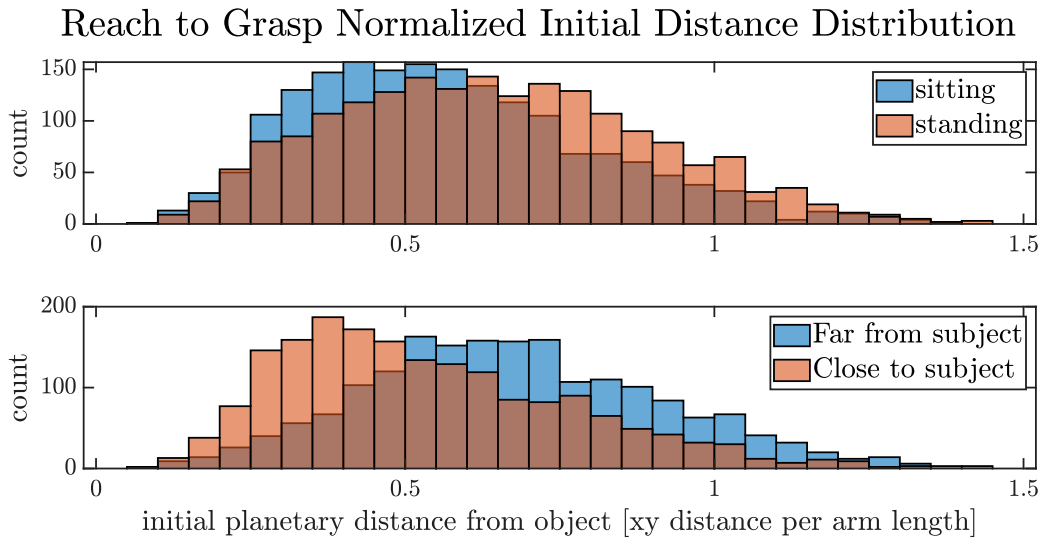


Figure 19. **R2G distance variety**. The distribution of each R2G initial planetary distance between the torso and object

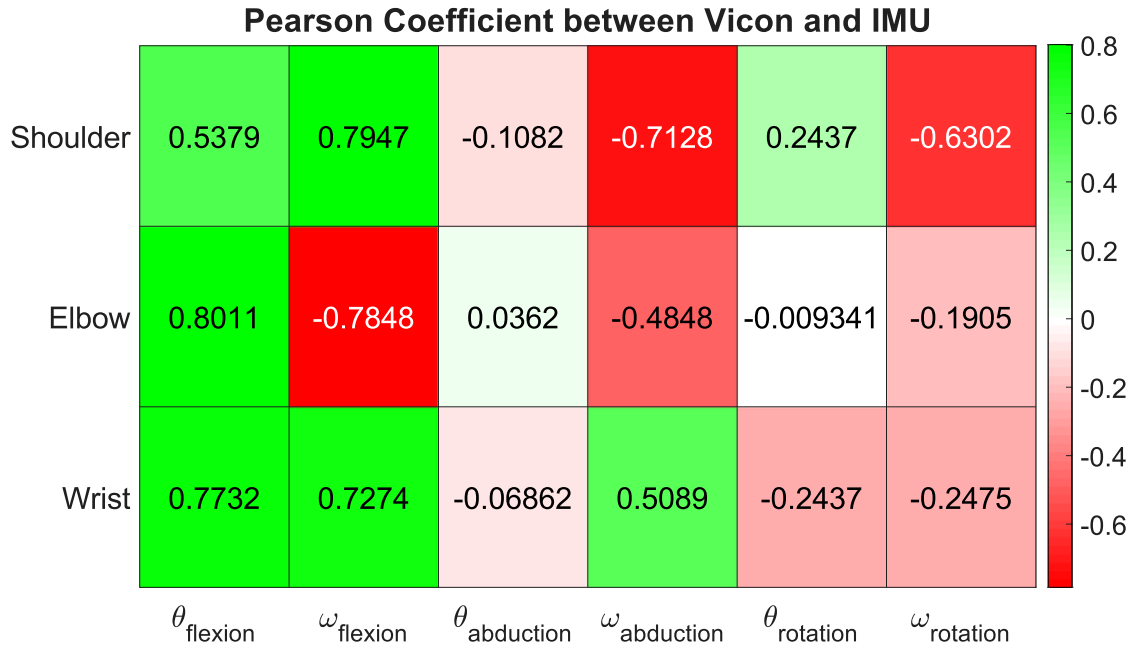
7.2 Validation

One of the main purposes of capturing the kinematics movement by the VICON™ system in addition to our *R2Gmotion* system was to validate the estimated joints angles according to the method we developed in comparison to the RUL model, NEXUS™, VICON™. Since we defined different CSs not only by rotation but by the position as well, it is not trivial to compare the joints angles values. For

instance, as presented in Figure 14, the RUL model does not have the clavicle joint as part of the upper body kinematic chain. Moreover, the forearm coordination system in the RUL model is defined to be at the distal end of the segment which means the elbow joint also includes the pronation-supination of the palm. Indeed, this approach is in line with the ISB recommendation of the upper body joints definitions[30] but is not necessarily adequate to the case of transradial amputation when the pronation-supination range of motion is only partial to not exist. Hence, we defined the forearm coordination to be at the proximal end just below the elbow. Due to the complexity of the radius-ulna movement and since angles' values in space are depended on the choice of the CS and its order, the comparison between two different coordination systems on the forearm was challenging.

But keeping in mind the main goal of the system – to predict the R2G stage and more specifically the grasp moment, in contrast to estimating the accurate joints angles, we chose to apply between our system and the state-of-the-art, VICON™ system a statistical comparison.

Here, we apply cross-correlation per each estimated joint angle along all time-points and calculate the Pearson coefficient between the *R2Gmotion* joint angle estimation and RUL™ VICON™ estimation. The values of Pearson coefficient [see eq. (8)] per each joint angle are represented as a colormap, in Figure 20. Hence the strong green or red values represent a high linear correlation at the same direction or the inverse direction, respectively. All values in Figure 20, are considered significant with $p - value \ll 0.001$. More specifically we can see that the shoulder's angular velocity at all direction and the flexion velocity and angle of the elbow and wrist have a high correlation between *R2Gmotion* and VICON™ ($> |0.7|$). In contrast, the elbow's rotation angle estimation by both systems is most likely not linearly correlated (~0.009), as expected according to the aforementioned difference in the forearm coordination system definition.



** all the above correlation coefficients are statistically significant (p<<0.001)*

Figure 20. **The cross-correlation results between *R2Gmotion* and VICON™** for each joint angle and velocity. A stronger color indicates a higher linear correlation (in the same direction or opposite direction).

A different way to validate our system with VICON™ is to compare the results of the selected learning algorithm we developed (see 6.2 for more details about the architecture selection). The data was split into train-test by choosing 7 out of 10 random subjects as the training dataset and the other 3 were considered as the test dataset. We train the net on the estimated joints angles calculated by *R2Gmotion* based on the acquired IMU data and extract the test loss results using the test dataset. Then the same architecture was used on the VICON™'s estimated joints angles and the test loss was extracted as well. It can be seen from Table 3 the classification success parameters – sensitivity and specificity are very similar between the IMU-based data and VICON™'s data. The algorithm has a better sensitivity rate when predicting if a sample is R2G or not based on VICON™ data in comparison to the IMU data, while the specificity value is better for using the IMU rather than the VICON™ data. For the regression problem of predicting the R2G stage, the test loss is smaller when the input data is based on the VICON™ data RUL model rather than the *R2Gmotion*, by more than 20%.

Table 3. Comparison between IMU-based input data and VICON™'s input data for net results

Modality	**Grasp Classification - Test loss**		**Grasp Stage (grasp only) – Test Loss**	**Grasp Stage (include all) – Test Loss**
	Sensitivity (TPR)	Specificity (TNR)		
VICON™	91.04%	69.74%	0.0262	0.0312
R2Gmotion	90.24%	75.60%	0.0337	0.0490

From here on, all results are based on the IMU-data only that was acquired by our system *R2Gmotion*.

7.3 Learning

As we defined in 6.1 the main task of our learning algorithm is to predict the grasp moment and to estimate the R2G stage. The results of the investigated architectures were calculated by randomly selecting 3 out of 10 subjects of the experiment to be the test dataset and the other 7 subjects were the training dataset.

As a first step we started by utilizing a simple approach (see 6.2, for more details) – all data is considered as input data while an RNN-based algorithm is designed to predict the R2G stage, solving a regression problem and a classifier-based algorithm is designed to classify the grasp moment from all other time points, solving a classification problem. Table 4 and Table 5 presented several architectures we investigated to answer the regression problem and the classification problem, respectively. For both problems, the input data were normalized according to the z-score method (i.e., transform the input data to be normally distributed, see 6.2.1). In addition, since the classification problem suffers from significant imbalanced data, we selected randomly the same number of time-points for non-grasp movements to match the number of grasp movements (total of 3,540 grasps).

The best-achieved result for predicting the grasp stage based on a window of 250 milliseconds samples including all data-points is 0.0387 Mean Square Error (MSE) for the test dataset (see Table 4). To distinguish the grasp moment from all other moments by classification, the best-achieved result was 79.48% sensitivity (True Positive Rate, TPR) and 88.75% specificity (True Negative Rate, TNR) for the test dataset (see Table 5).

Table 4. collection of results for predicting the R2G stage by training on all the data

	Layer Architecture	**Test loss [MSE]**
GRU	GRU, 10 units, ReLu GRU, 10 units, ReLu GRU, 1 unit, Sigmoid	0.0447*
GRU	GRU, 10 units, ReLu GRU, 1 unit, Linear	**0.0387**
GRU	GRU, 2 units, ReLu GRU, 1 unit, Linear	0.0490
LSTM	LSTM, 10 units, ReLu LSTM, 1 unit, Linear	0.0432*

** overfitted according to training loss value*

Table 5. collection of results for predicting the grasp moment by training on all the data

	Normalization method	**Balanced Data**	**Classifier Architecture**	**Test loss**	
				Sensitivity (TPR)	**Specificity (TNR)**
Random Forest	z-score	V	10 trees	67.08%	86.35%
Random Forest	z-score	V	50 trees	78.23%	87.81%
Random Forest	z-score	V	100 trees	77.77%	**88.75%**

SVM	z-score	V	Loss: log	73.54%	83.65%
SVM	z-score	V	Loss: hinge	**79.48%**	81.35%

To improve the results, we chose a different approach by utilizing a more complex net to solve two simpler questions than the one we started with (see 6.2, for more details). The results of the combined net architectures can be seen in Table 6 and Table 7.

The best-achieved result for identifying if a sample is part of an R2G movement by utilizing different classifier architectures is 95.68% sensitivity (TPR) and 83.67% specificity (TNR) for the test dataset (see Table 6). The investigated variables of the classifier were the normalization method (z-score or min-max), data balance between R2G and non-R2G movements, for RF the trees numbers, and for SVM the loss function (hinge or log). For more details on each parameter see section 6.2. For the suggested *R2Gmotion* system we selected the architecture with a balanced result between the sensitivity and specificity marked with a blue background in Table 6 (90.24% sensitivity, and 75.60% specificity).

In parallel, the best-achieved result for predicting the grasp stage based on a window of 250 milliseconds samples from R2G movements only (not including data from 'Walking' and 'Talking') is 0.0337 Mean Square Error (MSE) for the test dataset (see Table 7). For comparison, a uniformly random guess (between 0,1) was tested on the data and produced an MSE of 0.272, an order of magnitude bigger w.r.t the investigated architecture results.

Table 6. Results of R2G classification from other hand movements by training on all data

	Normalization method	**Class weight [T:F]**	**Classifier Architecture**	**Test loss**	
				Sensitivity (TPR)	**Specificity (TNR)**
Random Forest	z-score	1:2	10 trees	90.24%	75.60%
SVM	z-score	1:2	Loss: hinge	80.88%	74.82%
SVM	z-score	1:2	Loss: log	78.09%	77.69%
Random Forest	z-score	1:2	100 trees	**95.68%**	67.72%
Random Forest	z-score	1:2	5 trees	91.27%	68.68%
Random Forest	z-score	1:2	20 trees	93.94%	71.31%
Random Forest	Min-max	1:1	10 trees	89.35%	73.47%
Random Forest	Min-max	1:2	10 trees	89.38%	74.79%
SVM	Min-max	1:1	Loss: log	88.38%	62.71%
SVM	Min-max	1:2	Loss: log	75.67%	**83.67%**
Baseline [Random [0,1]]				49.67%	50.04%

Table 7. Results of predicting R2G stage by training on R2G data only

	Normalization method	Layer Architecture	Test loss [MSE]
GRU	z-score	GRU, 100 units, ReLu GRU, 1 unit, Linear	0.0374*
GRU	z-score	GRU, 10 units, ReLu GRU, 1 unit, Linear	0.0341
GRU	z-score	GRU, 2 units, ReLu GRU, 1 unit, Linear	0.0346
GRU	z-score	GRU, 2 units, ReLu GRU, 1 unit, Sigmoid	**0.0337**
GRU	Min-max	GRU, 2 units, ReLu GRU, 1 unit, Sigmoid	0.036
GRU-Dense	z-score	GRU, 10 units, ReLu Dense, 1 unit, Linear	0.056
LSTM	z-score	LSTM, 2 units, ReLu LSTM, 1 unit, Linear	**0.0323**
Baseline [Random [0,1]]			0.272

** overfitted according to training loss error value*

For evaluating the robustness of the results of the selected architectures, we conducted the leave-one-out method where each time the system trained on 9 out of 10 participants' datasets and was tested upon the 10th participant's data. The results are presented as a boxplot, a compact representation of a value distribution including the median and 25th and 75th percentiles, where the R2G stage prediction is shown in Figure 21 and the R2G classification is shown in Figure 22. The median success rate of our suggested architecture to classify R2G movement is 89.25% sensitivity and 85.28% specificity while the median MSE of the R2G stage prediction when assuming the sample was acquired from an R2G movement is 0.031.

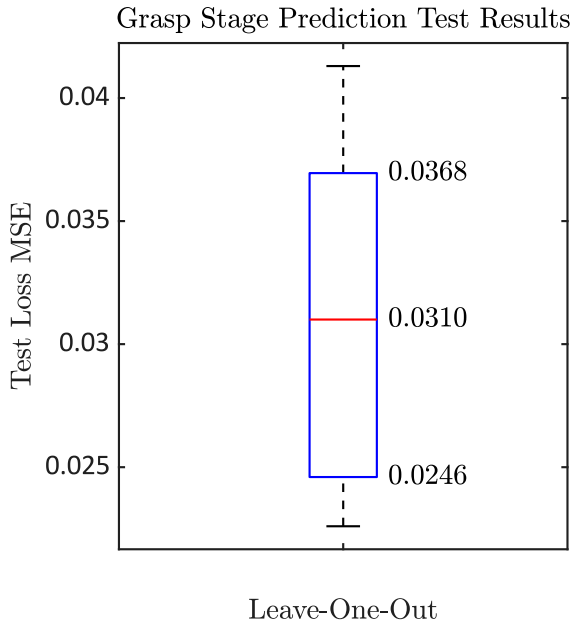


Figure 21. **The MSE distribution of R2G stage estimation**. The red line indicates the median and the blue end lines are the 25th and the 75th percentile. Low values of test loss MSE indicate better results.

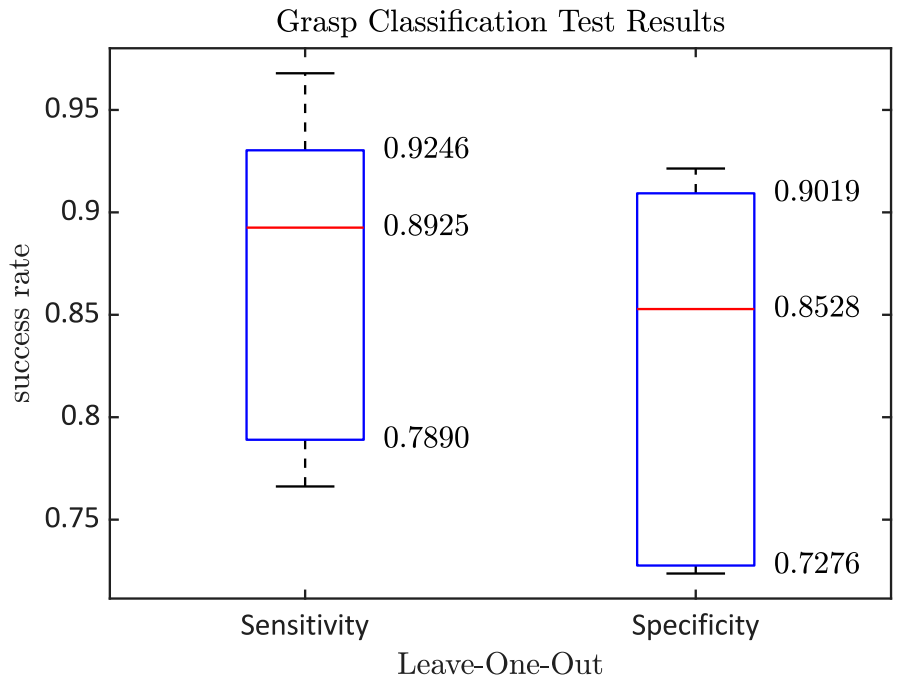


Figure 22. **The success rate distribution of R2G classification.** The red line indicates the median and the blue end lines are the 25th and the 75th percentile. High values of sensitivity and specificity indicate better results.

After selecting the combined net architecture there are still two decision points that needed to be investigated. Since the classifier acquires a one-sample-point and outputs the probability of this sample to be a part of the R2G movement, the first decision point is to decide if the last window of samples is a part of the R2G movement. For this decision, we defined two thresholds – threshold per sample (i.e. classification threshold, t_c) and for the whole window (window threshold, t_w). If the classifier probability output of a sample is higher than the threshold thr_{class} then a sample is considered R2G and if the R2G samples and non-R2G samples ratio is higher than the threshold thr_{win} then a window is considered an R2G window. The second decision point is when an R2G window is processed by the RNN and the RNN outputs the estimated R2G stage, then the system decides if the last window is a grasp moment or not. For this decision, we defined a regression threshold, thr_{reg}, so if the estimated R2G stage is above it a grasp moment is detected (for more details 6.2.2).

Again, we randomly selected 3-out-of-10 subjects to be the test dataset and did a grid-search on the range of possible threshold values. We simulated real-time results according to the aforementioned decision making and in Figure 23, a zoom-in of the best results in terms of high TPR and low FPR is shown to reflect the effect of each threshold on the results.

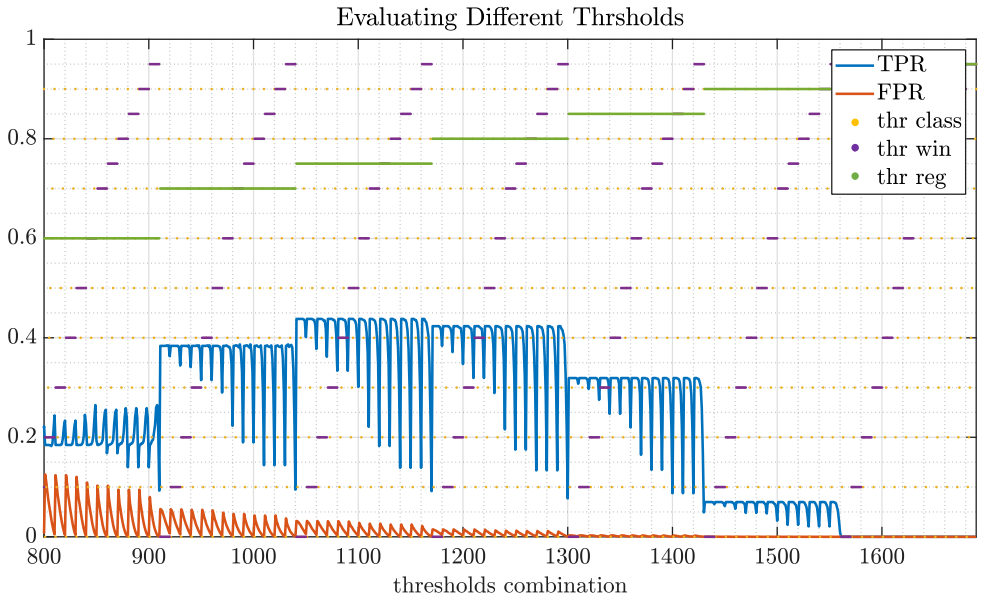


Figure 23. **Thresholds Optimization.** Optimizing the thresholds values for maximizing TPR (blue) and minimizing FPR (red), while each threshold combinations (x-axis) results in different success rates.

False-positive error, meaning detecting a grasp when the user did not mean to grasp is a huge problem specifically for prosthetic hands users. Hence, we limit ourselves to FPR under 2% which leads to detecting a grasp moment only if the level of confidence is high. Moreover, we needed to decide when a grasp moment detection is considered too early. This parameter does not have much meaning if it would be less than the human motoric response time, which is approximately 250 milliseconds. Naturally, in a real-life application, the user can detect if the hand starts to move before it is supposed to, and instinctively adopt the R2G movement accordingly. Thus, the results shown on the left side of Figure 24 are the worst-case results and by enlarging the 'Early detection time threshold' the TP values would be higher on the account of the 'earlyP' value, as it can be seen in Figure 24(right).

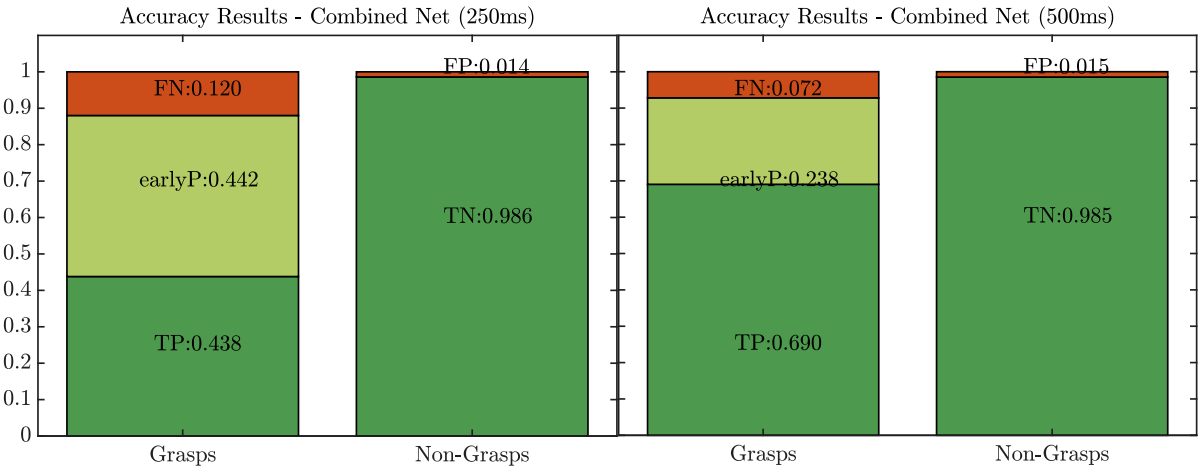


Figure 24. **Grasp Moment event detection results,** according to FPR<2% and early detection threshold of 250 and 500 msec. 'TP': the system identified the grasp moment accurately w.r.t the early detection threshold, 'earlyP': the system identified the grasp moment during an R2G movement but too early.

7.4 Generalization

Generalization is the ability of the system to produce an accurate prediction on data it has never seen before. Here we present a short demonstration on a dataset that was published in Nature – Scientific Data journal[15]. This dataset is a collection of three different systems – depth camera, RGB camera and motion sensor data based on XSENS suit, a high-end system that provides solutions for motion capture in outdoors environments. Our system was never trained on this dataset nor other XSENS outputs. Their experiment is based on extremely unstructured tasks, such as cooking in the kitchen. In Appendix B, attached a link with a movie clip that was taken from the mentioned dataset and published as a sample record. We edited the movie to demonstrate the resultant prediction from our *R2Gmotion* system, so when the system decides that a grasp moment has happened the screen is turned green. The predictions of our system were solely based on the IMU data, acquired with the XSENS suit. As shown in the movie clip, our system detects nicely whenever the participant reaches to grab an object with his right hand. Besides, it can be seen that when the participant places the object back to its place the system also detects it as a grasp moment. Although this is a wrong detection, it is expected due to the similar nature of the R2G movement and the reach-to-place movement. Naturally, this type of false detection can be prevented very easily by adding the logic of the last state of the hand (e.g., currently holding an object). Moreover, this type of false-detection can be harnessed in the future to detect pick-and-place type of grasp and also control the opening of the hand.

Chapter 8

Modeling R2G movement

In this section, we suggest additional insights, by treating our learning model as an interpretable model. As explained in the previous section, the chosen architecture of the learning algorithm is firstly built on classification R2G movements from all other movements by utilizing Random Forest classifier. For Random Forest there few approaches to determine the importance of the features. One of the approaches enables extracting the numeric value of the effect each feature had on the classifier output. There is not much meaning to the value itself but the relativity to other features' value. Figure 25 shows these features' importance values separated into the flexion, abduction, and rotation axes. It can be deduced that the most important feature to differentiate R2G from other movements is the synergy of the upper body joint's angles and not only a specific joint's angle. Yet, the thorax flexion during an R2G motion is the most important feature according to the experiment that we conducted. Notice that we constructed the experiment so the non-R2G motion was walking and talking. It would be interesting to enrich the dataset with the sit-down-stand-up task as non-R2G movements. During this movement, there is a slight flexion of the torso to balance the body, and indeed sometimes the system confuses this action as R2G movement depends on the other joints' position and motion. Also, the angular velocity of the shoulder abduction-adduction has a significant effect on the outputs. The clavicle angles are less important than the other joints, as expected.

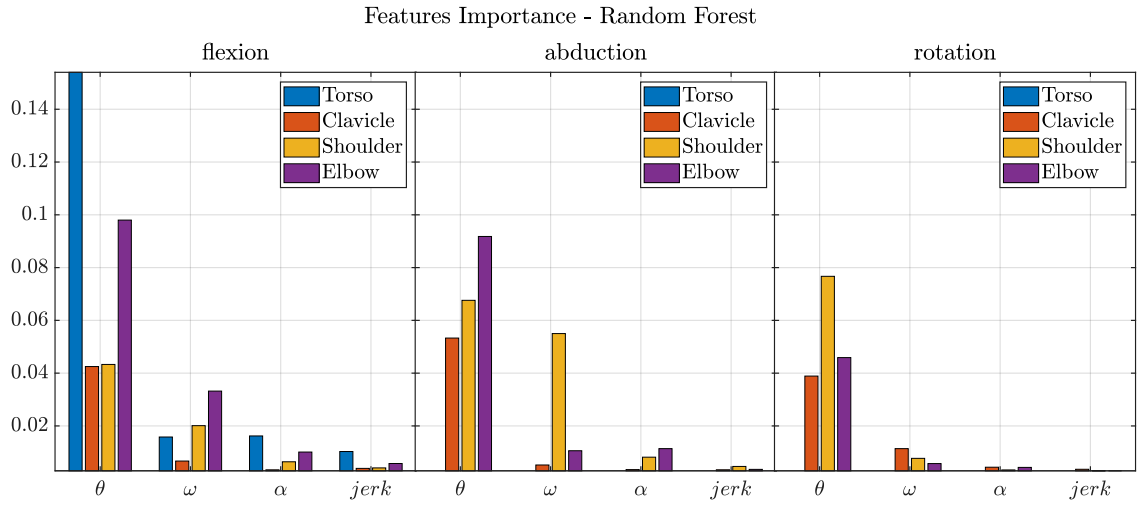


Figure 25. **Classifier Features importance** for distinguishing R2G and non-R2G samples. The y-axis value does have much meaning but the value of each feature with respect to other features is important.

The second part of the chosen architecture of the learning algorithm is the regression estimator of the R2G stage by utilizing RNN and more specifically GRU-based net. The feature's importance for estimating the R2G stage is less intuitive and can only be treated as a clue rather than proof. Here we investigated the net in two different ways, the first one was training the net on all features, but each time we dropped one joint and/or one kinematic feature. The results are presented in Figure 26, where each cell represents a different feature combination. The stronger the cell color hue is, the higher the test loss error and as a result, the significance of this feature is higher as well. Since if without a specific feature the test loss error is higher, this feature is presumably crucial for predicting the grasp stage. Without the velocity, the test error is higher hence it indicates that the velocity is important to estimate the grasp stage. Also, an interesting fact is that without the angles the test error is even less than when all features are presented, so it might indicate the angles information is confusing the net more than helping to predict the grasp stage. It can be the case since we used to enrich data of grasping from different angles and location exactly to prevent over-fitting toward a specific location at a specific pose. In addition, the clavicle data is probably less important since the test error is very similar with and without its information.

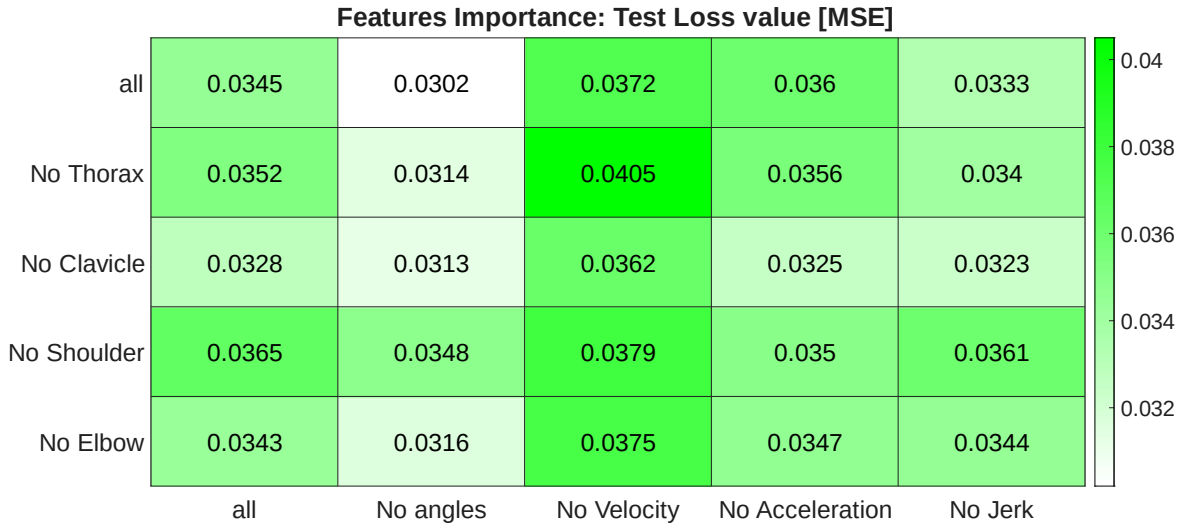


Figure 26. **Features importance for predicting R2G stage** according to test loss of different tests. The brighter the color, the smaller the MSE value which means better prediction results.

The second method is to compare the weights of the net per each feature. Although there is no broad agreement about the meaning of RNN calculated weights, since we normalized all features to be normally distributed, we might discover something about the relationship between them. In general, if the absolute value of the weight is larger the effect of each feature is larger as well. Even though it is not entirely correct and the selected architecture, GRU is not linear which makes it even more inaccurate it can give a hint or a gut feeling of the importance of the features. It can be seen from

Figure 27 that the Torso flexion angle and velocity, the Clavicle depression-elevation angle and velocity, the Shoulder abduction angle and angular acceleration, together with angular velocity for all directions, and the Elbow flexion-extension angle and velocity, have large weights values relatively.

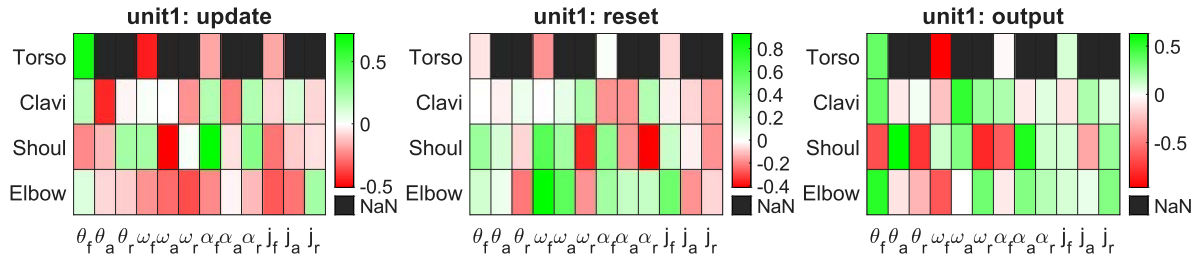


Figure 27. **GRU weights visualization.** An example of the 1st GRU unit's weights for the three gates w.r.t the kinematics features

On another topic, the R2G stage prediction is calculated from the extracted features, which are the numerical derivatives of the joint's angles. As explained in section 1.3.3 the numerical differentiation is a linear combination of the previous data-points. Each sample of the RNN architecture is a window of data-points, hence it raises a question if the calculation of the joints angles derivatives – the angular velocity, acceleration, and jerk, can be avoided by only enlarge the window inserted into the RNN. Notice, that the angular velocity is not a linear combination of the angle but a quadric-like relation [see eq. (3)], thus it is not completely trivial. Figure 28, presented several results of test loss error by enlarging the window size and training the net on all features and only the joints' angles. The selected architecture is based on a 250[ms] window and it can be assumed that for a window twice as big without calculating the kinematics derivatives the net would produce nice results as well.

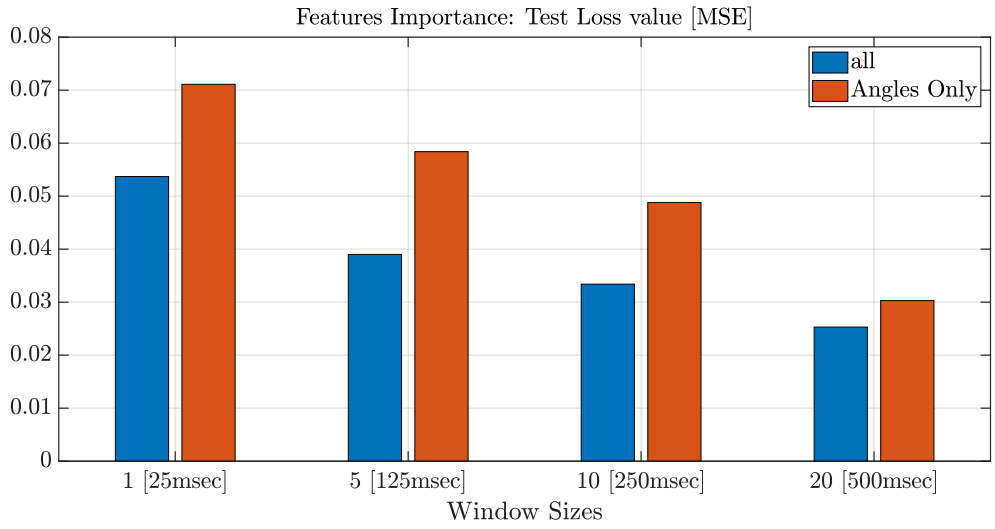


Figure 28. **The window-width importance**, of the number of evaluated samples for predicting the R2G stage. The lower the value, the more accurate the prediction is.

Chapter 9

Discussion

In this section, we discuss the main and most important results of this research. The focus of the research can be separated into two main topics – the system itself and the learning algorithm. The system itself, the *R2Gmotion* system is a proof-of-concept in which a wearable, low-cost IMU-based, insensible to sensors placed on the body and real-time system is capable to estimate the upper body joints' angle with an acceptable degree of accuracy for predicting the grasp moment. The learning algorithm is a proof-of-concept in which according to the upper body joint's angle the algorithm can identify the kinematics signature of R2G movement, estimate the progress from-Reach-to-Grasp and predict the grasp moment.

During the experiment, we gathered 3,540 different R2G movements, collected from 10 subjects who participated on two different days and each day had two different grasp tasks – one while sitting and the other while standing. We have shown the variety of the collected data in aspects of timing, objects and positions in Figure 17, Figure 18, and Figure 19. We validated our *R2Gmotion* system with the RUL model, NEXUS™, VICON™. The comparison of two different motion captures systems is not trivial due to the significant dependency of the anatomical joint modeling and the coordination system definition. Hence, the comparison was done as a statistical test according to the Pearson coefficient which indicates the level of linear correlation between two groups. Moreover, since we designated the system for people with transradial amputation we defined different coordination systems not only by rotation but by position with respect to VICON™'s convention and ISB recommendation [30]. As shown in Figure 20, indeed there is no correlation between the pronation-supination acquired by our system and VICON™'s. Nevertheless, Figure 20 emphasizes a higher linear correlation of the angular velocities in comparison to the angles, between our system and VICON™'s. Interesting to note, section 8.1 presents the significance of the angular velocity on the prediction results in comparison to the angles information which might be the results of this finding. To evaluate the quality of the validation result, we compared our system's results with the results of studies on markers-less motion capture solutions. For example, Bonnechère et al [11] compared their Kinect-based system with VICON™'s by evaluating the linear correlation of trajectory parameters calculated by both systems. Although in our study we chose a different set of parameters to compare and in [11] only shoulder and wrist were analyzed, the Pearson coefficients of our system were better for the shoulder's angular

velocity at all direction (0.795, 0.713, 0.630 w.r.t 0.5 at [11]) and the flexion velocity of the wrist (0.727 w.r.t 0.51 at [11]).

The motivation to use a statistical test is the high correlation suggests that a learning algorithm can potentially predict VICON™'s outputs based on our system outputs, which means that hypothetically both systems are capable to achieve similar results at predicting R2G features. Still, it can be assumed that VICON™'s RUL models' estimated joints angles represent more accurately the true biomechanical angles with respect to our low-cost system. Hence, it can be expected that VICON™ - based input data should yield better or similar test results than the IMU-based input data when predicting the R2G stage. If not, our architecture might suffer from overfitting toward a systematic noise in our experimental data and might reduce the robustness of our results. As evident in Table 3, both systems achieve similar results for classifying R2G movement over other movements, while the test loss of predicting the R2G stage is smaller for VICON™'s data rather than *R2Gmotion*'s data, by more than 20%, in accordance with our initial assumption.

In general, the learning algorithm we developed is divided into two parts, first applying classification then solving a regression problem. The classification problem was defined as a binary classification in which the algorithm needs to identify samples acquired during R2G movement from all other movements. The selected RF classifier achieved a 90.24% success rate and specificity of 75.60% when tested on 3 out of 10 subjects [Table 6], while according to the leave-one-out test the median of the classification results is 89.25% sensitivity and 85.28% specificity [Figure 22]. In the second part, after assuming a window of R2G samples has arrived, the regression problem was defined to be the estimation of the progress from-Reach-to-Grasp which was calculated as the normalized time, (i.e., the R2G stage) where 0 represents the reaching moment and 1 is the grasp moment. The selected regression estimator, GRU-based RNN, based on 250[ms] window, achieved MSE of 0.034 and 0.031 according to 3-out-of-10 subjects and leave-one-out, respectively [Figure 21].

In comparison to other suggested systems in the literature, the most similar system we have found is Gardner et al [10], where they used a fused system of MMG, webcam and IMU on the wrist and biceps to control prosthetic hands. The main differences between their system and ours are that their system is not real-time since it based on values calculated during the whole movement; their results were per subject due to generalization problem between subjects (for each subject the net trained again); they used pronation-supination angle which does not always exist after a transradial amputation.

Since we have not found a study that estimates the process of R2G, it is challenging to compare our regression results with others. Moreover, as mentioned before, there is no clear consensus about the R2G normalization, hence naturally it is more difficult to find a study that chooses the same parameters as we did. On the other hand, comparing our classification results with similar

biomechanics upper limb classification problems is a bit simpler. Goutsu et al [31] classified ADLs according to different body parts motion captures by using a camera-based (Kinect™) or marker-based capture system. They achieved an 81.1% classification rate for multiple classes (10-11 classes from different datasets), using the Hidden Markov model (HMM) to separate the human body into parts and classification using Multiple kernel learning (MKL) with SVM [31]. Although [13] achieved a recognition rate of only 75% from 7 classes, they show that each movement has its unique force and moment pattern [13]. Moreover, [35] implemented a portable camera-based system for controlling prosthetic hands. Using the analyzed images from the camera they identify the suitable grasp type for the hand's situation. They achieve a 95.90% accuracy for 6 different grasp types and additional non-grasp labels when the objects were placed at one of the two predefined places, and 88.65% accuracy when they tested their net on objects placed at different places from the training setup. Comparing our results and the presented result here, it can be seen that our results are better than [13]and [31], although they have classified much more groups and gestures than what we needed here. [35] results were better than ours, but they lack the enrich location and orientation of their database hence it is unclear which solution could be generalized better in a real-life application.

Although the classification and regression results comparison might help to understand better the quality of our system results, the main result of our system is the end-product – the identification of the grasp moment and its accuracy for ADLs when using prosthetic hands. Naturally, the feeling of assurance for prosthetic hands users is very important and consequently a false positive, meaning detecting a grasp when the user did not mean to grasp, is very much unwanted. As part of the system requirements, we limited the FPR to be under 2%. To evaluate our end-product, we needed to define a criterion of early-detection which indicates that the prosthetic hand started to close the fingers before the hand has reached the object. This parameter is dependent on the inherent lag between the user's movement and the activation of the hand's motors, on the motor's speed and It is also dependent on the user's preferences. For the demonstration captured in Appendix B, we used 500[ms] as the early-detection criterion which yields: 1.5% FPR, and 92.8% TPR from which 23.8% are early-detection [see Figure 24]. These results can be changed according to the user's preference and hence for better understanding the significance of the results, a future experiment with transradial users is needed to be conducted.

Chapter 10

Conclusions

The main challenge in the prosthetic hand field has been for many years and still today, the user interfaces for controlling the hand. In other words, to translate the user desires and needs into simple commands for the hand's actuators. Same as each one of us chooses differently how he or she makes their breakfast or brushes their teeth, we believe that for each prosthetic hand user, a different control method might be preferable. But all users with an amputation below the elbow reach their arm and approach an object before grabbing it. Although there is a major variety in the R2G movement between different people, different situations, different environments and more, the variance is seemingly similar for able-bodied people and people with under the elbow amputation. Therefore, a significantly larger dataset is available to study and develop a robust predictor algorithm.

The aim of this study is to predict the grasp moment and estimate the progress of the Reach-to-Grasp movement by learning the kinematics of the upper body segment and implement it as a real-time, ready-to-use system. In this thesis, we present our *R2Gmotion* system – a system we developed based on low-cost IMU sensors, with embedded sensor-to-body-segment calibration which requires the user to accomplish only a quick and simple calibration without an accurate sensor placement. In addition, according to a mathematical model the joints' relative kinematics is estimated and a learning algorithm analyzes the incoming data. First, the algorithm distinguishes between R2G movement and all others, then it estimates the R2G stage, finally by applying modified threshold parameters the grasp moment is determined. Here, as parallel research, we designed a low-cost, 3d-printed, robotic hand that can be controlled wirelessly, and can be used as a platform to implement other user control methods very effortlessly.

In conclusion, we proposed a novel concept – utilizing the nature, sure-to-happen reach-to-grasp movement to predict the grasp timing. Except [10] which is described in the Discussion section, to our knowledge, there is no published work that presents controlling prosthetic hand based on R2G movement although this is the first to come movement before almost any grasp as long as the amputation is not above the elbow. Our concept results in a better starting point for prosthetic hand control with zero training-time. This way a more sophisticated fused system, such as a camera-based system can focus on processing data only when the estimated R2G stage is high, hence reducing the control-burden, the required calculation power and diminishing the need to estimate the timing of the

operation. Another added value of the system design is that it is based on low-cost and accessible components to ease the transition into a real-life application for minimal resources.

Appendix A

Kinematics Equations

Extracting angular kinematics from quaternion:

The angular velocity calculation according to a quaternion, based on [28]:

$(\omega_I^B)_I = 2 \cdot \dot{q}_I^B \cdot q_I^{B*}$

The transformation from system I to system B is applied to represent the angular velocity of the distal segment (i.e., system B) w.r.t the proximal segment (i.e. system I) in the distal segment coordinates system, according to quaternion vector rotation operation (v is a general vector):

$v_B = q_B^I \cdot v_I \cdot q_B^{I*}$

We can develop all the derivates accordingly (eliminating the I and B notation for better visualization):

$$\dot{q} = \frac{1}{2}\omega \cdot q \Rightarrow \omega = 2\dot{q} \cdot q^*$$

$$\ddot{q} = \frac{1}{2}(\alpha \cdot q + \omega \cdot \dot{q}) = \frac{1}{2}(\alpha \cdot q + 2\dot{q} \cdot q^* \cdot \dot{q}) \Rightarrow \alpha = (2\ddot{q} - 2\dot{q} \cdot q^* \cdot \dot{q}) \cdot q^*$$

$$\dddot{q} = \tfrac{1}{2}(\dot{\alpha} \cdot q + 2\alpha \cdot \dot{q} + \omega \cdot \ddot{q}) = \tfrac{1}{2}\dot{\alpha} \cdot q + \alpha \cdot \dot{q} + \tfrac{1}{2}\omega \cdot \ddot{q} = \tfrac{1}{2}\dot{\alpha} \cdot q + (2\ddot{q} \cdot q^* - 2\dot{q} \cdot q^* \cdot \dot{q} \cdot q^*) \cdot \dot{q} +$$

$$\tfrac{1}{2}(2\dot{q} \cdot q^*) \cdot \ddot{q} \Rightarrow \dot{\alpha} = 2\dddot{q} \cdot q^* - 4(\ddot{q} \cdot q^* - \dot{q} \cdot q^* \cdot \dot{q} \cdot q^*) \cdot \dot{q} \cdot q^* - 2(\dot{q} \cdot q^*) \cdot \ddot{q} \cdot q^*$$

To obtain the following relations:

$$\begin{cases} \omega = 2\dot{q} \cdot q^* \\ \alpha = (2\ddot{q} - \omega \cdot \dot{q}) \cdot q^* \\ \dot{\alpha} = 2\dddot{q} \cdot q^* - 2\alpha \cdot \dot{q} \cdot q^* - \omega \cdot \ddot{q} \cdot q^* \end{cases}$$

Where the final results can be obtained by extract only the vector part from the 4-element result, since it has been calculated as a quaternion:

$\omega_q = [q_0, q_i, q_j, q_k]^T \Rightarrow \omega = [q_i, q_j, q_k]^T$

And then by applying the transformation to the CS of the rotating system by using:

$\omega_{rot} = q^* \cdot \omega \cdot q$

When the q is the orientation of the rotating system w.r.t the fixed system as defined in the previous equation.

Extracting angular kinematics from Euler angles:

The angular velocity calculation according to Euler angles, based on [28]:

$$[(\omega_I^B)_I]_\times = \dot{R}_I^B \cdot (R_I^B)^T$$

When the notation $[.]_\times$ is the Plücker matrix, such that:

$$[(\omega_i)_i]_\times = \begin{bmatrix} 0 & -\omega_{z,i} & \omega_{y,i} \\ \omega_{z,i} & 0 & -\omega_{x,i} \\ -\omega_{y,i} & \omega_{x,i} & 0 \end{bmatrix} \Rightarrow (\omega_i)_i = \begin{bmatrix} \omega_{x,i} \\ \omega_{y,i} \\ \omega_{z,i} \end{bmatrix}_i$$

We can develop all the derivates accordingly:

$$\dot{R}_I^B = [(\omega_I^B)_I]_\times \cdot R_I^B \Rightarrow [(\omega_I^B)_I]_\times = \dot{R}_I^B \cdot (R_I^B)^T$$

$$\begin{aligned} \ddot{R}_I^B = [(\alpha_I^B)_I]_\times \cdot R_I^B + [(\omega_I^B)_I]_\times \cdot \dot{R}_I^B \Rightarrow [(\alpha_I^B)_I]_\times &= \ddot{R}_I^B (R_I^B)^T - [(\omega_I^B)_I]_\times \cdot \dot{R}_I^B (R_I^B)^T \\ &= \ddot{R}_I^B (R_I^B)^T - ([(\omega_I^B)_I]_\times)^2 \end{aligned}$$

$$\begin{aligned} \dddot{R}_I^B = \frac{\partial}{\partial t}[(\alpha_I^B)_I]_\times \cdot R_I^B + 2[(\alpha_I^B)_I]_\times \cdot \dot{R}_I^B + [(\omega_I^B)_I]_\times \cdot \ddot{R}_I^B \Rightarrow [(\dot{\alpha}_I^B)_I]_\times \\ = \dddot{R}_I^B (R_I^B)^T - 2[(\alpha_I^B)_I]_\times \cdot \dot{R}_I^B (R_I^B)^T - [(\omega_I^B)_I]_\times \cdot \ddot{R}_I^B (R_I^B)^T \end{aligned}$$

Then transformation from system I to system B is applied to represent the angular velocity of the distal segment (i.e., system B) w.r.t the proximal segment (i.e. system I) in the distal segment coordinates system:

$$[(\omega_I^B)_B]_\times = (R_I^B)^T [(\omega_I^B)_I]_\times$$

Where the final results can be obtained by extract the vector part from the Plücker matrix:

$$[\omega]_\times = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix} \Rightarrow \omega = \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix}$$

Appendix B

Video Links

Sensor-to-Segment Calibration process video link:

https://drive.google.com/file/d/1tix7GqBaI9rVHO7oaf5AXhp_oroDEae1/view?usp=sharing

R2Gmotion demonstration video links:

https://drive.google.com/file/d/11V3wMGJXVcw6t8FpyKb3j8RhkM2RVyZF/view?usp=sharing

https://drive.google.com/file/d/1CA-8ORt-dEFpJzi1n1vmBeOHnLQ3_2bZ/view?usp=sharing

Generalization on [15] database video link:

https://drive.google.com/file/d/1n2CGUbxapB5NvDToAEYs-K_YFMmlC0jR/view?usp=sharing

Appendix C

Technical Docs

3d-printed low-cost prosthetic hand project link containing the mechanical and electronic design together with a detailed manual on manufacturing and assembly of the hand:

https://github.com/Haifa3D

The LabVIEW code of the Experiment system and the real-time *R2Gmotion* system:

https://drive.google.com/drive/folders/1fQsS4s9WhMq-bZJMJ0AVE0o-gqXCkc6H?usp=sharing

The Python code including the trained classifier and net:

https://drive.google.com/drive/folders/1C713gwcM5MzHoL6eYgnnSKyJIwg8wW5u?usp=sharing

References

[1] L. Santisteban, M. Térémetz, J. P. Bleton, J. C. Baron, M. A. Maier, and P. G. Lindberg, "Upper limb outcome measures used in stroke rehabilitation studies: A systematic literature review," *PLoS One*, vol. 11, no. 5, pp. 1–16, 2016.

[2] E. E. Butler, A. L. Ladd, S. A. Louie, L. E. LaMont, W. Wong, and J. Rose, "Three-dimensional kinematics of the upper limb during a Reach and Grasp Cycle for children," *Gait Posture*, vol. 32, no. 1, pp. 72–77, 2010.

[3] F. Cordella *et al.*, "Literature review on needs of upper limb prosthesis users," *Front. Neurosci.*, vol. 10, no. MAY, pp. 1–14, 2016.

[4] M. Atzori and H. Müller, "Control Capabilities of Myoelectric Robotic Prostheses by Hand Amputees: A Scientific Research and Market Overview," *Front. Syst. Neurosci.*, vol. 9, no. November, pp. 1–7, 2015.

[5] S. L. Carey, D. J. Lura, and M. J. Highsmith, "Differences in myoelectric and body-powered upper-limb prostheses: Systematic literature review," *J. Rehabil. Res. Dev.*, vol. 52, no. 3, pp. 247–262, 2015.

[6] E. Cho, R. Chen, L.-K. Merhi, Z. Xiao, B. Pousett, and C. Menon, "Force Myography to Control Robotic Upper Extremity Prostheses: A Feasibility Study," *Front. Bioeng. Biotechnol.*, vol. 4, no. March, pp. 1–12, 2016.

[7] Y. Meirovitch, D. Bennequin, and T. Flash, "Geometrical Invariance and Smoothness Maximization for Task-Space Movement Generation," *IEEE Trans. Robot.*, vol. 32, no. 4, pp. 837–853, 2016.

[8] M. Zago, A. Matic, T. Flash, A. Gomez-Marin, and F. Lacquaniti, "The speed-curvature power law of movements: a reappraisal," *Exp. Brain Res.*, vol. 236, no. 1, pp. 69–82, 2018.

[9] N. Hu, A. Bestick, G. Englebienne, R. Bajscy, and B. Kröse, "Human intent forecasting using intrinsic kinematic constraints," in *IEEE International Conference on Intelligent Robots and Systems*, 2016, vol. 2016-Novem, pp. 787–793.

[10] M. Gardner, R. Vaidyanathan, E. Burdet, and B. C. Khoo, "Motion-based grasp selection: Improving traditional control strategies of myoelectric hand prosthesis," *IEEE Int. Conf. Rehabil. Robot.*, vol. 2015-Septe, pp. 307–312, 2015.

[11] B. Bonnechère, V. Sholukha, L. Omelina, S. V. S. Jan, and B. Jansen, "3D analysis of upper limbs motion during rehabilitation exercises using the kinect™ sensor: Development, laboratory validation and clinical application," *Sensors (Switzerland)*, vol. 18, no. 7, pp. 1–18, 2018.

[12] D. G. Liebermann, A. Biess, C. C. A. M. Gielen, and T. Flash, "Intrinsic joint kinematic planning. II: Hand-path predictions based on a Listing's plane constraint," *Exp. Brain Res.*, vol. 171, no. 2, pp. 155–173, 2006.

[13] T. Yabuki and G. Venture, "Human motion classification and recognition using wholebody contact force," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2015, pp. 4251–4256.

[14] G. Sadarangani and C. Menon, "A wearable sensor system for rehabilitation applications," in *IEEE International Conference on Rehabilitation Robotics (ICORR)*, 2015, pp. 672–677.

[15] A. Saudabayev, Z. Rysbek, R. Khassenova, and H. A. Varol, "Human grasping database for activities of daily living with depth, color and kinematic data streams," *Sci. data*, vol. 5, p.

180143, 2018.

[16] M. Han, S. Y. Günay, G. Schirner, T. Padır, and D. Erdoğmuş, “HANDS: a multimodal dataset for modeling toward human grasp intent inference in prosthetic hands,” *Intell. Serv. Robot.*, vol. 13, no. 1, pp. 179–185, 2020.

[17] J. Martinez, M. J. Black, and J. Romero, “On human motion prediction using recurrent neural networks,” in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition(CVPR)*, 2017, pp. 2891–2900.

[18] H. S. Koppula, R. Gupta, and A. Saxena, “Learning human activities and object affordances from RGB-D videos,” *Int. J. Rob. Res.*, vol. 32, no. 8, pp. 951–970, 2013.

[19] P. Muller, M. A. Begin, T. Schauer, and T. Seel, “Alignment-Free, Self-Calibrating Elbow Angles Measurement Using Inertial Sensors,” *IEEE J. Biomed. Heal. Informatics*, vol. 21, no. 2, pp. 312–319, 2017.

[20] L. S. Vargas-Valencia, A. Elias, E. Rocon, T. Bastos-Filho, and A. Frizera, “An IMU-to-body alignment method applied to human gait analysis,” *Sensors (Switzerland)*, vol. 16, no. 12, pp. 1–17, 2016.

[21] J. K. Lee and W. C. Jung, “Quaternion-based local frame alignment between an inertial measurement unit and a motion capture system,” *Sensors (Switzerland)*, vol. 18, no. 11, 2018.

[22] M. M. B Morrow *et al.*, “Validation of Inertial Measurement Units for Upper Body Kinematics HHS Public Access Author manuscript,” *J Appl Biomech*, vol. 33, no. 3, pp. 227–232, 2017.

[23] R. Woodward, S. Shefelbine, and R. Vaidyanathan, “Integrated grip switching and grasp control for prosthetic hands using fused inertial and mechanomyography measurement,” *2015 Swarm/Human Blended Intell. Work. SHBI 2015*, pp. 1–8, 2015.

[24] A. J. Young, T. A. Kuiken, and L. J. Hargrove, “Analysis of using EMG and mechanical sensors to enhance intent recognition in powered lower limb prostheses,” *J. Neural Eng.*, vol. 11, no. 5, 2014.

[25] M. T. Wolf, C. Assad, M. T. Vernacchia, J. Fromm, and H. L. Jethani, “Gesture-Based Robot Control with Variable Autonomy from the JPL BioSleeve,” *IEEE Int. Conf. Robot. Autom.*, pp. 1160–1165, 2013.

[26] E. Repnik, U. Puh, N. Goljar, M. Munih, and M. Mihelj, “Using Inertial Measurement Units and Electromyography to Quantify Movement during Action Research Arm Test Execution,” *Sensors*, vol. 18, no. 2767, pp. 1–23, 2018.

[27] S. Alavi, D. Arsenault, and A. Whitehead, “Quaternion-Based Gesture Recognition Using Wireless Wearable Motion Capture Sensors,” *Sensors*, vol. 16, no. 5, 2016.

[28] Henryk Flashner and Jill L. McNitt-Gray, “3D Kinematics: Using Quaternions for Modeling Orientation and Rotations in Biomechanics,” in *Biomechanical Principles and Applications in Sports*, 2019, pp. 155–234.

[29] C. Gramkow, “On averaging rotations,” *J. Math. Imaging Vis.*, vol. 15, no. 1–2, pp. 7–16, 2001.

[30] G. Wu *et al.*, “ISB recommendation on definitions of joint coordinate systems of various joints for the reporting of human joint motion - Part II: Shoulder, elbow, wrist and hand,” *J. Biomech.*, vol. 38, no. 5, pp. 981–992, 2005.

[31] Y. Goutsu, W. Takano, and Y. Nakamura, “Classification of Multi-class Daily Human Motion using Discriminative Body Parts and Sentence Descriptions,” *Int. J. Comput. Vis.*, vol. 126, no. 5, pp. 495–514, 2018.

[32] B. Sensortec, “BNO055: Intelligent 9-axis absolute orientation sensor,” 2016.

[33] U. L. Model, “Vicon Upper Limb Model:Product Guide,” 2007.

[34] M. Schepers and M. Giuberti, “Xsens MVN : Consistent Tracking of Human Motion Using

Inertial Sensing," 2018.

[35] L. T. Taverne, M. Cognolato, T. Butzer, R. Gassert, and O. Hilliges, "Video-based Prediction of Hand-grasp Preshaping with Application to Prosthesis Control," in *International Conference on Robotics and Automation (ICRA)*, 2019, pp. 4975–4982.

תקציר

גוף האדם הוא מערכת יעילה מאוד המורכבת מאינסוף רכיבים שונים העובדים יחד בתיאום וסינרגיה מעוררים השראה. רתימת עקרון זה עבור התנועה הטבעית של הושטת-יד-לשם-אחיזה היא אבן הפינה של עבודה זו לקראת פיתוח יד תותבת חכמה יותר שיכולה ללמוד את דפוסי תנועת הושטת-יד-לשם-אחיזה וכך לפעול באופן חלק ודינמי יותר. בהשוואה למחקרים קודמים, כאן רכשנו ופענחנו את תנועת הושטת-יד-לשם-אחיזה בהתחשב בתלות הזמנית, במקום לסווג מספר מחוות מוגדרות מראש שמשתמשים נדרשים ללמוד בעל פה, כפי שרוב אסטרטגיות הבקרה מבוססות על. ההשערה שלנו היא שניתן לחזות את כוונת המשתמש לאחוז באובייקט וניתן להבדיל בין תנועת הושטת-יד-לשם-אחיזה לבין תנועות ידיים אחרות אצל אנשים עם קטיעת גפיים מתחת למרפק, באמצעות ניתוח הקינמטיקה של פלג הגוף העליון, ובזכות המורכבות והסינרגיה של תנועת הושטת-יד-לשם-אחיזה. מכיוון שהמטרה הסופית היא לשלוט בידיים תותבות, הנחנו שאין לנו נתונים קינמטיים עבור תנועות מתחת למרפק. למטרה זו פיתחנו מערכת מבוססת חיישני תנועה בעלות נמוכה הרוכשת את הקינמטיקה של הזרוע ופלג גוף עליון, לא כולל פרונציה-סופינציה של כף היד. ביצענו ניסוי בו השתתפו עשרה משתתפים בריאים ימניים, ביומיים שונים, בו השתמשנו בשתי מערכות - המערכת שלנו, ובמערכת אופטו-אלקטרונית מקובלת בספרות לאימות. כל משתתף ביצע משימות שונות של הושטת-יד-לשם-אחיזה עבור אחיזת אובייקטים שונים, ממקומות שונים ואקראיים, ובזמנים אקראיים על פי הוראות חזותיות וקוליות המנוהלות על ידי האפליקציה שבנינו, תוך כדי ישיבה ועמידה (סה"כ כ3.5 אלף תנועות של הושטת-יד-לשם-אחיזה). כמו כן, המידע הקינמטי של המשתתפים נרכש תוך כדי הליכה ודיבור בשביל לחקות תנועות יד אחרות שאינן הושטת-יד-לשם-אחיזה, כדי לאשרר ולבדוק את היכולות של האלגוריתם שלנו כאשר אנו מבדילים בין תנועת הושטת-יד-לשם-אחיזה ממשימות אחרות. ניתוח הנתונים נעשה על ידי השוואה בין סוגים שונים של ארכיטקטורות למידת מכונה לבין סוגי נתוני קלט שונים כגון דוגמאות מגוונות של חלונות זמן ומאפיינים קינמטיים שונים. יתר על כן, הוחלו כמה שיטות נורמליזציה וממוצע כדי לחלץ מאפיינים משמעותיים שעשויים לשפוך אור על הביומכניקה של תנועת הושטת-יד-לשם-אחיזה. התוצאות מראות שדיוק המערכת שלנו בזמן אמת עבור אומדן הקינמטיקה של מפרקי פלג הגוף העליון טובה מספיק בכדי לחזות את רגע האחיזה. מודל למידת המכונה שנבחר הוא שילוב של מסווג המשערך אם אכן מתרחשת כרגע תנועת הושטת-יד-לשם-אחיזה (חציון תוצאות: רגישות 89.25% וספציפיות של 85.28%), ומשערך רגרסיה המעריך את השלב הנוכחי בתהליך תנועת הושטת-יד-לשם-אחיזה (תוצאות חציוניות: שגיאה ריבועית ממוצעת של 0.031 עבור טווח פלט של [0,1]). על פי תוצאות המודל והפרמטרים המותאמים אישית, המערכת מזהה את רגע האחיזה (עבור פרמטרים נבחרים: 1.5% חיובי כוזב, ו 92.8% חיובי נכון ממנו 23.8% נחשבים לגילוי מוקדם) ומפעילה את היד התותבת המודפסת תלת ממדית שפיתחנו, עבור ביצוע הפעולה. לסיכום, אנו מציגים כאן תפיסת בקרה חדשה לשליטה בידיים תותבות הרותמת את הקינמטיקה של תנועת הושטת-יד-לשם-אחיזה אשר מתרחשת באופן טבעי לפני כל אחיזה. רעיון זה יכול להעניק נקודות פתיחה טובה יותר עבור מערכות לשליטה ביד תותבת הכוללות חיישנים רבים, בכך שממקד את הזמן בו המערכת צריכה לבצע את ניתוח המידע לפני כל אחיזה. מערכת ה-R2Gmotion שפיתחנו, הינה מערכת לבישה, הפועלת בזמן אמת, המבוססת על רכיבים זולים ונגישים, והמוכנה להטמעה, יכולה לשמש כמערכת בקרה עצמאית או כפלטפורמה שניתן ליישם על גביה שיטות בקרה נוספות ובאמצעות כך לבנות מערכת בקרה משולבת שהינה טובה, ואמינה עבור ידיים תותבות.

המחקר בוצע בהנחייתו של פרופסור אלון וולף, בפקולטה להנדסת מכונות.

תודות

אני מודה לטכניון על התמיכה הכספית הנדיבה בהשתלמותי.

גישת בקרה מבוססת חיישני תנועה לשליטה על ידיים תותבות באמצעות למידת הקינמטיקה של תנועת הושטת היד לקראת אחיזה

חיבור על מחקר

לשם מילוי חלקי של הדרישות לקבלת התואר

מגיסטר למדעים

שונית פולינסקי

הוגש לסנט הטכניון — מכון טכנולוגי לישראל

ניסן התשפ"א חיפה מרץ 2021

גישת בקרה מבוססת חיישני תנועה לשליטה על ידיים תותבות באמצעות למידת הקינמטיקה של תנועת הושטת היד לקראת אחיזה

שונית פולינסקי