Syrian Arab Republic

Ministry of Higher Education and Scientific Research

Manara University-Faculty of Engineering

Robotic and Intelligent Systems Department

Design and Dynamic Modeling of a Hand Exoskeleton Assisted with Hand Pose Estimation for Rehabilitation

Prepared by

Alaa Hassoun

Aya Sadek

Tala Khaddour

Zein Sulieman

Supervised by

PhD. Nael Daoud

2023-2024

ملخص

أصبحت الهياكل الخارجية لليد نقطة محورية في مجال تكنولوجيا إعادة التأهيل. توفر الهياكل مساعدة و تعزيز لاستعادة القدرات الوظيفية لليد البشرية، مما يؤدي إلى تحسينات كبيرة في نوعية حياة المرضى.

في هذا المشروع، تم دراسة الأبحاث السابقة المعنية بالهياكل الخارجية لليد من أجل تحديد المتطلبات والتكوين المناسب للتصميم. يتكون التصميم المقترح من حلقتين مغلقتين لأداء حركات ثنائية الاتجاه لدرجات الحرية الثلاث: الثني والتمديد في المفصل السنعي السلامي (MCP) والمفاصل السلامية القريبة (PIP)؛ والاختطاف/التقريب في مفصل MCP. تم استخلاص الدراسة الكينماتيكية والديناميكية لكلا النموذجين. استنادًا إلى الدراسة الحركية والديناميكية، تم استخدام مؤشرات أداء مختلفة في عملية تحسين معاملات الأبعاد لتصميم الهيكل الخارجي لليد. تم عرض نتائج التحسين ونموذج CAD الذي تم تطويره للهيكل الخارجي.

تمت دراسة الشبكات متعددة الطبقات و الشيكات العصبية الالتفافية، جنبًا إلى جنب مع مجموعة من الشبكات الكلاسيكية لتطبيق تقدير لوضعية اليد. أجريت دراسة مقارنة لخوارزميات تقدير وضعية اليد ثنائية الأبعاد من حيث الدقة والسرعة. تضمنت هذه الدراسة خوارزميات آلات الوضعية التلافيفية (CPM)، والساعة الرملية المكدسة (SHG)، وتقدير الوضعية المتعددة النماذج في الوقت الفعلي (RTMPose)، وMediapipe، والشبكات الخفيفة عالية الدقة (LiteHR Nets). أظهرت النتائج أن RTMPose كان لديه أفضل حل وسط بين الدقة والسرعة في مجموعات البيانات المعيارية.

Abstract

Hand exoskeletons have become a focal point in the realm of rehabilitation technology. They offer to assist, enhance, or restore the functional capabilities of the human hand, leading to significant improvements in the quality of life for patients.

In this project, prior studies on hand exoskeletons were investigated in order to set the requirements and the proper configuration of the design. The proposed design introduces two closed-loop chains to perform bidirectional movements of the three degrees of freedom: flexion and extension in metacarpophalangeal (MCP) joint and proximal interphalangeal (PIP) joint; and abduction/adduction in MCP joint. The kinematics and dynamics analysis of both chains were derived. Based on the kinematics and dynamics manipulation, different performance indices were used in the optimization process of the dimensional parameters of the hand exoskeleton design. The results of the optimization and the CAD model developed for the exoskeleton were shown.

MultiLayer perceptron neural networks and Convolutional neural networks were studied along with a set of classic nets for the application of hand pose estimation. A comparison study of 2D hand pose estimation algorithms was conducted according to the accuracy and speed. This study included convolutional pose machines (CPM), stacked hourglass (SHG), real-time multi-model pose estimation (RTMPose), Mediapipe, and lite high-resolution nets (LiteHR Nets). The results have shown that RTMPose had the best compromise between accuracy and speed on benchmark datasets.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS

CNN Convolutional **N**eural **N**etwork

CPM Convolutional **P**ose **M**achines

DIP Distal **I**nter-**P**halangeal

HES Hand **E**xo-**S**keleton

LiteHR Lite High **R**esolution

MCP Meta-**C**arpo-**P**halangeal

P Prismatic joint

PIP Proximal **I**nter-**P**halangeal

R Revolute joint

ROM Range **o**f **M**otion

RTMPose Real **T**ime **M**ulti **Pose**

1 Introduction

Hand exoskeletons have become a focal point in the realm of rehabilitation technology, particularly for individuals who have suffered from hand impairments due to conditions such as stroke, spinal cord injuries, cerebral palsy, and other neurological disorders. Hand exoskeletons offer to assist, enhance, or restore the functional capabilities of the human hand, leading to significant improvements in the quality of life for patients and facilitating their reintegration into daily activities.

Hand disabilities are a major concern worldwide. According to the World Health Organization (WHO), approximately 15 million people suffer from strokes annually, with about 5 million of these individuals experiencing permanent disabilities, including impaired hand function. Additionally, spinal cord injuries affect between 250,000 and 500,000 people globally each year, many of whom require extensive rehabilitation to regain hand mobility and dexterity. Traditional rehabilitation methods, though effective, often require prolonged one-on-one sessions with therapists, making them both time-consuming and costly. Hand exoskeletons offer a viable solution to these challenges by providing consistent, repetitive, and intensive therapy that is crucial for neural plasticity and functional recovery.

The research regarding hand exoskeletons have been heavily focused on the mechanical designing, control and manipulation of the devices. This project proposes a method for incorporating AI into hand exoskeletons, to create more adaptive, responsive, and effective rehabilitation tools. Following

The structure of this project consists of two main parts. First, designing the hand exoskeleton according to rehabilitation requirements and standards stated in prior studies. Furthermore, simulation and experiments are conducted. Second, the AI part which is also subdivided into smaller tasks. The first task in this part is Hand Pose Estimation to find key-points of the hand which correspond to joints. The next mission is training a forecasting model that takes two inputs. These inputs are the current and past key-points’ coordinates alongside the current and past EMG sensors’ outputs. The model returns the next coordinates after a specified period of time according to the EMG sensors. Such a forecasting model plays a role in assisting the development of patients in rehabilitation and physical treatment.

1.1. Research thesis

Researching the effects of incorporating a forecasting model for predicting the hand key-points in the manipulation of the hand exoskeleton when performing complex movements.

1.2. Aims of project

For this part of the project, the aims are:

-Building a cohesive study on previous works and research with hand exoskeletons.

-Defining the design configuration and deriving the kinematics and dynamic model for the proposed mechanism.

- Building a cohesive study on the common architectures used in hand pose estimation.

-Developing the CAD model for the hand exoskeleton.

-Analyzing the performance of these architectures on testing datasets.

2 Hand Exoskeletons in the literature

Hand exoskeleton are wearable robotic devices with functions of assisting the fingers, of the human hand, to complete its range of motion (ROM), amplify power, and rehabilitating impaired hands. The traditional methods of rehabilitation involve the sessions being heavily carried by the physical therapist. Hence the idea of including robotic systems that can assist the therapy process by taking over the physical load of therapists. This also can result in longer training periods and better repetitive exercises that can be executed for a more rapid recovery. Researchers in [1] have shown that the active involvement of the patient can improve their motor learning their cognitive–motor skills, which can lead to a more rapid recovery. Exoskeleton can be classified into assistive and rehabilitation. Regarding rehabilitation, HES can either provides passive movement or the resistive force against the patient’s active movement. Some work developed to work in both modes. Hand exoskeleton designs in literature can be classified according to various criteria. For this literature review, only the criteria relating directly to the functionality and mechanical design are discussed.

Rigid links and soft links: Hand exoskeletons can be designed using either rigid links, which provide more precise control and strength, or soft links, which offer better comfort and adaptability to the user's hand. [2].

Fully actuated and under-actuated: Fully actuated exoskeletons have individual actuators for each joint, allowing for independent control and complex movements, whereas under-actuated exoskeletons have fewer actuators than degrees of freedom, relying on mechanical linkages and the natural motion of the hand to achieve movement.

Intention sensing method: The methods mainly include three types: force sensing using force sensing resistors (FSRs), pneumatic pressure sensors, or strain gauge sensors. Surface Electromyography (sEMG) can also be employed to detect the electrical activity produced by skeletal muscles [2].

Power transmission: Power transmission mechanisms can include cables, gears, pulleys, or linkages. These systems are responsible for transferring the power from the actuators to the joints of the exoskeleton, enabling the intended movements and ensuring that the force is applied accurately and efficiently.

2.1. Design Requirements

The first step for designing the hand exoskeleton is setting the requirements of the design. In this project, the requirements are chosen aligning with what have been discussed in previous study cases [2] [3] [4], relating to both the functionality or the rehabilitation experience:

- Hand Anatomy: A hand exoskeleton must align with the hand's anatomy, the natural range of motion and the hand joints’ constraints, to ensure proper functionality and movement. This will be discussed in depth when introducing the hand model.

- High output forces: patients with hand disabilities suffer from high stiffness along their joints, so rotating their finger joints requires efficient output forces.

- Effective Force Transmission: Exoskeletons must transmit actuator forces naturally to the user's fingers. Torques around finger joints should be balanced to prevent discomfort, and forces between the exoskeleton and finger phalanges should be perpendicular to avoid slippage.

2.2. Selection of mechanical design

In this research we investigated different designs for the HES finger mechanical model, the kinematics of these design were studied.

2.2.1. HANDEXOS

The novel design by [3] is composed of five separately actuated fingers, each provided with three active rotational joints, one passive rotational joint, and one passive translational joint. The design uses an under-actuated configuration, where motion of each finger is carried by one motor and six connected pulleys, two for the actuation of each active joint placed along both sides of the HANDEXOS.

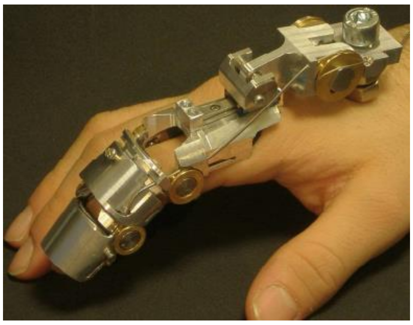


Figure 2.1 The cable under-actuated configuration of HANDEXOS.

2.2.2. Sarac et al.

Authors in [4] proposed a design for an underactuated hand exoskeleton to assist patient to grasp objects with different sizes without any mechanical adjustments. For selecting links parameter, an optimization based on an extensive search procedure has been conducted.

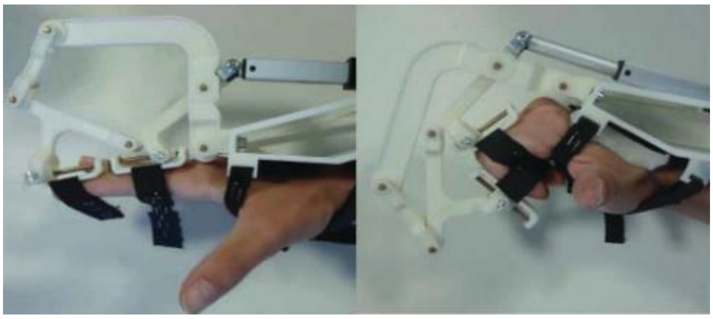


Figure 2.2 The exoskeleton design proposed by Sarac et al.

2.2.3. Wege et al.

This design developed by [5], provides actuation in the bidirectional movement for the four degrees of freedom of joints via a Bowden cable driven by an electric motor. Moreover, the author suggested the sliding mode control to deal with the design parameters variations.

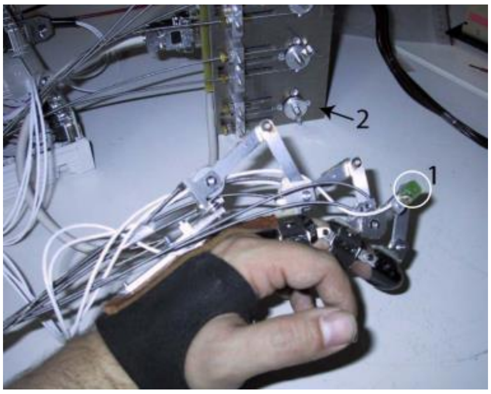


Figure 2.3 The full-actuated hand exoskeleton proposed by Wege et al.

2.2.4. iHandRehab

This design by [6] provides an actuation all four degrees of freedom for each finger through a cable/sheath transmission mechanism. The design defines an exact angular displacement for both the finger's joints and the device's joints by the use of a parallelogram mechanism.

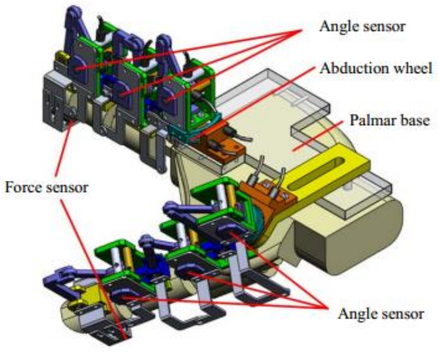


Figure 2.4 The CAD model of iHandRhab.

2.2.5. Discussion

Studying the previous cases, it is determined that the device should be lightweight, and easy to attach to the different sizes of patients' hands. The design should allow the hand's touch sense interaction with the environment. Each finger should be able to perform bidirectional movements of the three degrees of freedom: Flexion and extension in metacarpophalangeal (MCP) joint and proximal interphalangeal (PIP) joint; and abduction/adduction in MCP joint. Lastly, link lengths must allow nearly full range of motion. A four-bar mechanism is chosen to control each joint of the index and thumb fingers. This design addresses the requirement of simplicity and efficiency.

3 Mathematical Background

3.1. Human Hand Modeling

Hand exoskeletons are closely coupled with the human hand, therefore studying the human hand anatomy in a sufficient way is important to design a hand exoskeleton that is compatible with the hand’s DOFs (degrees of freedom) and ROM (range of motion). The human hand is composed of 5 digits: 4 fingers (index, ring, middle,and little fingers) which are composed of three different articulations and four phalanges, and the thumb which is characterized by three articulations and three phalanges. Names and acronyms of each articulation and phalanx that may be used later in the document are illustrated in Figure 3.1 [7].

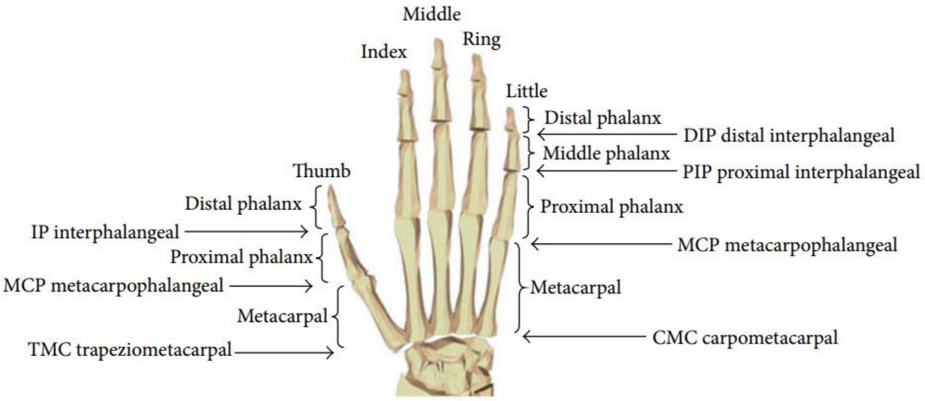


Figure 3.1 The structure of the human hand.

3.1.1. Range and constraints of finger motion

Generating the natural movements for the human hand requires introducing the constraints of fingers motion. There are three types for these constraints: Static Constraints. Applying these constraints can reduce the complexity of solving the inverse kinematics problem. The constraints provided in study [8] were used in this project, they can be viewed in Table 3.1.

Table 3.1 Natural movements range for the human hand.

Finger	flexion	Extension	Abduction/adduction
Thumb			
MCP	75°-80°	0°	5°
IP	75°-80°	5°-10°	5°
Index			
MCP	90°	30 °-40°	60°
PIP	110°	0°	0°
DIP	80°-90°	5°	0°
Middle			
MCP	90°	30 °-40°	45°
PIP	110°	0°	0°
DIP	80°-90°	5°	0°
Ring			
MCP	90°	30°-40°	45°
PIP	120°	0°	0°
DIP	80°-90°	5°	0°
Little			
MCP	90°	30°-40°	50°
PIP	135°	0°	0°
DIP	90°	5°	0°

3.1.2. Intra-finger Constraints

Intra-finger constraints are the constraints between the different joints in the same finger. Study cases [8] presented these different constraints, the following equation defines the relation between the joints of the fingers excluding the thumb:

$$\theta_{DIP} \approx \frac{2}{3}\theta_{PIP}\theta_{PIP} \approx \frac{3}{4}\theta_{MCP\ flex} \qquad \textbf{(3.1)}$$

The following equation defines the constraint relations between the joints when flexing or extending the thumb:

$$\theta_{IP} \approx \frac{1}{2}\theta_{MCP\ f/e} \qquad \textbf{(3.2)}$$

$$\theta_{MCP\ f/e} = \frac{5}{4}\theta_{TMC\ f/e} \qquad \textbf{(3.3)}$$

3.1.3. Interfinger Constraints

Interfinger constraints correlate two joints belonging to different fingers. The respective angles of these joints vary more or less proportionally, unless a voluntary counter force is applied. However, imposing forces in an unnatural way often results in oddly fatiguing efforts, so the following constraints must be respected:

$$\theta_{MCP\ f/e\ Middle} \approx \theta_{MCP\ \ f/e\ Ring} \qquad \textbf{(3.4)}$$

$$\theta_{MCP\ f/e\, Middle} \approx \theta_{MCP\ f/e\, Little} \tag{3.5}$$

$$\theta_{MCP\ ab/ad\, Ring} \approx \theta_{MCP\ ab/ad\, Little} \tag{3.6}$$

Some joints can be constrained by more intricate relations. The following equations indicate that when there is flexion, the ring and middle fingers will also flex in a certain range with respect to the little finger.

$$\theta_{MCP\ f/e\, Ring} \approx \frac{7}{12}\theta_{MCP\ f/e\, Little}, \theta_{MCP\ f/e\, Middle} \approx \frac{2}{3}\theta_{MCP\ f/e\, Ring},$$

$$\theta_{MCP\ f/e\, Ring} - \theta_{MCP\ f/e\, Middle} < 60°,$$

$$\theta_{MCP\ f/e\, Little} - \theta_{MCP\ f/e\, Ring} < 50° \tag{3.7}$$

3.1.4. Kinematics Analysis of human fingers

In this project, two kinematic configurations are considered, one for the thumb and the other for the rest of fingers. The thumb is modeled as 3 links and 4 joints and another for the other fingers (index, middle, ring, and little fingers). The fingers exept for the thumb as a three-links planar manipulator. Studying the hand anatomy, metacarpophalangeal joint (MCP) is modeled by a 2 DOF universal joint whereas proximal interphalangeal (PIP) and distal interphalangeal (DIP) have 1 DOF.

3.1.4.1. Direct Kinematics of the index, middle, ring, and little fingers

Direct kinematics are developed using Denavit-Hartenberg tables which can be illustrated below. Each articulation presented for these four fingers, corresponds to the joints: MCP, PIP, and DIP. The derived model can describe all the adduction/abduction and flexion/extension movements carried by the joints [7].

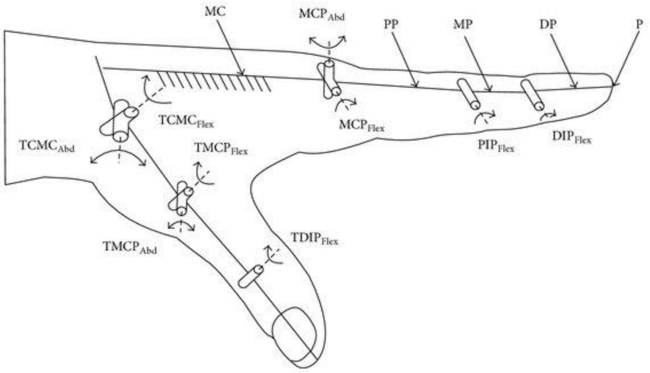


Figure 3.2 Kinematic chain of the index and thumb fingers.

Table 3.2 Denavit-hartenberg table for the index, middle, ring and little fingers.

Joint	a_i	α_i	d	θ_i

1	0	$\frac{\pi}{2}$	0	$\theta_{MCP\ Abd}$
2	L_{PP}	0	0	$\theta_{MCP\ flex}$
3	L_{MP}	0	0	θ_{PIP}
4	L_{DP}	0	0	θ_{DIP}

The transformation matrix is derived as follows:

$$T_0^E = \prod_{j=0}^{3} T_j^{j+1}\left(\theta_j\right) \quad \textbf{(3.8)}$$

Where T_0^E represents the position and orientation of the fingertip with respect to the base frame. For the purpose of simplifying the expressions of the transformation matrix and those equations that will be mentioned later the following notations are used:

$$\theta_{MCP\ Abd} = \theta_1,\ \theta_{MCP\ flex} = \theta_2,\ \theta_{PIP} = \theta_3,\ \theta_{DIP} = \theta_4, \cos(\theta_i) = c_i, \sin(\theta_i) = s_i,\ \cos\left(\theta_i + \theta_j\right) = c_{ij},\ \sin\left(\theta_i + \theta_j\right) = s_{ij} \quad \textbf{(3.9)}$$

The transformation matrix is deduced as (3.10):

$$T_0^E = \begin{bmatrix} c_{234}c_1 & -s_{234}c_1 & s_1 & c_1(L_{PP}c_2 + L_{MP}c_{23} + L_{DP}c_{234}) \\ c_{234}s_1 & -s_{234}s_1 & -c_1 & s_1(L_{PP}c_2 + L_{MP}c_{23} + L_{DP}c_{234}) \\ s_{234} & c_{234} & 0 & L_{PP}s_2 + L_{MP}s_{23} + L_{DP}s_{234} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \textbf{(3.10)}$$

3.1.4.2. Direct kinematics for the thumb:

Table 3.3 Denavit-hartenberg table for the thumb.

Joint	a_i	α_i	d	θ_i
1	0	$\frac{\pi}{2}$	0	$\theta_{TMC/\ Abd}$
2	L_{TMC}	0	0	$\theta_{TMC/flex}$
3	L_{MCP}	0	0	θ_{MCP}
4	L_{IP}	0	0	θ_{IP}

A similar notation (3.11) to the previous configuration is used to simplify equations:

$$\theta_{TMC/\ Abd} = \theta_1, \theta_{TMC/\ flex} = \theta_2, \theta_{MCP} = \theta_3, \theta_{IP} = \theta_4 \quad \textbf{(3.11)}$$

This yields a similar transformation matrix (3.12):

$$T_0^E = \begin{bmatrix} c_{234}c_1 & -s_{234}c_1 & s_1 & c_1(L_{TMC}c_2 + L_{MCP}c_{23} + L_{IP}c_{234}) \\ c_{234}s_1 & -s_{234}s_1 & -c_1 & s_1(L_{TMC}c_2 + L_{MCP}c_{23} + L_{IP}c_{234}) \\ s_{234} & c_{234} & 0 & L_{TMC}s_2 + L_{MCP}s_{23} + L_{IP}s_{234} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \textbf{(3.12)}$$

3.1.5. Inverse Kinematics

The kinematic model described above is sufficient for the description of flexions, extensions, abductions, adductions movements of fingers. Solution of the inverse kinematics can be derived from a geometric method such as triangles' relations. The duplication of solutions, when solving an arctangent, for example, is eliminated because the ranges of all physiological angles are smaller than 180° and most of them are smaller than 90°. It is assumed that the end effector position (P_x, P_y, P_z) and orientation angles (α, β, γ) are given. For the following equations the notation R_{ij} is used for the entry at ith row and jth column for the orientation matrix of R.

$$\theta_{MCP\ Abd} = \text{atan}\left(\frac{P_y}{P_z}\right) \quad \textbf{(3.13)}$$

Then defining an intermediate value:

$$r = \left(P_x c_1 + P_y s_1 - L_{dp} R_{31}\right)^2 + \left(P_z - L_{dp} R_{31}\right)^2 \quad \textbf{(3.14)}$$

The angle of the PIP, DIP and MCP (flexion) joints can be obtained then:

$$\theta_{PIP} = \text{acos}(\frac{r-(L_{PP})^2-(L_{MP})^2}{2L_{PP}L_{MP}}) \quad \textbf{(3.15)}$$

Defining three intermediate variables:

$$c = P_x^2 + P_y^2 + P_z^2 - L_{pp}^2 - L_{mp}^2 - L_{dp}^2 - 2L_{pp}L_{mp}\cos\left(\theta_{PIP}\right) \quad \textbf{(3.16)}$$

$$a = 2L_{pp}L_{dp}\cos(\theta_{PIP}) + 2L_{mp}L_{dp} \quad \textbf{(3.17)}$$

$$b = -2L_{pp}L_{dp}\sin\left(\theta_{PIP}\right) \textbf{(3.18)}$$

$$\theta_{DIP} = \tan\left(\frac{c}{\sqrt{a^2+b^2-c^2}}\right) - \text{atan}\left(\frac{a}{b}\right) \quad \textbf{(3.19)}$$

$$\theta_{MCP\ flex} = \text{atan}\left(\frac{R_{31}}{R_{32}}\right) - \theta_{PIP} - \theta_{DIP} \quad \textbf{(3.20)}$$

3.1.6. Dynamic model of the human hand

The dynamic model for a single finger is derived using Euler-Lagrange equations basing on the study of a 4-Dof serial manipulator. The position vector for the center of mass of the jth phalanx with respect to the jth reference frame is represented as:

$$\overrightarrow{G_J} = \begin{bmatrix} d_{jx} \\ d_{jx} \\ d_{jx} \end{bmatrix} \quad \textbf{(3.21)}$$

From this the position vectors for the centers of mass for the phalanges with respect to the base frame located at the center of the MCP joint, are:

$$\overrightarrow{P_{g1}} = \begin{bmatrix} d_{1x}c_1c_2 - d_{1y}c_1s_2 - d_{1z}s_1 \\ d_{1x}s_1c_2 - d_{1y}s_1s_2 + d_{1z}c_1 \\ -d_{1x}s_2 - d_{1y}c_2 \end{bmatrix} \textbf{(3.22)}$$

$$\overrightarrow{P_{g2}} = \begin{bmatrix} L_{PP}c_2c_1 \\ L_{PP}c_2s_1 \\ -L_{PP}s_2 \end{bmatrix} + \begin{bmatrix} d_{1x}c_1c_{23} - d_{1y}c_1s_{23} - d_{1z}s_1 \\ d_{1x}s_1c_{23} - d_{1y}s_1s_{23} + d_{1z}c_1 \\ -d_{1x}s_{23} - d_{1y}c_{23} \end{bmatrix} \quad \textbf{(3.23)}$$

$$\overrightarrow{P_{g3}} = \begin{bmatrix} L_{PP}c_2c_1 \\ L_{PP}c_2s_1 \\ -L_{PP}s_2 \end{bmatrix} + \begin{bmatrix} L_{MP}c_{23}c_1 \\ L_{MP}s_{23}s_1 \\ -L_{MP}s_{23} \end{bmatrix} + \begin{bmatrix} d_{1x}c_1c_{234} - d_{1y}c_1s_{234} - d_{1z}s_1 \\ d_{1x}s_1c_{234} - d_{1y}s_1s_{234} + d_{1z}c_1 \\ -d_{1x}s_{234} - d_{1y}c_{234} \end{bmatrix} \quad \textbf{(3.24)}$$

The velocity vectors are obtained by differentiating the position vectors with respect to time.

$$\overrightarrow{v_{g1}} = \begin{bmatrix} d_{1x}\left(-s_1c_2\dot{\theta}_1 - c_1s_2\dot{\theta}_2\right) - d_{1y}(-s_1s_2\dot{\theta}_1 + c_1c_2\dot{\theta}_2) - d_{1z}c_1\dot{\theta}_1 \\ d_{1x}\left(c_1c_2\dot{\theta}_1 - s_1s_2\dot{\theta}_2\right) - d_{1y}\left(c_1s_2\dot{\theta}_1 + s_1c_2\dot{\theta}_2\right) - d_{1z}s_1\dot{\theta}_1 \\ -d_{1x}c_2\dot{\theta}_2 + d_{1y}s_2\dot{\theta}_2 \end{bmatrix} \quad \textbf{(3.25)}$$

The rest of the velocity vectors can be obtained similarly. The rotational velocity of the center of masses can be obtained from following equations:

$$\overrightarrow{\omega_{g1}} = \begin{bmatrix} \dot{\theta}_2s_1 \\ -\dot{\theta}_2c_1 \\ \dot{\theta}_1 \end{bmatrix} \quad \textbf{(3.26)}$$

$$\overrightarrow{\omega_{g2}} = \begin{bmatrix} s_1(\dot{\theta}_2 + \dot{\theta}_3) \\ -c_1(\dot{\theta}_2 + \dot{\theta}_3) \\ \dot{\theta}_1 \end{bmatrix} \quad \textbf{(3.27)}$$

$$\overrightarrow{\omega_{g3}} = \begin{bmatrix} s_1(\dot{\theta}_2 + \dot{\theta}_3 + \dot{\theta}_4) \\ -c_1(\dot{\theta}_2 + \dot{\theta}_3 + \dot{\theta}_4) \\ \dot{\theta}_1 \end{bmatrix} \quad \textbf{(3.28)}$$

The kinetic energy with respect to the base reference frame can be obtained as follows:

$$K = \frac{1}{2}\sum_{i=1}^{4} m_i\left(v_{gi}\right)^2 + \frac{1}{2}\sum_{i=1}^{4} I_i\left(\omega_{gi}\right)^2 \quad \textbf{(3.29)}$$

Where: m_i and I_i are the mass and the inertia of the ith link respectively.

The gravitational potential energy for each phalange can be obtained as:

$$P_g = g.m_1\left(d_{1\mathrm{y}}c_2 + d_{1\mathrm{x}}s_2\right) + g.m_2\left(d_{2\mathrm{y}}c_{23} + d_{2\mathrm{x}}s_{23} + L_{\mathrm{pp}}s_2\right) + g.m_3\left(L_{\mathrm{mp}}s_{23} + L_{\mathrm{pp}}s_2 + d_{3\mathrm{y}}c_{234} + d_{3\mathrm{x}}s_{234}\right) \quad \textbf{(3.30)}$$

For a precise of the hand dynamics, the elastic potential energy should be included when modeling the finger as:

$$P_e = \frac{1}{2}\sum_{i=1}^{4} k_i(\theta_i)^2 \quad \textbf{(3.31)}$$

Where: $k_1, k_2, k_3$are the values of stiffness for the MCP, PIP, and DIP joints respectively.

Compensating for the dumping forces, the Rayleigh dissipation function F is used:

$$F = \frac{1}{2}\sum_{i=1}^{4} c_i\left(\dot{\theta_i}\right)^2 \textbf{ (3.32)}$$

Where: c_1, c_2, c_3, c_4 accounts for the dumping factors for the MCP, PIP, and DIP joints respectively.

Euler-Lagrange equations are then derived for the generalized variables $\theta_1, \theta_2, \theta_3, \theta_4$:

$$L = K - P_g - P_e \textbf{ (3.33)}$$

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta_1}}\right) - \frac{\partial L}{\partial \theta_1} + \frac{\partial F}{\partial \dot{\theta_1}} = \tau_1 \textbf{ (3.34)}$$

3.2. Mathematical model for the hand exoskeleton

Based on the design requirements and the study of the natural movement of the hand, the design proposed for the hand exoskeleton is composed of two closed-loop chains created by the connection of the robotic structure with the user’s index. These chains are driving the flex/extension and abduction/adduction movements of the MCP joint, and the flex/extension movement of the PIP joint.

The proposed model is based on the studies [9] [10] ,and composed of (P3RP2R) chain to control MCP f/e and a/a motion and four bar linkage which is (4R) kinematic chain to control PIP f/e. the Since the PIP and DIP joints’ motions are coupled, the DIP joint’s motion is not considered here. And the chosen design ensures the only normal force on the finger.

One of the most critical issues in design exoskeletons is human–robot axes misalignment generated by the soft tissue. Instantaneous center of rotation (ICR) of the human joint, with respect to the ICR of the robot-actuated joint, should be fully defined by the geometry and posture of the closed chain. If this is not defined correctly, the ICR are misaligned, and the transfer of torque is compromised. The MCP joint’s misaligning displacements are modeled as three translations in three orthogonal directions $\delta_1, \delta_2, \delta_3$. And The MCP joints are presented as two orthogonal rotations α, θ. for the robot chains $q1, q2, q3, q4, q5$ are the angles and displacements of the MCP exoskeleton. $q2, q3$ are controlled using Bowden cables and the other ends of the cables are connected to two motors. For the PIP chain φ is connected to the motor using Bowden cables. And θ represents the PIP joint.

3.2.1. MCP closed-loop chain

3.2.1.1. Kinematics modeling

The denavit-hartenberg tables for both the MCP exoskeleton joints and the MCP finger joints can be viewed in Table 3.4and Table 3.5.

Table 3.4 Denavit-Hartenberg table for the MCP exoskeleton joints.

Joint	a_i	α_i	d	θ_i
1	0	0	0	q_1
2	L_1	$\frac{\pi}{2}$	0	q_2
3	L_2	$-\frac{\pi}{2}$	0	q_3
4	L_3	0	0	q_4
5	L_4	$\frac{\pi}{2}$	0	$\frac{\pi}{2}$

Table 3.5 Denavit-Hartenberg table for the MCP joint with displacements.

Joint	a_i	α_i	d	θ_i
1	V	0	0	$\frac{\pi}{2}$
2	H	0	0	$-\frac{\pi}{2}$
3	0	0	δ_1	0
4	0	$-\frac{\pi}{2}$	δ_2	$\frac{\pi}{2}$
5	0	$-\frac{\pi}{2}$	δ_3	0
6	0	$-\frac{\pi}{2}$	0	$\alpha-\frac{\pi}{2}$
7	0	$-\frac{\pi}{2}$	0	θ
8	L_6+q_5	0	0	$-\frac{\pi}{2}$
9	L_5	$-\frac{\pi}{2}$	0	$-\frac{\pi}{2}$

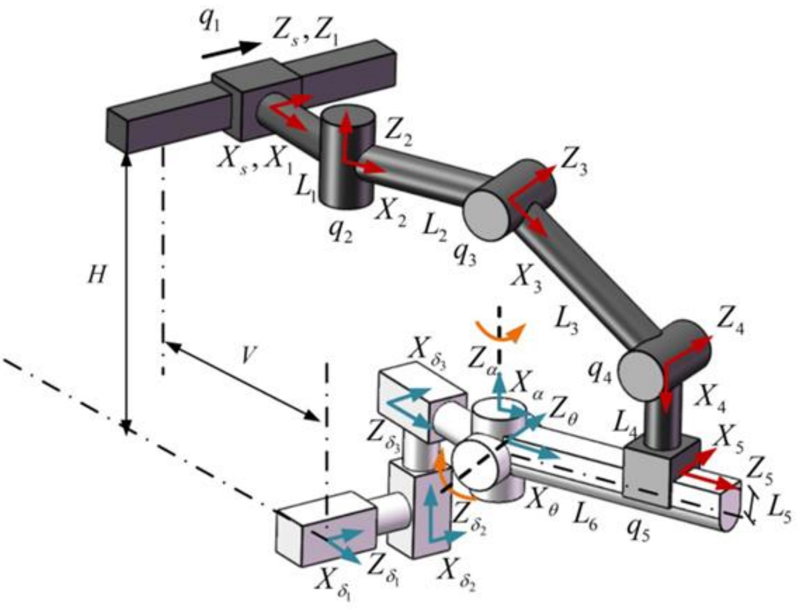


Figure 3.3 Kinematic chain of the MCP exoskeleton.

The loop closure equation for the proposed (MCP) chain of the index finger exoskeleton is given by:

$$Te_s^e = Th_s^e \quad \textbf{(3.35)}$$

Where Te_s^e is the transformation matric for the exoskeleton, given by:

$$Te_s^e = T_0^1 T_1^2 T_2^3 T_3^4 T_4^5 \qquad \textbf{(3.36)}$$

And Th_s^e is the transformation matrix of the human MCP joint, given by:

$$Th_s^e = T_0^{\delta_1} T_{\delta_1}^{\delta_2} T_{\delta_2}^{\delta_3} T_{\delta_3}^{\alpha} T_{\alpha}^{\theta} T_{\theta}^{5} \qquad \textbf{(3.37)}$$

The kinematic equation of the spatial closed-loop mechanism can be established from substituting equations (3.36) and (3.37) into equation (3.35):

$$\begin{cases} \theta = q3 + q4 - \frac{\pi}{2} \\ \alpha = q2 \\ \delta_1 = q1 + \left((L4 + L5) \cdot \cos(q3 + q4) + L2 + L3 \cdot \cos(q3) - (L6 + q5) \cdot \sin(q3 + q4)\right) \cdot \sin(q2) \\ \delta_2 = H - \left((L6 + q5) \cdot \cos(q3 + q4) - (L4 + L5) \cdot \sin(q3 + q4) - L3 \cdot \sin(q3)\right) \\ \delta_3 = L1 - V + (L2 + (L4 + L5) \cdot \cos(q3 + q4) + L2 + L3 \cdot \cos(q3) - (L6 + q5) \cdot \sin(q3 + q4)) \cdot \cos(q2) \end{cases}$$

(3.38)

By some mathematical calculation and converting equations (4.4) into the universal formula of trigonometric function the equation can be written as:

$$(\mathrm{B} + \mathrm{C})\mathrm{t}^2 - 2\mathrm{At} + \mathrm{C} - \mathrm{B} = 0 \quad \textbf{(3.39)}$$

Where:

$$A = (H - \delta_2 - L3 \cdot \sin(q3)) \cdot \cos(q2)$$

$$B = L1 - V - \delta_3 + (L2 + L3 \cdot \cos(q3)) \cdot \cos(q2)$$

$$C = (L4 + L5) \cdot \cos\,(q2)$$

$$t = \tan\,(\frac{\theta}{2})$$

The MCP f/e angle is given by:

$$\theta = 2\,atan\left(\frac{A+\sqrt{A^2-B^2+C^2}}{B+C}\right) \qquad \textbf{(3.40)}$$

3.2.2. Dynamic modeling

Based on the virtual work principle, there is:

$$\tau_q = J^T(q)\tau_\Theta \qquad \textbf{(3.41)}$$

Where:

$\tau_\Theta = [\,\tau_\theta\ \tau_\alpha\ \tau_{\delta 1}\ \tau_{\delta 2}\ \tau_{\delta 3}]$ represents the torques of the MCP joint. τ_θ and τ_α are the actuated torques applied in the flex/extension and abduction/adduction motion respectively. $\tau_\text{q} = [\,\tau_{q1}\ \tau_{q2}\ \tau_{q3}\ \tau_{q4}\ \tau_{q5}]$ represents the torques actuating the exoskeleton's joints.

The forces in the misaligning directions must be null to eliminate the reaction forces of the MCP joint in the translation directions, therefore:

$$\tau_{q1} = \tau_{q5} = 0,\ \tau_{q2} = \tau_\alpha,\ \tau_{q3} = \ \tau_{q4} = \tau_\theta \qquad \textbf{(3.42)}$$

Based on the latter, we define the generalized variable vector Θ to be $[q_1\ q_2]$.

Total kinetic energy for the closed-loop can be found as:

$$K = \frac{1}{2}m_1\left(v_{g1}\right)^2 + \frac{1}{2}\sum_{i=2}^{4} m_i\left(v_{gi}\right)^2 + \frac{1}{2}\sum_{i=2}^{4} I_i\left(\omega_{gi}\right)^2 + \frac{1}{2}I_{mcp}\left(\omega_{g0}\right)^2 \textbf{(3.43)}$$

where m_i and I_i are the mass and moment of inertia of the ith link, respectively. I_{mcp} is the estimated moment of inertia matrix of the human finger, and ω_{g0} is the angular velocity vector related to θ_1 and θ_2.

Potential energy can be derived as in:

$$P = \sum_{i=1}^{4} m_i g h_{gi} + m_0 g h_{g0} \qquad \textbf{(3.44)}$$

where h_{gi} is the position vector of the center of mass of the ith exoskeleton's link. h_{g0} is the position vector of the center of mass of the human finger.

The motion equation will be written in the form:

$$M\ddot{\Theta} + C\dot{\Theta} + G + J^T\tau_e \ = \tau \qquad \textbf{(3.45)}$$

Where M is the inertia matrix. C is the damping matrix. τ_e is the torque vector of the MCP joint. τ is the actuated torque vector of the exoskeleton joints q_2 and q_3.

3.3. PIP closed-loop chain

3.3.1. Kinematics modeling

This study was based on [4]. The closed loop equation of the four-bar mechanism in the planar case is described below:

$$l_1 + l_2\, cos\,\varphi - l_3\, cos\,\alpha - l_4\, cos\,\theta = 0 \quad \textbf{(3.46)}$$

$$l_2\, sin\,\varphi - l_3\, sin\,\alpha - l_4\, sin\,\theta = 0 \quad \textbf{(3.47)}$$

Where $l_1, l_2, l_3. l_4$ represents the dotted Links 1, 2, 3, 4 respectively.

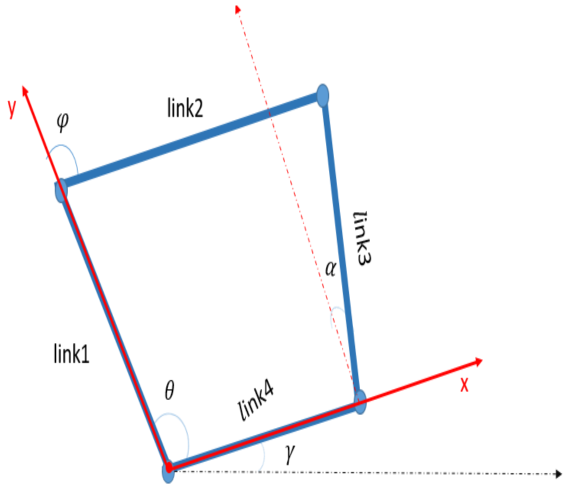


Figure 3.4 The diagram for the proposed 4-bar mechanism

Since θ is the f\e angle PIP joint, can be calculated directly by knowing the rotating angle of the finger's joint. We expressed φ (The rotating angle of the driving link) in terms of θ. So first, we rearranging the above equations into the following forms to eliminate α :

$$l_3 \cos\alpha = l_1 + l_2 \cos\varphi - l_4 \cos\theta \quad \textbf{(3.48)}$$

$$l_3 \sin\alpha = l_2 \sin\varphi - l_4 \sin\theta \quad \textbf{(3.49)}$$

The sum of the squares of equations (4.3) and (4.4) yields the well-known Freudenstein equation:

$$k_1(\theta) \sin\varphi + k_2(\theta) \cos\varphi + k_3(\theta) = 0 \quad \textbf{(3.50)}$$

Where k_1, k_2, k_3 are functions of θ ,and

$$k_1(\theta) = -2l_2 l_4 \sin\theta \quad \textbf{(3.51)}$$

$$k_2(\theta) = 2l_2(l_1 - l_4 \cos\theta) \quad \textbf{(3.52)}$$

$$k_3(\theta) = l_1^2 + l_2^2 - l_3^2 + l_4^2 - l_1 l_4 \cos\theta \quad \textbf{(3.53)}$$

Equation (3.50) can be solved in closed form and determines φ in terms of φ .So we define:

$$t = \frac{\tan\varphi}{2} \quad \textbf{(3.54)}$$

$$\sin\varphi = \frac{2t}{1-t^2} \quad \textbf{(3.55)}$$

$$\cos\varphi = \frac{1-t^2}{1+t^2} \quad \textbf{(3.56)}$$

Substituting equations (4.7) and (4.8) in equation (4.5) yields the following quadratic equation:

$$(k_3 - k_3)t^2 + (2k_1)t + (k_3 + k_2) = 0 \; \textbf{(3.57)}$$

We obtain the solution t equals to:

$$t = \frac{-k_1 \pm \sqrt{{k_1}^2 + {k_2}^2 - {k_3}^2}}{k_3 - k_2} \quad \textbf{(3.58)}$$

Substituting equation (4.6) into equation (4.10) yields:

$$\varphi(\theta) = 2\,\mathrm{arctan2}\;(-k_1 \pm \sqrt{{k_1}^2 + {k_2}^2 - {k_3}^2}, k_3 - k_2) \; \textbf{(3.59)}$$

And from equations (4.3) and (4.4) we obtain:

$$\alpha(\varphi, \theta) = \mathrm{arctan2}\;(l_2 \sin\varphi - l_4 \sin\theta\,, l_1 + l_2\cos\varphi - l_4\cos\theta) \; \textbf{(3.60)}$$

3.3.2. Velocity analysis:

Differentiating the equations (3.46) and (3.47) with respect to time and writing them in matrix form yield:

$$\begin{bmatrix} l_4\sin\theta & l_3\sin\alpha & l_2\sin\varphi \\ -l_4\cos\theta & -l_3\cos\alpha & l_2\cos\varphi \end{bmatrix} \begin{bmatrix} \dot{\theta} \\ \dot{\alpha} \\ \dot{\varphi} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \; \textbf{(3.61)}$$

we rearrange the equation into the following:

$$\begin{bmatrix} l_3\sin\alpha & l_2\sin\varphi \\ -l_3\cos\alpha & l_2\cos\varphi \end{bmatrix} \begin{bmatrix} \dot{\alpha} \\ \dot{\varphi} \end{bmatrix} = \begin{bmatrix} -l_4\sin\theta \\ l_4\cos\theta \end{bmatrix} [\dot{\theta}] \; \textbf{(3.62)}$$

And obtain the velocity model:

$$\begin{bmatrix} \dot{\alpha} \\ \dot{\varphi} \end{bmatrix} = \begin{bmatrix} S_1(\theta,\alpha,\varphi) \\ S_2(\theta,\alpha,\varphi) \end{bmatrix} [\dot{\theta}] \quad \textbf{(3.63)}$$

Where S_1 and S_2 are define as fallow:

$$S_1(\theta,\alpha,\varphi) = \frac{\partial\alpha}{\partial\theta} = \frac{l_4\sin(\varphi-\theta)}{l_3\sin(\alpha-\varphi)} \quad \textbf{(3.64)}$$

$$S_2(\theta,\alpha,\varphi) = \frac{\partial\varphi}{\partial\theta} = \frac{l_4\sin(\alpha-\theta)}{l_2\sin(\alpha-\varphi)} \quad \textbf{(3.65)}$$

We can also rearrange the equation (3.58) into the following:

$$\begin{bmatrix} l_3\sin\alpha & l_4\sin\theta \\ -l_3\cos\alpha & -l_4\cos\theta \end{bmatrix} \begin{bmatrix} \dot{\alpha} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} -l_2\sin\varphi \\ -l_2\cos\varphi \end{bmatrix} [\dot{\varphi}] \quad \textbf{(3.66)}$$

We can also write the velocity model in another form:

$$\begin{bmatrix}\dot{\alpha}\\ \dot{\theta}\end{bmatrix} = \begin{bmatrix}\frac{\partial\theta}{\partial\varphi}\\ \frac{\partial\alpha}{\partial\varphi}\end{bmatrix} = \begin{bmatrix}\frac{l_2\sin(\alpha-\varphi)}{l_4\sin(\theta-\alpha)}\\ \frac{l_2\sin(\theta-\varphi)}{l_3\sin(\alpha-\theta)}\end{bmatrix} = \begin{bmatrix}N_2(\theta,\alpha,\varphi)\\ N_1(\theta,\alpha,\varphi)\end{bmatrix}[\dot{\varphi}] \qquad \textbf{(3.67)}$$

3.3.3. Dynamic analysis

The dynamic model is derived using the Lagrange formulations. The total kinetic energy of the system is determined as:

$$T = \frac{1}{2}(m_2\|\mathcal{v}_{c2}\|^2 + I_2\dot{\varphi}^2) + \frac{1}{2}(m_3\|\mathcal{v}_{c3}\|^2 + I_3\dot{\alpha}^2) + \frac{1}{2}\left(m_{41}\|\mathcal{v}_{c41}\|^2 + I_{41}\dot{\theta}^2\right) + \frac{1}{2}\left(m_{42}\|\mathcal{v}_{c42}\|^2 + I_{42}\dot{\theta}^2\right) \qquad \textbf{(3.68)}$$

Where:

$$\|\mathcal{v}_{c2}\|^2 = {l_{c2}}^2\dot{\varphi}^2$$

$$\|\mathcal{v}_{c3}\|^2 = {l_4}^2\dot{\theta}^2 + {l_{c3}}^2\dot{\alpha}^2 + 2 * l_4 l_{c3}\cos(\theta-\alpha)\,\dot{\theta}\dot{\alpha}$$

$$\|\mathcal{v}_{c41}\|^2 = ({l_{42}}^2 + {l_{c41}}^2)\dot{\theta}^2$$

$$\|\mathcal{v}_{c42}\|^2 = {l_{c42}}^2\dot{\theta}^2$$

m_i is the mass of link i.

$\mathcal{v}_{ci}$ is the velocity of the center of mass of link i.

I_i is the moment of inertia of link i about its center of mass.

The potential energy of the mechanism is given as below:

$$V = m_1 g y_{c1} + m_2 g y_{c2} + m_3 g y_{c3} + m_4 g y_{c4} \qquad \textbf{(3.69)}$$

Where g is the gravitational acceleration, and

$$y_{c1} = l_{c1}\cos\gamma + h \qquad \textbf{(3.70)}$$

$$y_{c2} = l_1\cos\gamma + l_{c2}\cos\varphi\cos\gamma + h \qquad \textbf{(3.71)}$$

$$y_{c3} = l_{c3}\cos\alpha\cos\gamma + l_4\cos\theta\cos\gamma + h \qquad \textbf{(3.72)}$$

$$y_{c4} = l_{c4}\cos\theta\cos\gamma + h \quad \textbf{(3.73)}$$

And h is the height of the finger's joint above the ground. The motion equation of the entire system is given by:

$$\mathcal{L} = T - V \quad \textbf{(3.74)}$$

From Lagrangian equation we can determine the dynamic model and the equation of motion for the system:

$$\frac{d}{dt}\left(\frac{\partial\mathcal{L}}{\partial\dot{\phi}}\right) - \frac{\partial\mathcal{L}}{\partial\phi} = \tau_{ext} \qquad \textbf{(3.75)}$$

3.4. Kinematic manipulability

The scaled joint velocity of the exoskeleton can be defined as:

$$\widetilde{\dot{\Theta}} = W_{\dot{q}}\dot{\Theta} = \begin{bmatrix} 1/q_{2,max} & 0 \\ 0 & 1/q_{3,max} \end{bmatrix}\dot{\Theta} \quad \textbf{(3.76)}$$

The local kinematic manipulability measure w_v represents the velocity transmission ability from the actuated joints' velocities to the MCP joint's velocities [9].

$$w_v = \sqrt{\det\left(J^{-T}(\Theta)W_{\dot{q}}^{T}W_{\dot{q}}J^{-1}(\Theta)\right)^{-1}} \qquad \textbf{(3.77)}$$

Local kinematic manipulability measure w_v shows the performance for a given configuration. For evaluating the kinematic manipulability for different dimensional parameters, the global kinematic manipulability measure (GKM) will be used. It can be expressed as:

$$w_{gv} = \frac{\int_W w_v dW}{\int_W dW} \quad \textbf{(3.78)}$$

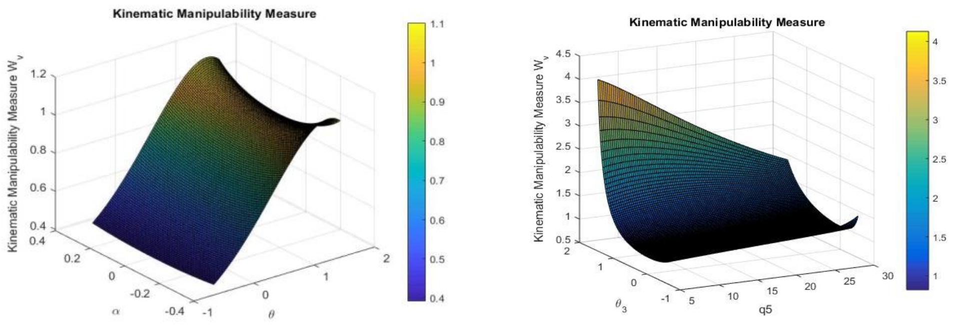


Figure 3.5 Kinematic manipulability measure with respect to changes in control variables for MCP (left) and PIP (right).

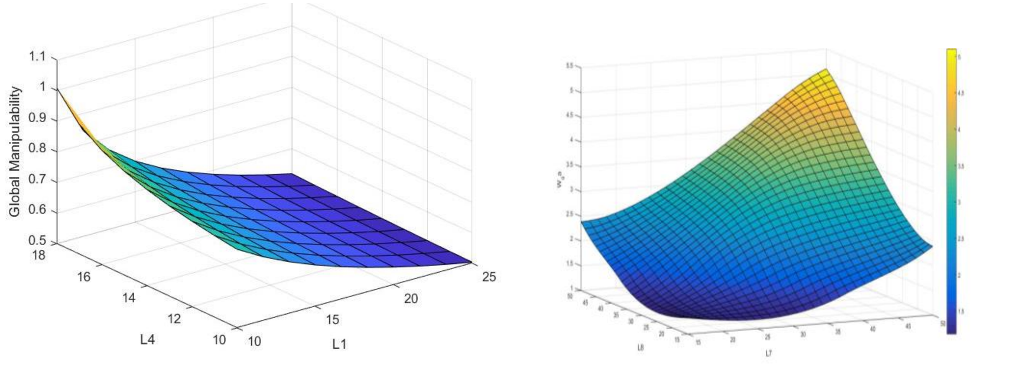


Figure 3.6 Global kinematic manipulability measure with respect to changes in dimensional variables for MCP (left) and PIP (right).

3.5. Dynamic manipulability

The scaled joint torque can be written as in:

$$\tilde{\dot{\tau}} = W_{\dot{\tau}}\dot{\tau} \quad \textbf{(3.79)}$$

Where $W_{\dot{\tau}}$ is the scaling matrix with the limits of the joints torques.

The acceleration manipulability measure is used to describe the ellipsoid of the acceleration from the actuated joints of the hand exoskeleton to the MCP and PIP joints. The local and global measure can be expressed as in respectively:

$$w_a = \sqrt{det\left(B_1{}^T W_\tau^T W_\tau B_1\right)^{-1}} \quad \textbf{(3.80)}$$

Where $B_1 = M\,J^{-1} + J^T Me$, M is the inertia matrix of the hand exoskeleton and Me is the estimated inertia matrix of the finger's phalange.

$$w_{ga} = \frac{\iint w_v d\alpha d\theta}{\iint d\alpha d\theta} \quad \textbf{(3.81)}$$

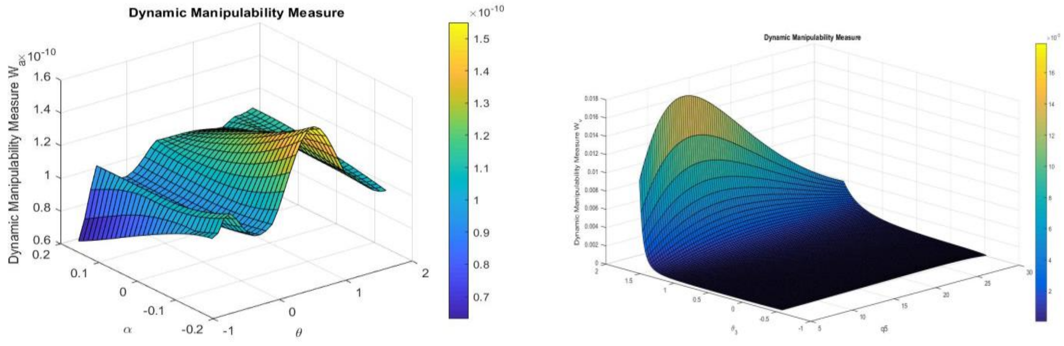


Figure 3.7 Local dynamic manipulability measure with respect to changes in dimensional variables for MCP (left) and PIP (right).

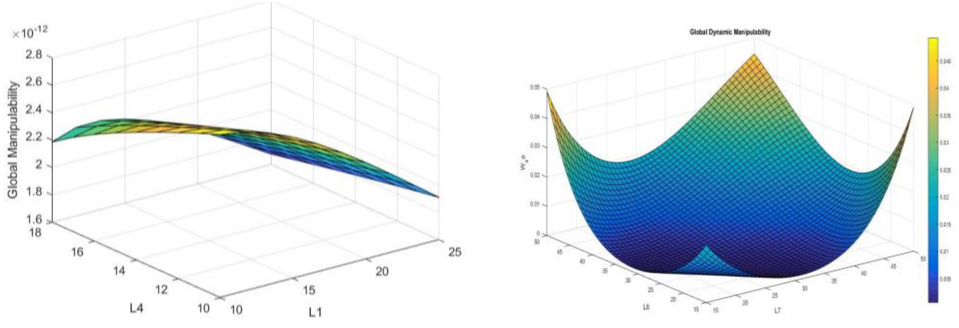


Figure 3.8 Global Dynamic manipulability measure with respect to changes in dimensional variables for MCP (left) and PIP (right).

3.6. Hand CAD model

Regarding the dimensions of the fingers and the palm; different previous case studies have been conducted to collect these data. In this document, we refer to data reported in [11] for phalanges lengths of the index, middle, ring, and little fingers, the measurements that were obtained from a variant group of 159 subjects, 82 males and 77 females (18-25years).

Table 3.6 Lengths of the fingers' phalanges of a male obtained from a reported study.

	I1	I2	I3	M1	M2	M3	R1	R2	R3	L1	L2	L3
Male right hand	2.32	2.37	2.65	2.60	2.78	2.80	2.29	2.56	2.76	1.96	1.92	2.51
Male Left hand	2.32	2.39	2.61	2.60	2.82	2.75	2.30	2.59	2.78	1.95	1.98	2.49

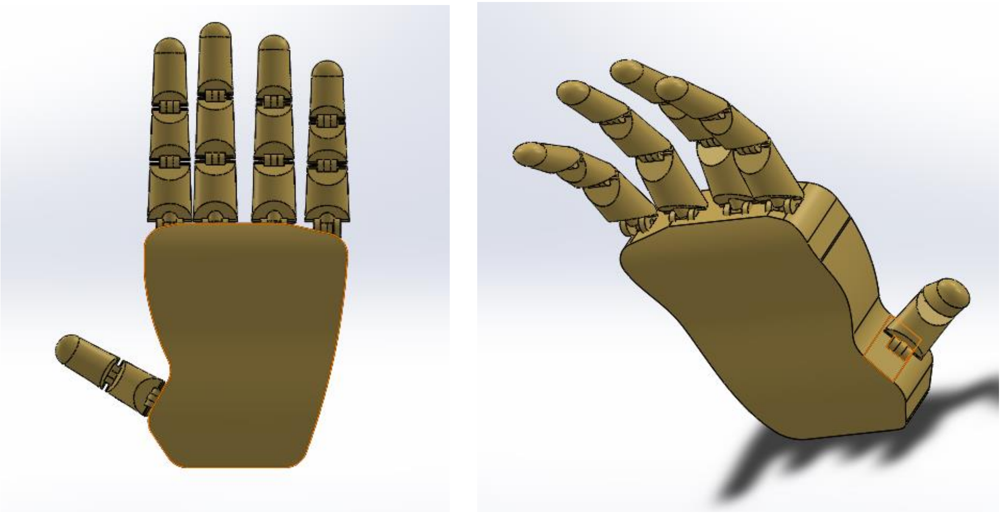


Figure 3.9 The developed CAD model of the human hand.

3.7. Exoskeleton CAD model

Based on the kinematic manipulation results, the dimensional parameters were chosen for the exoskeleton of the index finger. The parameter set can be viewed in

Table 3.7 The dimensional parameters set for the exoskeleton of the index finger.

L1	10.5mm
L2	20.5mm
L3	45mm
L4	8.6mm

L5	13.2mm
L6	36mm
H	22mm
V	39mm

The CAD model developed can be viewed in Figure 3.10.

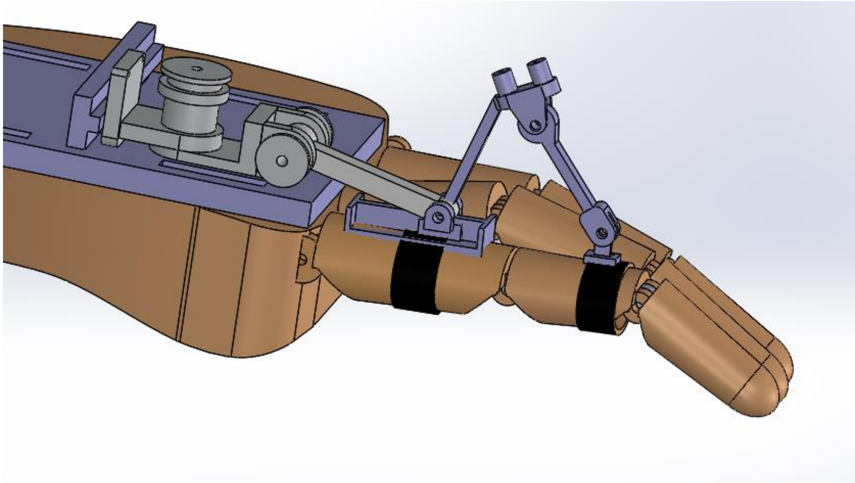


Figure 3.10 The CAD model of the hand exoskeleton showing the design for the MCP and PIP closed-loop chains.

4 Hand Pose Estimation

In this section, the problem of hand pose estimation will be discussed which is an essential part in the process of building a forecasting model for the hand exoskeleton.

Pose Estimation is the problem when achieving coordinates of specific key-points in an image is required. It can be face, body, hand, or animal pose estimation where the key-points are joints in the desired pose-estimated body. It has applications in augmented reality, virtual reality, …etc. The algorithms used in this approach use CNNs as the main building blocks because they imbed steps of feature extraction and post-processing [12] making an end-to-end system that takes an image and directly returns the coordinates of each key-point. Furthermore, CNNs have shown robust performance over hand-crafted features and graphical models [12] [13]. The pose estimation algorithms aim to find the structured relations between key-points. They tend to achieve large receptive fields on the image.

A large receptive field on the image indicates the area in the input image that contributes to the value of a particular feature in the convolutional layer. A large receptive field makes the model capture more contextual information from the image. Many models were proposed for pose estimation such as Convolutional Pose Machines (CPM) [13], Stacked Hourglass (SHG) [14], Real Time Multi Pose nets(RTMPose) [15], Lite High Resolution Nets (LiteHR Nets) [16], and many more. Their architectures will be explained in this section.

4.1. Convolutional Layers

These layers are the building blocks of Convolutional Neural Networks. Together with Pooling layers form the feature extractors in CNNs. A CNN usually consists of several Convolutional and Pooling layers followed by one or two MLP layers [12] [17]. The importance of the Convolutional layers comes from the outstanding performance of CNNs in Image classification tasks. Treating each pixel in the image as an individual feature like in a pure MLP Network has disadvantages. The full connection in MLP layers makes the model search for a relation among all the features. However, no relation exists between a pixel in the middle and one far from it at the corner of the image. Therefore, this increases the overfitting state which gives superiority to CNNs over MLPs. In CNNs, the model is giving priority to nearest neighbors over other pixels. In other words, it is looking for relations between a pixel and its neighbors in an image. In MLPs, the pre-activation step was [17]:

$$z^{l} = a^{l-1} \times W^{l} + a_{0}^{l-1} \quad \textbf{(4.1)}$$

Where in Convolutional layers it is:

$$A^{l} = X^{l-1} * W^{l} + b^{l} \qquad \textbf{(4.2)}$$

Where * here denotes the convolution process not normal vector-matrix multiplication. X^{l-1} is a matrix representing pixels in a $height \times width \times channels$ form, W^{l} is a kernel with a specified size and number of channels which corresponds to the updating weights in the training process, and b^{l} is the bias of this layer. The activation function step remains the same as in MLPs:

$$H = \varphi(A) \qquad \textbf{(4.3)}$$

4.1.1. Cost Functions

The most popular cost functions are those two [12]:

$$J = \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{m}\left(\tilde{y}_{j}^{i} - y_{j}^{i}\right)^{2} + \frac{\lambda}{2}\|w\|^{2} \quad \textbf{(4.4)}$$

$$J = \sum_{i=1}^{n}\sum_{j=1}^{m} -\tilde{y}_{j}^{i} \times \log\,(y_{j}^{i}) + \frac{\lambda}{2}\|w\|^{2} \qquad \textbf{(4.5)}$$

Where n is the number of samples in the dataset, m is the number of neurons in the output layer, y_{j}^{i} is the prediction of the model for sample i and neuron j, $\tilde{y}_{j}^{i}$ is the target of sample i and neuron j, and $\frac{\lambda}{2}\|\boldsymbol{w}\|^{2}$ is a term usually added to the cost function to reduce overfitting. The first cost function is mostly used in applications of regression while the second takes place in classification tasks.

4.2. Classic networks:

4.2.1. LeNet-5

The goal of this network is to recognize the handwritten digit on the gray image but the model actually can be a starting point on how to build recognition systems. Back then it replaced the designed learning machine based on pixel operation, instead, it used a combination of automatic learning techniques and they are two main models. The first model is the feature extractor and the trainable classifier is the second model [18]. It consists of 7 layers:

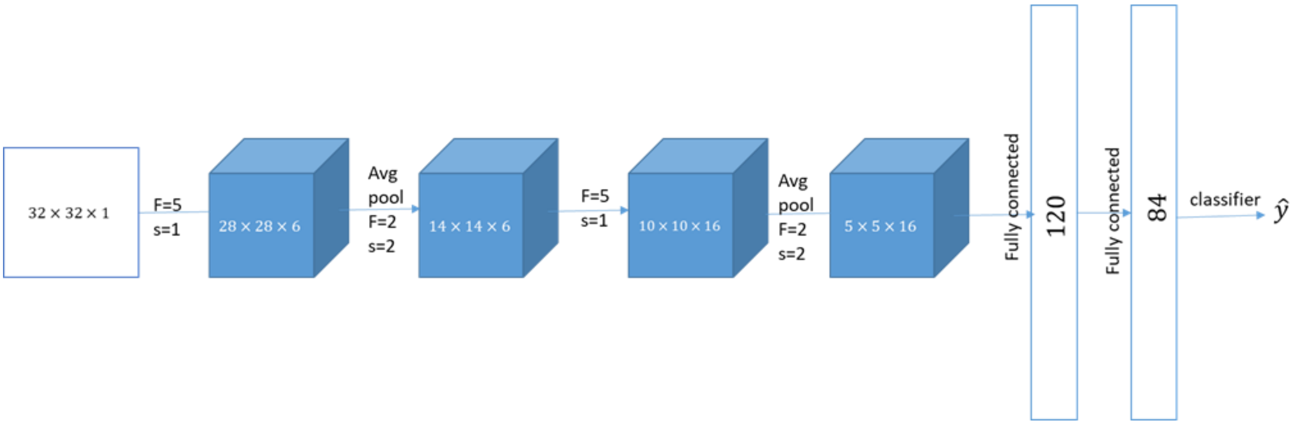


Figure 4.1 Architecture of LeNet-5

4.2.2. Alex Net:

This model works in the object recognition approach. The network consists of five convolutional layers, some of which are followed by max-pooling layers and then three fully connected layers. It is bigger than LeNet-5. [19]

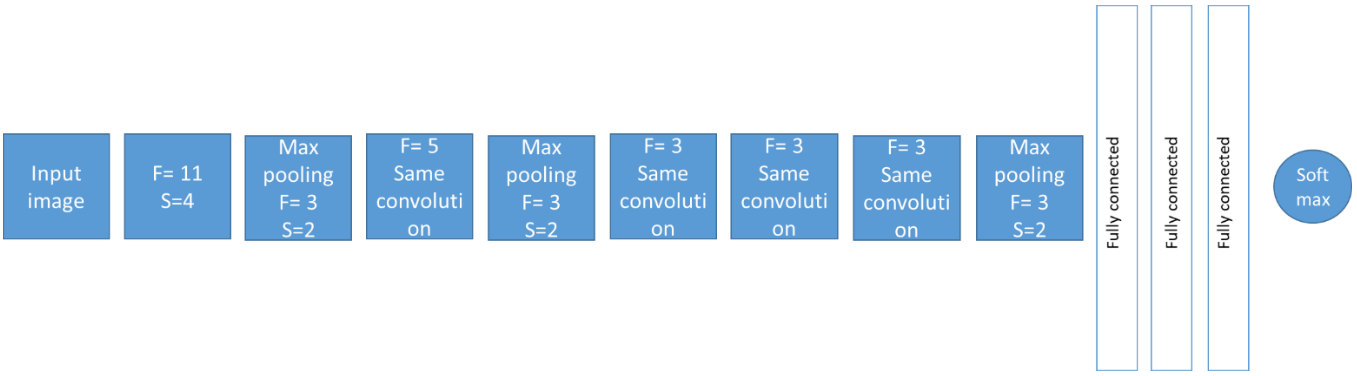


Figure 4.2 Architecture of Alex Net

4.2.3. VGG16:

This network was typically used in the ILSVRC competition for image classification and object localization. Their research achieved somehow accurate results back so they took the first and second place at the competition in 2014 [20].

4.2.4. ResNet

To make the process of training deep networks easier, [21] proposed a framework that reformates the layers of the network with references to increase the accuracy of the results and achieve optimization. As the network becomes deeper and deeper, the accuracy decreases and it becomes more complicated to be trained because of the vanishing and exploding gradient. Theoretically, building a deep network decreases the error but it makes it worse at a determined number of layers. The main idea is to build many residual blocks and then stack them together. This framework helps build a deep network without any loss in performance.

4.3. Modern HPE models

What makes hand pose estimation challenging among pose estimation genres is that the hand suffers from self-occlusion and self-dexterous, and it is highly articulated [22]. 21 key-points are demanded to be predicted which are 3 joints in each finger in addition to 5 fingertips and a point on the palm, as shown in Figure 4.3.

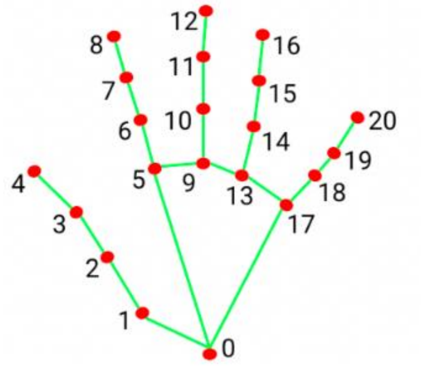


Figure 4.3 The 21 key-points used for graphing the hand pose.

Hand Pose Estimation can be 2D or 3D pose estimation where the difference is finding the depth of the key-points additionally in 3D hand pose estimation. It can also be depending on multi-view RGB systems [22], depth-based solutions [23], or monocular RGB solutions [22], [13]. 2D hand pose estimation is essential as it can be a sub-module in 3D hand pose estimation [22]. Next, the architectures of CPM, SHG, RTM-Pose, and LiteHR-Nets are introduced with the flow of data in each one of them.

4.3.1. Deep Pose:

In [24], the usage of deep learning for pose estimation is first introduced as a DNN-based regression problem. Early methods had limitations in detecting the joints as in [25], [26]. The main task is to determine the head, the shoulder, etc. They addressed challenges to determine the small visible joints and the occlusion problem.

4.3.2. Stacked Hourglass Network:

It is used to capture and process information at every scale for the face, hand, and body features. The datasets used were FLIC and MPII Human Pose. It uses a symmetrical structure composed of repeated bottom-up and top-down layers. It consists of 3 essential modules:

4.3.2.1. Hourglass Module

The symmetric topology makes the network able to extract the global details and the local details. The convolution and max pooling reduce the resolution (bottom-up layers) while in the middle, there are a series of conv layers that process the features. Then, using the up-sampling technique for example nearest neighbor or bilinear interpolation. However, the convolutional layers maintain the important features.

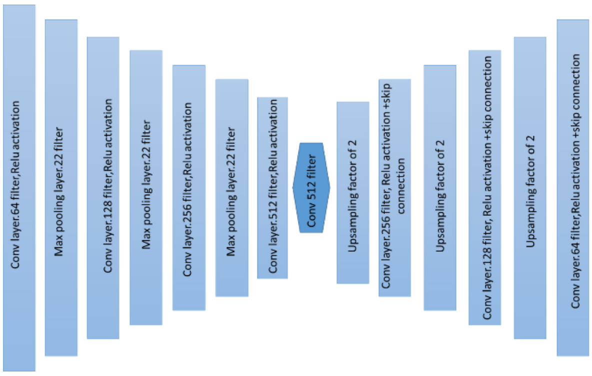

Conv layer.64 filter,Relu activation
Max pooling layer.22 filter
Conv layer.128 filter,Relu activation
Max pooling layer.22 filter
Conv layer.256 filter,Relu activation
Max pooling layer.22 filter
Conv layer.512 filter,Relu activation
Conv 512 filter
Upsampling factor of 2
Conv layer.256 filter, Relu activation +skip connection
Upsampling factor of 2
Conv layer.128 filter, Relu activation +skip connection
Upsampling factor of 2
Conv layer.64 filter,Relu activation +skip connection

Figure 4.4 Architecture of hourglass module.

4.3.2.2. Residual Modules

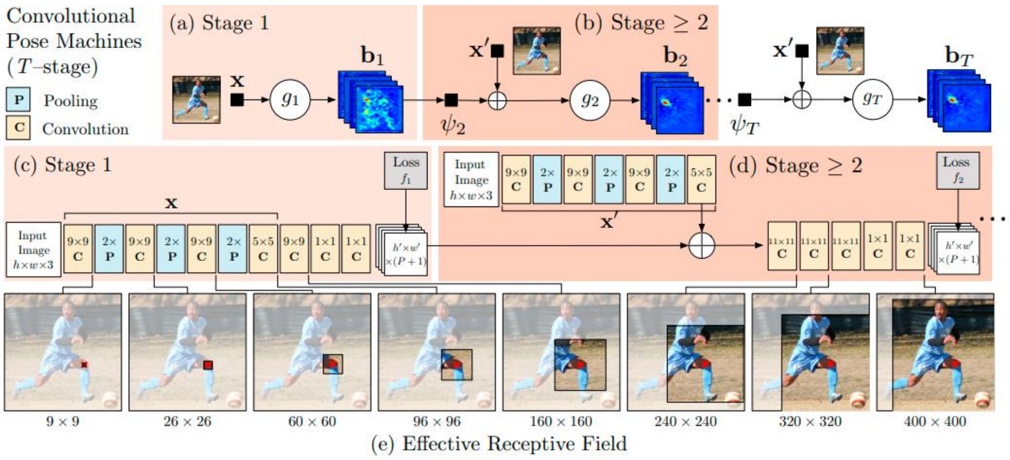
They improve performance and reduce memory usage by preserving the loss of information.

4.3.2.3. Stacking Hourglass Modules

Multiple hourglass modules are stacked together end to end each module tacks the output of the previous module. The features are combined from different scales to produce a heat map indicating the probability of the joint. The networks predict the heat map H for each joint. The network splits the produced heatmaps and applies 1 × 1 convolution which remaps the output to extract the needed feature [14].

4.3.3. Convolutional Pose Machines (CPM)

Figure 4.5 shows the complete architecture of CPM model [13]:



(e) Effective Receptive Field

Figure 4.5 Architecture and Receptive Field of CPM.

CPM is composed of multiple stages T such that after each stage the outputted belief maps are refined and closer to the target belief maps. In stage 1, The input image is passed through multiple convolutional and pooling layers as shown in Figure 4.5 (c). This produces x_z which corresponds to the features extracted from the input image at the location $\forall z \epsilon Z$ where Z is the set of all locations in the image. x_z is then passed to g_1 which consists of only convolutional layers. Note that g_1 is three kernels of size $9 \times 9 \times (P+1)$, $1 \times 1 \times (P+1)$, and $1 \times 1 \times (P+1)$ respectively. Passing x_z to g_1 will produce belief maps b_1 which is $h' \times w' \times (P+1)$:

$$g_1(x_z) = \{b_1^p(Y_p = z)\}_{p\in\{0,P\}} \quad \mathbf{(4.6)}$$

Where $b_1^p(Y_p = z)$ is the score predicted by classifier g_1 assigning each part p to the image location z. In stages 2-T, the belief maps are computed according to both the features in the current stage and the belief maps produced by the previous stage. All b_i belief maps are resized to $h \times w \times (P+1)$. Then, the function ψ apply a convolution to b_1 taking local neighborhood around z. The result is combined with x'_z which is the features of stages 2-T. Note that x_z is allowed to be different from x'_z and x'_z is the same for all stages 2-T. Multiple stages increase the size of the receptive field which leads to a highly refined estimation. However, the problem of vanishing gradient makes it difficult to train such a deep network with multiple stages. Therefore, Intermediate supervision is used in which a cost function is applied multiple times within the network. In this case, the cost function is defined at the output of each stage as follows:

$$f_t = \sum_{p=1}^{P+1} \sum_{z \epsilon Z} \left\| b_t^p(z) - b_*^p(z) \right\|_2^2 \qquad \mathbf{(4.7)}$$

Where $b_t^p(z)$ is the belief map p of stage t and $b_*^p(z)$ is the ideal belief map which can be obtained by putting Gaussian peaks at the ground truth locations of each body part p. The overall cost function is then:

$$F = \sum_{t=1}^{T} f_t \quad \textbf{(4.8)}$$

All the weights in the architecture are updated simultaneously using the overall cost function by applying optimizers. In [13], stochastic gradient descent is used.

4.3.4. Real Time Multi-Person Pose nets (RTMPose)

This Algorithm aims to make a good compromise between latency and accuracy [15]. The architecture can be shown in Figure 4.6:

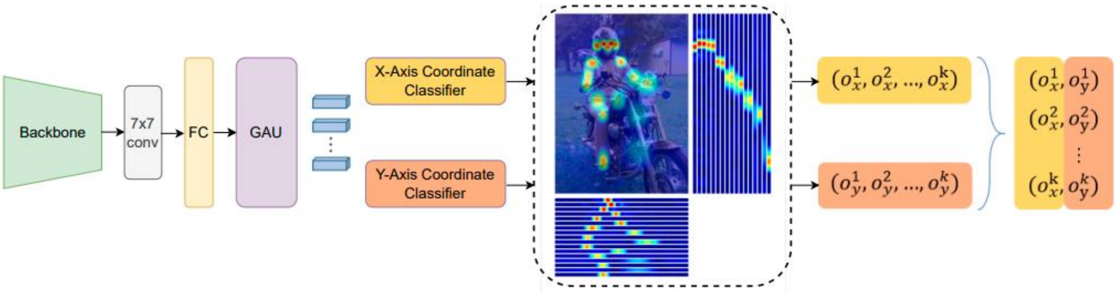

Backbone
7x7 conv
FC
GAU
X-Axis Coordinate Classifier
Y-Axis Coordinate Classifier
$(o_x^1, o_x^2, ..., o_x^k)$
$(o_y^1, o_y^2, ..., o_y^k)$
(o_x^1, o_y^1)
(o_x^2, o_y^2)
(o_x^k, o_y^k)

Figure 4.6 The overall architecture of RTMPose.

RTMPose does not adopt the heatmap approach. Instead, it uses the SimCC algorithm that treats the keypoints localization as a classification problem. The backbone in Figure 4.6 is simply a CNN that corresponds to the feature extractor. Choosing the backbone adjusts the speed/accuracy tradeoff. In RTMPose, the backbone is chosen to be CSPNeXt because it shows a good balance and a friendly deployment according to [13]. The Fully Connected layer (FC) classifies the features and passes the results to the Gate Attention Unit (GAU) which refines them.

4.3.5. Lite HRnet

Lite HRnet is high resolution network designed for different resolution feature learning. The main contribution of this paper is the use of conditional channel weighting (CCW) to replace costly 1×1 convolutions. Conditional Weighting computation:

$$(W_1, W_2, \dots., W_s) = H_s(X_1, X_2, \dots X_s) \quad \textbf{(4.9)}$$

$H_s$is the light-weighted function that is used to compute conditional weight for the maps. It consists of multiple stages as follows: Conv→ ReLU → Conv → sigmoid. All in order to reduce the channels and compute the weight.

Conditional Channel Weighting (CCW)

$$Y_s = W_s \odot X_s \qquad \textbf{(4.10)}$$

Where: X_s is the input feature map, W_s is a weight map (computed for each channel). $\odot$ is element-wise multiplication. [16]

4.4. Metrics

The Metrics used to compare between algorithms are: AP, AR, GFLOPs, and PCK.

4.4.1. Average Precision & Average Recall (AP & AR)

These two metrics are computed using the Precision and Recall concepts:

$$Precision = \frac{TP}{TP+FP} \qquad \textbf{(4.11)}$$

$$Recall = \frac{TP}{TP+FN} \qquad \textbf{(4.12)}$$

The precision is the true positive values over the whole number of positives. Recall is the ratio of True Positives to the total number of actual positives. Where TP refers to True Positives, FP refers to False Positives, and FN refers to False Negatives.

After predicting key-points, each key-point is given a confidence level. For different thresholds on the confidence level, true positives among the key-points are determined. First, calculate Both the Precision and Recall for all the instances in the dataset for each threshold. Note that lower values for this threshold generally increase recall and decrease precision. Plot precision on y-axis and recall on x-axis. Interpolate to get more points. AP is area under the precision-recall curve.

$$AP = \int_0^1 P(r)dr \qquad \textbf{(4.13)}$$

Where n is the number of points in the Precision-Recall plane. AR is the recall values averaged for different threshold values in the range [0.5,1].

$$AR = 2 \times \int_{0.5}^{1} recall(thr)dthr \quad \textbf{(4.14)}$$

Both AP and AR are always in the range [0,1]. The larger the value of AP, the more the model showed robustness to conditions when confidence level is low. Also, larger values for AR reflect the same conclusion. These two were essential in evaluating algorithms in [16] and [25].

4.4.2. GFLOPs

The computation cost can affect the algorithm’s performance to work in real time applications. The metric for that benchmark is GFLOPs in order to estimate the performance in speed and resource usage [16] [25]. Most scientist calculate floating-point as an important benchmark focusing on multiplying and other operations.

$$FLOPs\ per\ second = \frac{Total\ FLOPS}{excution\ time} \quad \textbf{(4.15)}$$

$$GFLOPS = \frac{Total\ FLOPSFLOPs\ per\ second}{10^9} \quad \textbf{(4.16)}$$

FLOPS can be determined from the network architecture. How PyTorch provides functions that can help calculating this metric. The function is profile_macs where we pass the model and the data to calculate FLOPS while we calculate the execution time.

4.4.3. PCK and PCKh

PCK stands for percentage of correct keypoints while PCKh stands for percentage of correct keypoints with head-normalization. Those metrics measure the accuracy of hand pose estimation methods.

$$PCK = \frac{number\ of\ keypoint\ within\ threshold}{total\ keypoint} \times 100 \ \textbf{(4.17)}$$

4.5. MediaPipe

In 2020, google researchers developed an open-source framework that can track hand’s key-points accurately in real time. The pipeline consists of two models which are the palm detector and the hand landmark model. The palm detector operates on the image to identify the location of the hand then the hand landmark returns the position of the key-point on the image [27].

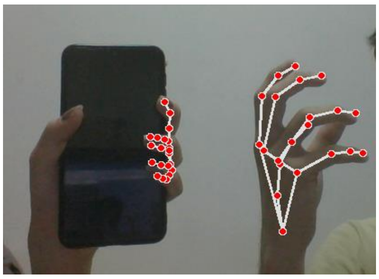


Figure 4.7 Results of testing MediaPipe.

4.6. MMpose

MMpose is a pose estimation framework which provides a wide range of abilities. It is the most famous and commonly used among scientific community, highly rated in the GitHub community among pose estimation frameworks and toolboxes. It provides many architectures for face, body, hand, whole-body, and animal pose estimation. All trained and tested on benchmark datasets such as Freihand and Coco-Whole-body, which are all embedded within the framework with several backbones to implement new models or further train the existing models.

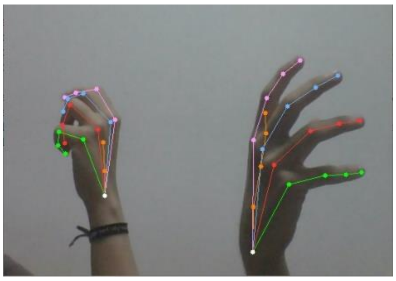


Figure 4.8 Results of testing MMpose.

These are the results comparing the hand pose estimation algorithms together

Table 4.1 Results of comparing the hand pose estimation algorithms using the GFLOP matrix.

Algorithm	GFLOP
Mediapipe	N/A
HRNets	10.27
LiteHRNets	0.4253
8-stage Hourglass	14.3
RTMPose	2.581

Table 4.2 shows the similarity of RTMPose and HRNets in terms of accuracy. Only the RTMPose model are pre-trained on AIC+COCO and fine-tuned on the corresponding dataset.

Table 4.2 Comparison between RTMPose and HRNets in terms of accuracy

Dataset	**Methods**	**AP%**
AP-10k	HRNets	72,2
	RTMPose	72,2
CrowdPose	HRNets	67.5
	RTMPose	70.6

Table 4.3 shows that HRNets overcomes both LiteHRNets and 8-stage Hourglass in terms of accuracy. The models are compared to each other according to Coco validation set.

Table 4.3 Comparison between HRNets, LiteHRNets and 8-stage Hourglass in terms of accuracy

Model	AP%	AR%
HRNets	73.4	78.9
8-stage Hourglass	66.9	-
LiteHRNets	67.2	73.3

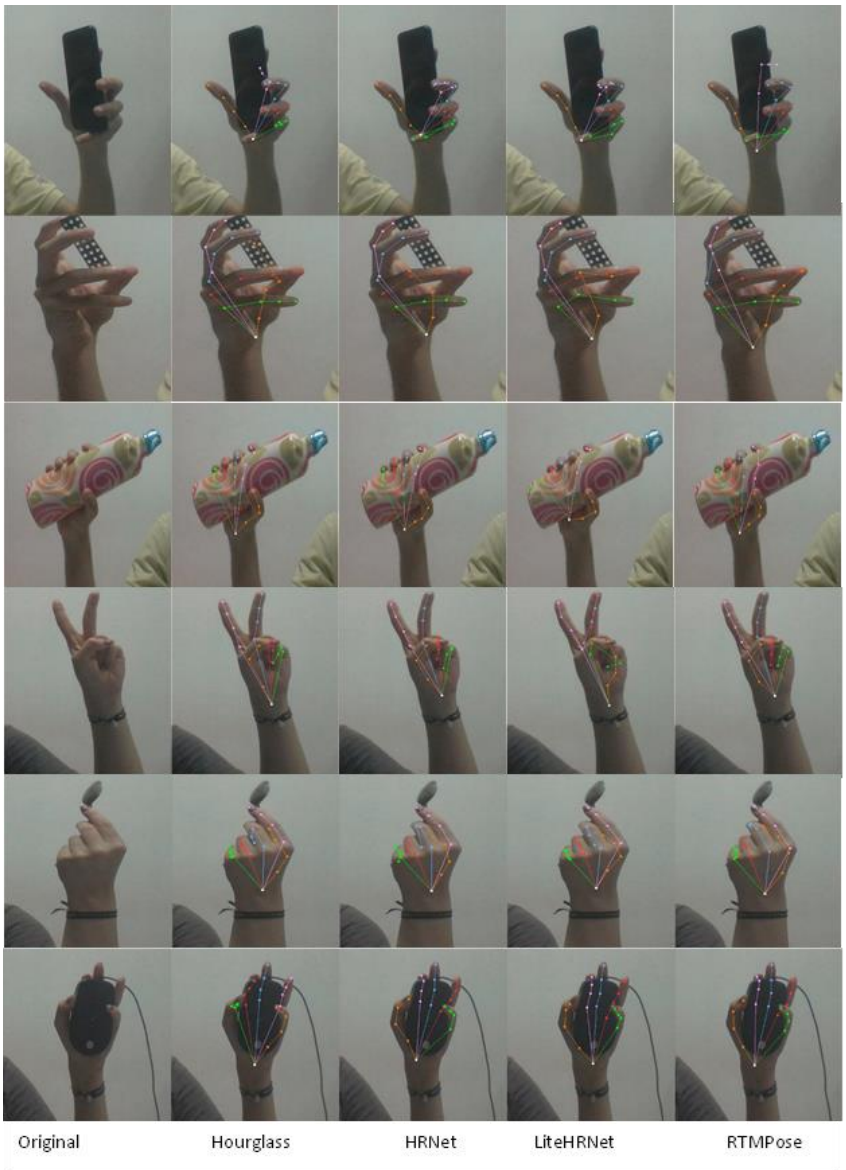


Figure 4.9 Results of testing HPE algorithms (Hourglass, HRNet, LiteHRNet, RTMPose).

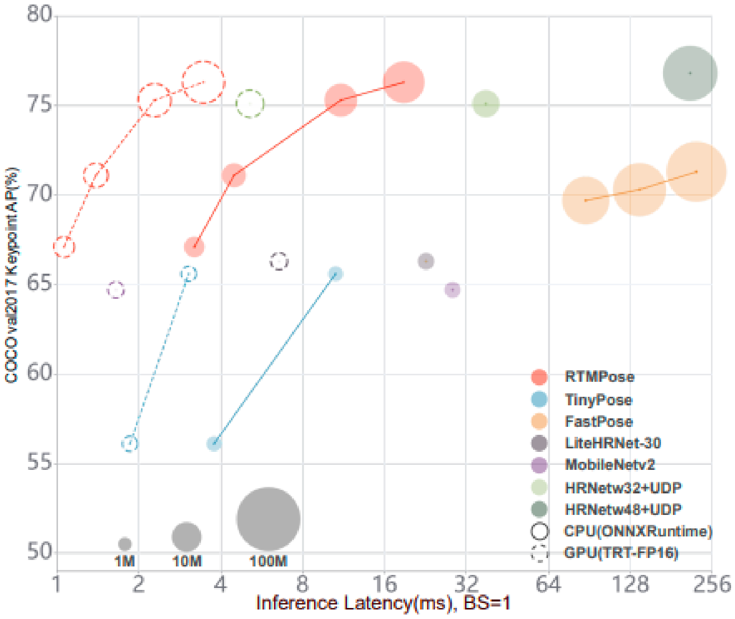


Figure 4.10 Comparison of RTMPose and open-source libraries on COCO-val set regarding model size, latency, and precision. [11].

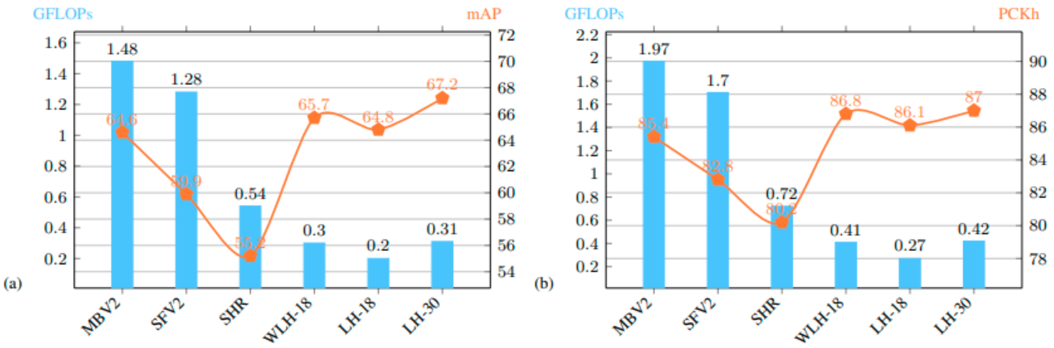


Figure 4.11 the complexity and accuracy comparison on the COCO val and MPII val sets. (a) Comparison on COCO val with 256 × 192 input size. (b) Comparison on MPII val with 256 × 256 input size. MBV2= MobileNet V2. SFV2= ShuffleNet V2. SHR= Small HRN [19].

In conclusion, the most accurate model was HRNets and RTMPose. However, HRNets are extremely computationally complex. On the other hand, LiteHRNets are the most efficient in terms of time complexity. RTMPose has the best compromise between accuracy and real time efficiency. It works more accurate than LiteHRNets with an acceptable time complexity. Note that both RTMPose and LiteHRNet showed an outstanding performance comparing to other online libraries and algorithms on different datasets.

Conclusion and Recommendations

In this document, we presented our review of the hand exoskeleton in the literature, discussed prior works and designs and presented the configuration proposed in this project. Based on the design requirements and the study of the natural movement of the hand, the design proposed for the hand exoskeleton is composed of two closed-loop chains driving the flex/extension and abduction/adduction movements of the MCP joint, and the flex/extension movement of the PIP joint. The proposed model is composed of (P3RP2R) chain to control MCP f/e and a/a motion and four bar linkage which is (4R) kinematic chain to control PIP f/e. The study of the kinematics and dynamics of the hand exoskeleton’s two closed-loop chains was presented. Furthermore, the CAD model for the human hand was developed taken into account anatomy data from reported studies; and the CAD model for the exoskeleton was developed based on the kinematics manipulability analysis.

For the problem of hand pose estimation, our study included CPM, SHG, RTMPose, LiteHRNet and Mediapipe. The comparison was based on accuracy metrics that included AP, AR, and PCK and time-cost metrices like GFLOPs. Those metrics were benchmarks to choose RTMPose to integrate in the overall project. We concluded that RTMPose has the best performance achieving accurate results in real time.

For the next project, the study of the design will be continued with building the proper control system and power transmission using Bowden cables. Regarding the AI integration, the 2D hand pose estimation will be a sub-module for 3D hand pose estimation using a depth map alongside. This serves as a part of the forecasting model in the overall project.

References

[1] M. F. Levin, P. L. Weiss and E. A. Keshner, "Emergence of Virtual Reality as a Tool for Upper Limb Rehabilitation: Incorporation of Motor Control and Motor Learning Principles", Physical Therapy: Journal of the American Physical Therapy Association, vol. 95, 2015, pp. 415-425, DOI: 10.2522/ptj.20130579

[2] M. Sarac, M. Solazzi and A. Frisoli, "Design Requirements of Generic Hand Exoskeletons and Survey of Hand Exoskeletons for Rehabilitation, Assistive or Haptic Use",IEEE Transactions on Haptics PP(99):1-1, 2019, DOI:10.1109/TOH.2019.2924881

[3] A. Chiri, F. Giovacchini, N. Vitiello, E. Cattin, S. Roccella, F. Vecchi and M. Carrozza, HANDEXOS: Towards an exoskeleton device for the rehabilitation of the hand, IEEE/RSJ International Conference on Intelligent Robots and Systems, 2009, pp. 1106-1111, DOI: 10.1109/IROS.2009.5354376

[4] M. Sarac, M. Solazzi, E. Sotgiu, M. Bergamasco and A. Frisoli, Design and Kinematic Optimization of a Novel Underactuated Robotic Hand Exoskeleton, Mechanica: An International Journal of Theoretical and Applied Mechanics, Volume 52, pages 749–761, 2017, DOI: 10.1007/s11012-016-0530-z

[5] A. Wege, K. Kondak and G. Hommel, Mechanical design and motion control of a hand exoskeleton for rehabilitation, IEEE International Conference Mechatronics and Automation, 2005, Volume: 1. pp. 155-159. DOI: 10.1109/ICMA.2005.1626539, Print ISBN:0-7803-9044-X

[6] J. Li, R. Zheng, Y. Zhang and J. Yao, "iHandRehab: an Interactive Hand Exoskeleton for Active and Passive Rehabilitation", IEEE International Conference on Rehabilitation Robotics, 2011, DOI:10.1109/ICORR.2011.5975387, Online ISBN:978-1-4244-9861-1

[7] S. Cobos, M. Ferre, M. S. Uran, J. Ortego and C. Pena, "Efficient Human Hand Kinematics for Manipulation Tasks," IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 2246-2251, 2008 .

[8] F. Chen Chen, S. Appendino, A. Battezzato, A. Favetto, M. Mousavi and F. Pescarmona, Constraint study for a hand exoskeleton: Human hand kinematics and dynamics, Journal of Robotics, 2013. DOI:10.1155/2013/910961

[9] N. Sun, G. Li and L. Cheng, "Design and Validation of a Self-Aligning Index Finger Exoskeleton for Post-Stroke Rehabilitation", IEEE Transactions on Neural Systems and Rehabilitation Engineering, 2021:29:1513-1523. DOI: 10.1109/TNSRE.2021.3097888

[10] G. Li, L. Cheng and N. Sun, "Design, manipulability analysis and optimization of an index finger exoskeleton for stroke rehabilitation", Mechanism and Machine Theory, 2022, DOI:10.1016/j.mechmachtheory.2021.104526.

[11] V. Kishore Ayyadevara and Y. Reddy, Modern Computer Vision with PyTorch, Packt Publishing , 2020. ISBN 978-1-83921-347-2

[12] S.-E. Wei, V. Ramakrishna, T. Kanade and Y. Sheikh, Convolutional Pose Machines, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 4724-4732. DOI: 10.1109/CVPR.2016.511, Electronic ISSN: 1063-6919, Electronic ISBN:978-1-4673-8851-1

[13] A. Newell, K. Yang and J. Deng, Stacked Hourglass Networks for Human Pose Estimation, The European Conference on Computer Vision (ECCV), 2016, pp. 483-499. Online ISBN 978-3-319-46484-8, https://doi.org/10.48550/arXiv.1603.06937

[14] T. Jiang, P. Lu, L. Zhang, N. Ma, R. Han, C. Lyu, Y. Li and K. Chen, RTMPose: Real-Time Multi-Person Pose Estimation based on MMPose, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2023, https://doi.org/10.48550/arXiv.2303.07399

[15] C. Yu, B. Xiao, C. Gao, L. Yuan, L. Zhang, N. Sang and J. Wang, Lite-HRNet: A Lightweight High-Resolution Network, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2021, pp 10435-10445, DOI:10.1109/CVPR46437.2021.01030

[16] S. Raschka and V. Mirjalili, Python Machine Learning Second Edition, Packt Publishing, 2017, ISBN 978-1-78712-593-3

[17] Y. Lecun, L. Bottou, Y. Bengio and P. Haffner, Gradient-based learning applied to document recognition, Proceedings of the IEEE, vol. 86, 1998,

pp. 2278-2324, DOI: 10.1109/5.726791, Electronic ISSN: 1558-2256

[18] A. Krizhevsky, I. Sutskever and G. E. Hinton, ImageNet classification with deep convolutional neural networks, Communications of the ACM, vol. 60, 2017, pp. 84-90, doi/10.1145/3065386

[19] K. Simonyan and A. Zisserman, Very Deep Convolutional Networks for Large-Scale Image Recognition, Clinical Orthopaedics and Related Research (CORR), 2014, https://doi.org/10.48550/arXiv.1409.1556

[20] K. He, X. Zhang, S. Ren and J. Sun, "Deep Residual Learning for Image Recognition,", IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, DOI: 10.1109/CVPR.2016.90

[21] Y. Chen, H. Ma, D. Kong, X. Yan, J. Wu, W. Fan and X. Xie, Nonparametric Structure Regularization Machine for 2D Hand Pose Estimation, IEEE Winter Conference on Applications of Computer Vision (WACV), 2020, pp. 370-379, DOI:10.1109/WACV45572.2020.9093271

[22] J. Y. Chang, G. Moon and K. M. Lee, V2V-PoseNet: Voxel-to-Voxel Prediction Network for Accurate 3D Hand and Human Pose Estimation from a Single Depth Map, IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 5079-5088, DOI: 10.1109/CVPR.2018.00533, Electronic ISBN:978-1-5386-6420-9

[23] A. Toshev and C. Szegedy, DeepPose: Human Pose Estimation via Deep Neural Networks, IEEE Conference on Computer Vision and Pattern Recognition, 2014, Electronic ISBN:978-1-4799-5118-5, DOI: 10.1109/CVPR.2014.214

[24] P. F. Felzenszwalb and D. P. Huttenlocher, Pictorial Structures for Object Recognition, International Journal of Computer Vision, vol. 61, 2005, pp. 55-79, DOI:10.1023/B:VISI.0000042934.15159.49

[25] M. Fischler and R. Elschlager, The Representation and Matching of Pictorial Structures, IEEE Transactions on Computers, Vols. C-22, 1973, pp. 67-92, DOI: 10.1109/T-C.1973.223602

[26] S. R. Habib and N. N. Kamal, Stature estimation from hand and phalanges lengths of Egyptians, Journal of Forensic and Legal Medicine, vol. 17, 2010, pp. 156-160, DOI:10.1016/j.jflm.2009.12.004

[27] S. Roy, D. Sinwar, N. Dey, T. Perumal and J. M. R. S. Tavares, Eds., Innovations in Computational Intelligence and Computer Vision, ICICV: International Conference on Innovations in Computational Intelligence and Computer Vision, vol. 680, Singapore: Springer Nature Singapore, 2023, https://doi.org/10.1007/978-981-99-2602-2, eBook ISBN 978-981-99-2602-2

[28] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser and I. Polosukhin, Attention Is All You Need, 31st Conference on Neural Information Processing Systems (NIPS), vol.30, 2017, https://doi.org/10.48550/arXiv.1706.03762

[29] K. He, X. Zhang, S. Ren and J. Sun, Deep Residual Learning for Image Recognition, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp 770-778, DOI:10.1109/CVPR.2016.90

[30] M. Oberweger and V. Lepetit, DeepPrior++: Improving Fast and Accurate 3D Hand Pose Estimation, IEEE International Conference on Computer Vision and Pattern Recognition, 2017, pp. 585-594. DOI:10.1109/ICCVW.2017.75

[31] J. Chung, C. Gulcehre, K. Cho and Y. Bengio, Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling, Clinical Orthopaedics and Related Research (CORR), 2014, https://doi.org/10.48550/arXiv.1412.3555

[32] C. Zimmermann and T. Brox, Learning to Estimate 3D Hand Pose from Single RGB Images, IEEE International Conference on Computer Vision (ICCV), 2017, pp. 4913-4921, DOI:10.1109/ICCV.2017.525

[33] S. Hochreiter and J. Schmidhuber, Long Short-Term Memory, Neural Cmoputation, vol. 9, 1997, pp. 1735-1780, DOI:10.1162/neco.1997.9.8.1735

[34] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto and H. Adam, MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications, IEEE International Conference on Computer Vision and Pattern Recognition, 2017, https://doi.org/10.48550/arXiv.1704.04861

[35] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov and L.-C. Chen, MobileNetV2: Inverted Residuals and Linear Bottlenecks, IEEE

International Conference on Computer Vision and Pattern Recognition, pp 4510-4520, 2018, DOI: 10.1109/CVPR.2018.00474

[36] D. Bahdanau, K. Cho and Y. Bengio, Neural Machine Translation by Jointly Learning to Align and Translate, Clinical Orthopaedics and Related Research (CORR), 2014, https://doi.org/10.48550/arXiv.1409.0473

[37] K. Cho, B. van Merrienboer, D. Bahdanau and Y. Bengio, On the Properties of Neural Machine Translation: Encoder-Decoder Approaches, Solid State Science and Technology, 2014, DOI, 10.3115/v1/W14-4012

[38] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhutdinov, R. Zemel and Y. Bengio, Show, Attend and Tell: Neural Image Caption Generation with Visual Attention, International Conference on Machine Learning, 2015. vol. 37, pp 2048--2057, https://doi.org/10.48550/arXiv.1502.03044

[39] P. Rafael, N. Sergio Lima, da Silva. Edwardo A. B. , A Survey on Performance Metrics for Object-Detection Algorithms, Conference: International Conference on Systems, Signals and Image Processing (IWSSIP), 2020, pp 237-242, DOI:10.1109/IWSSIP48289.2020.9145130

[41] H. Georg and G. Wellein, Introduction to High Performance Computing for Scientists and Engineers, Boca Raton, FL: CRC Press, 2011.