DESIGN, MODELING AND SIMULATION OF ROBOTIC KRAR STRUMMER FOR ONE-HANDED INDIVIUALS

A Thesis

Submitted to EiT-M, School of Mechanical and Industrial Engineering in Partial Fulfillment of the Requirements for the Award of the

Degree of Masters of Science

In

Mechatronics Engineering

By

MR. ABRAHALEY KAHASY BRHANE

Supervisor of the thesis

Dr. RIESOM W/GERIMA

(Assoc. Prof.)

Co. Adviser of the thesis

Mr. NEBEYAT G/HIER (MSc)

INDUSTRIAL AUTOMATION AND CONTROL CHAIR

SCHOOL OF MECHANICAL AND INDUSTRIAL ENGINEERING

ETHIOPIAN INSTUTE OF TECHNOLOGY – MEKELLE

MEKELLE UNIVERSITY

MEKELLE, 2024

DECLARATION

Declaration of the Candidate

I hereby declare that this thesis entitled " Design, Modeling and Simulation of Robotic Krar Strummer for One-handed Individuals'' is the result of my own work, unless otherwise acknowledged. I have properly credited and cited all sources from which I have used data, ideas, or words, either verbatim or paraphrased. I also declare that this work has not been submitted for any other degree or award in any university or institution. I understand that any violation of academic integrity, including plagiarism or fabrication, will result in severe consequences, as outlined by the institutional legislation.

Name of the candidate: Abrahaley Kahsay Signature: _____Date: _____

Declaration of the Advisor

I certify that, I have supervised the research work presented in this thesis and throughout the duration of this study, I have provided guidance, feedback, and support to the candidate in conducting the research, analyzing the data, and writing the thesis. I confirm that the thesis is adequate for the award of master of science in Mechatronics Engineering.

Name of the candidate: Dr. Riesom W/gerima Signature: _____Date: ____

THESIS ACCCEPTANCE APPROVAL FORM

This is to certify that Mr. Abrahaley Kahsay has incorporated all comments forwarded by the external and internal examiners as well as by the chairperson during the thesis defense held in ____/11/2024.

Members of the Examination Board

Dr. Riesom		29/03/2025
Supervisor	signature	date
Dr. Kinde Anlay		29/03/2025
External examiner	sign	date
Mussie W/gebriel		29/03/2025
Chairman	Signature	Date

Confirmation: Industrial Automation and Control Chair

Mr. Nebyat Gebregziabhier (MSc.)		29/03/2025
Chair head	Signature	Date

ABSTRACT

Playing traditional musical instruments, such as the Krar, typically requires two hands, which limits accessibility for individuals with upper-limb disabilities. This thesis addresses this challenge by designing, modeling, and simulating a robotic Krar strummer to assist one-handed individuals in playing the Krar, a traditional string instrument from Ethiopia and Eritrea. Despite advances in assistive technology, few solutions exist for traditional instruments, leaving a gap in accessibility for one-handed musicians. A robotic manipulator was developed to replicate the strumming motion of a human hand, enabling users with one functional hand to engage with the instrument. A three-dimensional model of the robotic strummer mechanism was created using SolidWorks, and its dynamic behavior and control system were simulated in MATLAB/Simulink. Rhythmic trajectories based on traditional Krar performances were analyzed, and a PID controller was implemented to ensure precise strumming patterns. Key findings from the simulations show that the robotic strummer accurately replicates rhythmic patterns with minimal error. The system demonstrated high accuracy in following predefined trajectories, with errors well within acceptable limits set for RKS.

Joint space trajectory errors were minimal, with Joint 1 showing a maximum deviation of 0.091 degrees and Joint 2 a maximum deviation of 1.56 degrees without disturbance. With disturbance present, Joint 1 showed a maximum deviation of 1.7 degrees and Joint 2 a maximum deviation of 3 degrees. The controller effectively maintained precise alignment with the desired trajectories, ensuring stable operation. The simulation results under disturbance also demonstrated the system's ability to maintain stability and accuracy, with errors considered insignificant in terms of the rhythmic pattern. This research contributes to enhancing accessibility, enabling one-handed individuals to participate in music creation, promoting inclusivity in musical expression. The detailed design, modeling, and simulation results confirm the feasibility of the proposed robotic Krar strummer system. This research represents a step forward in assistive technology, bridging the gap between traditional music instrument and accessibility for individuals with disabilities.

Keywords: RKS, Upper-limb Disabilities, SolidWorks, MATLAB/Simulink, Rhythmic Trajectories, PID Controller, Musical Accessibility

ACKNOWLEDGEMENTS

I want to begin by expressing my deepest gratitude to Life for firm stay throughout my journey, for guiding me to this point. My sincere thanks go to my supervisor, Cap. Dr. Riessom Weldegiorgis (Assoc. Prof.) and Mr. Nebyat G/biher (MSc.), for their guidance, advice, and constant motivation throughout my thesis and entire MSc. studies. I extend my appreciation to Mekelle University, the Ethiopian Institute of Technology-Mekelle's School of Mechanical and Industrial Engineering, and the Industrial Automation and Control Chair for providing me with the opportunity to pursue my MSc. program and for their continued support. Finally, my deepest thanks go to my family and friends, whose unwavering support has been instrumental in bringing me to where I stand today.

Thank you,

Abrahaley Kahsay

TABLE OF CONTENT

LIST OF FIGURES

LIST OF TABLES

SYMBOL AND ABBREVIATION

COG	Center of gravity
COM	Center of mass
DOF	degree of freedom
DH	Denavit-Hartenberg
α	Link twist angle
θ	Joint angle
θ_{ijk}	$\theta_i + \theta_j + \theta_k$
ω	angular velocity vector
τ	torque of an actuator
B	body coordinate frame
C	cosine
S	Sine
RKS	robotic Krar strummer

CHAPTER ONE

INTRODUCTION

1.1. BACKGROUND

Music holds profound cultural and social significance across the globe, often serving as a means of expression, communication, and connection. In many cultures, traditional musical instruments carry the heritage of a community and play a pivotal role in preserving and transmitting cultural identity. Among such instruments is the Krar, a string instrument native to northern Ethiopia and Eritrea. The Krar is integral to cultural events and modern music within the region, characterized by its distinctive sound and rhythmic patterns that have enriched the musical landscape for centuries [1].

The Krar is typically an acoustic or electrically amplified instrument with five or six strings, played on a pentatonic scale. Its construction features a hollowed-out rectangular wooden resonating box covered with a wooden sheet. The Krar primarily accompanies solo performances, where the player often also sings. The songs played typically consist of multiple melodically and rhythmically distinct sections, arranged cyclically, mimicking the song melody and complementing the singer's breathing gaps. This cyclical arrangement allows for a rich interplay between vocal and instrumental elements, making the Krar an essential component of Ethiopian musical traditions [2].

Despite its cultural significance, individuals with upper-limb disabilities face substantial challenges in accessing and engaging with the Krar. Playing this instrument typically requires coordinated use of both hands to pluck the strings and produce melodies. For individuals with only one functional hand, engaging with the Krar can be prohibitively difficult, thereby limiting their ability to participate fully in cultural and musical activities [2].

Recognizing the importance of inclusivity in music, researchers and engineers have increasingly turned to technological solutions to address these challenges. Robotic technology offers promising avenues for creating assistive devices in various domains, including music. Robotic arms can be programmed to perform complex motions that replicate the actions needed to play string instruments. However, the application of robotics to string instruments like the Krar remains relatively unexplored. However, there is limited scope for instruments like the Krar, which require a unique combination of strumming and fingerpicking techniques [3].

Developing a robotic Krar strummer specifically designed for one-handed individuals would represent a significant contribution to assistive technology and music accessibility. This thesis

addresses this challenge by designing, modeling, and simulating a robotic Krar strummer tailored for one-handed individuals. By merging mechanical design principles with control engineering techniques and music analysis, this research aims to create a virtual prototype that empowers one-handed individuals to play the Krar effectively [4][5].

In addition to enhancing accessibility for musicians with disabilities, this research endeavors to promote greater inclusivity in musical expression. By harnessing modern design tools such as SolidWorks for three-dimensional modeling and MATLAB/Simulink for dynamic behavior simulation, this thesis seeks to bridge the gap between technology and creativity. The outcomes hold promise in advancing assistive technologies for musicians with upper-limb disabilities while fostering an environment where all individuals can engage in making regardless of physical limitations [6].

The Krar's unique playing techniques require specific design adaptations for robotic systems. Unlike guitars, which primarily involve plucking or simple strumming, the Krar combines these techniques with complex rhythmic patterns that often require precise hand movements, posing a challenge for a one-handed musician. Existing robotic string instruments typically focus on straightforward strumming or picking patterns, which do not accommodate the Krar's intricacies. Thus, a key design focus for this thesis is to develop a strumming mechanism capable of capturing both the rhythmic and tonal distinctions specific to the Krar.

1.2. PROBLEM STATEMENT

A significant number of individuals with upper-limb disabilities, especially those with only one functioning hand, face considerable obstacles in playing traditional string instruments like the Krar. The Krar’s unique sound and complex rhythmic patterns require the coordinated use of both hands to pluck its strings and produce melodies, making it inaccessible to one-handed individuals. This exclusion not only impacts their personal well-being but also reduces the richness of cultural expression within their communities. The Krar’s elaborate playing techniques demand precise work from both hands, creating substantial challenges for one-handed individuals and limiting their participation in cultural and musical traditions.

To address these challenges, this research proposes the development of a robotic Krar strummer, designed to empower one-handed individuals to overcome physical limitations and achieve their musical aspirations. Such innovative solution holds the potential to enhance their quality of life, foster social inclusion, and support overall well-being [7][8].

1.3. OBJECTIVES

1.3.1. General objective:

The objective of this study is to design, model, and simulate a robotic Krar strummer that can assist a one-hand disabled individuals.

1.3.2. Specific objectives:

The specific objectives of this thesis are

- To design and model detail physical model of the robotic Krar Strummer
- To develop the kinematic and dynamic model of the robotic Krar Strummer
- To formulate a rhythmic trajectory for the robotic Krar strummer
- To control the robotic strummer using PID-controller on MATLAB/SIMULINK environment

1.4. SCOPE

This thesis investigates the design, modelling, and simulation of a robotic Krar strummer to assist one-handed individuals in playing the Krar, a traditional stringed instrument. The study encompasses kinematics and dynamics modelling to develop a mathematical model of the strummer's movements, which supports accurate simulation and control. A user-friendly design for one-handed use will be created in SolidWorks, with a 3D model integrated into MATLAB/Simulink for verification, while rhythmic trajectory generation translate the Krar's rhythms into movement patterns. which to be managed by a PID control system in Simulink to ensure precise strumming. This thesis excludes hardware development and physical prototype. Advanced control techniques like adaptive control or neural networks, as well as complex musical expressions such as fingerpicking or vibrato, are set aside for possible future enhancements.

The design, modelling, and simulation of the robotic Krar strummer's kinematic behavior and control are the central concerns of this research. It is important to note that, within the scope of this work, the dynamic state response and drive system dynamics are not considered. This delimitation was necessary to maintain focus and manage the complexity of the thesis.

1.5. THESIS OUTLINE

This thesis presents the design, modelling, and simulation of a Robotic Krar Strummer (RKS). The RKS aims to emulate the strumming technique of Krar. To establish a theoretical foundation, Chapter 2 reviews existing literature on robotic string instrument strummers and analyses the unique challenges posed by the Krar's playing style. Chapter 3 researches into the development of the RKS's

physical model, including its kinematic and dynamic mathematical representation. Additionally, rhythmic trajectories were generated using MATLAB/Simulink by extracting time beats from Krar music recordings by the help of Audacity. Chapter 4 presents the results of the RKS simulation, evaluating its performance in terms of accuracy, stability, and naturalness. Finally, Chapter 5 concludes the study, summarizing key findings, discussing limitations, and providing recommendations for future research and improvements in the RKS.

CHAPTER-TWO

LITERATURE REVIEW

2.1. ACCESSIBILITY AND ASSISTIVE TECHNOLOGY IN MUSIC

Traditional musical instruments often pose significant challenges for individuals with upper-limb disabilities, as many require precise hand coordination. Instruments like the Krar, which rely on intricate strumming and fingerpicking, are particularly difficult for those with limited mobility in one hand. Limited access to these instruments not only restricts creative expression but also prevents participation in cultural traditions, which are essential for identity and community belonging.

Assistive technologies in music have made strides in recent years, particularly with adaptive modifications for instruments such as pianos and guitars, using custom grips and simplified controls. While such innovations have enhanced musical inclusivity, stringed instruments like the Krar remain underexplored in terms of adaptive technology. Robotic assistive devices could address this gap by providing fully or semi-automated control over certain aspects of performance, enabling disabled musicians to participate with minimal physical effort. By developing a robotic strumming device specifically for the Krar, this research aims to create new opportunities for one-handed musicians to experience and express the unique rhythmic qualities of the instrument, thus bridging an important accessibility gap. Research by Prabuwono et al.[9] emphasizes the importance of user-centered design in assistive robotics, supporting the adaptation of traditional instruments to better accommodate disabled musicians.

2.2. ROBOTIC MUSICAL INSTRUMENT STRUMMERS

The field of robotic musical instrument strummers has a rich history, rooted in early electromechanical innovations aimed at replicating human actions such as strumming or tapping. These early devices were typically designed for experimental music performances or research, offering initial insights into how robotics could assist in musical performance. A significant milestone in this area was the development of the GuitarBot by Eric Singer in the early 2000s, which showcased how robotic devices could achieve precise and repeatable strumming actions, bridging technical gaps in musical robotics [10].

Since then, technological advancements have allowed for more complex and responsive strumming devices across various instruments. Modern robotic strummers in guitars and ukuleles, for example, now incorporate actuators, sensors, and control systems that replicate the action of human strumming. Many of these systems employ digital signal processing to synchronize with live music, allowing the robotic strummers to complement human musicians. This reveals the potential for robotic strummers to inclusivity in music, though much of the innovation has focused on popular Western instruments, highlighting a need for devices tailored to traditional or regional instruments like the Krar[10].

Despite these advancements, robotic strummers face significant challenges in usability and accessibility. Many systems struggle with real-time responsiveness, which is essential for achieving expressive strumming patterns, and high production costs limit access for musicians with disabilities. Designing intuitive control interfaces for diverse user needs also remains challenging. This research project addresses these limitations by developing an adaptive robotic strummer for the Krar, focusing on affordability, ease of use, and adaptability to meet the instrument's unique demands.

The GuitarBot offers a foundation for control and synchronization in a robotic Krar strummer, yet the adaptation of such systems for non-guitar instruments remains relatively unexplored. Insights from Moreno-Valenzuela's [11] work on trajectory tracking for robot manipulators, along with control strategies from literature[12] on adaptive control and Nakahara et al.[13]. On tempo-tracking, could enhance the precision needed for expressive, real-time strumming on the Krar. Adaptive control could ensure that the robotic strummer adjusts speed and pressure dynamically, supporting the human expressiveness required in traditional music.

2.3. DESIGN AND CONTROL OF ROBOTIC STRUMMING MECHANISMS

Kinematic modeling plays a crucial role in designing robotic strumming mechanisms, as it governs the movement and interaction of each component in the robotic arm relative to the Krar strings. Forward and inverse kinematic models enable precise control over the robotic arm’s position and orientation, ensuring accurate string strikes and the replication of the Krar’s unique playing style. These models also facilitate the calculation of necessary movements for each strum or pluck, improving the device’s effectiveness in producing the desired musical output. Additionally, the use of kinematic models allows for adjustments to the strumming technique, enabling the device to adapt to different rhythmic patterns based on the musician's preference [20][21].

Control strategies are essential for translating the robotic device's kinematic model into rhythmic strumming actions. Basic PID control methods are widely used to regulate movement speed and precision, ensuring that each strum aligns with the desired rhythm. These methods allow for real-time adjustments based on rhythm and tempo changes, making the robotic strummer adaptable to various playing styles. MATLAB/Simulink serves as a robust simulation environment in this research, where control strategies can be fine-tuned before physical implementation [23]. Research by Chen and Naidu [14] on adaptive control and literature [15] on tempo-tracking control strategies offer valuable insights into methods that could benefit real-time, responsive strumming.

The rhythmic trajectory of the Krar involves periodic motion and requires precise positional accuracy, making a PID controller a favorable choice. The Proportional (P) control offers immediate corrections, while the Integral (I) control eliminates rhythmic drift over time. The Derivative (D) control smooths the controller's response during rhythmic transitions, minimizing overshoot and oscillations. The PID controller is recommended as it provides comprehensive control over the dynamic and periodic behavior of the robotic Krar strummer, ensuring accurate tracking of the rhythmic trajectory.

2.4. KINEMATIC AND DYNAMIC MODELING OF ROBOTIC SYSTEMS IN MUSIC

Kinematic and dynamic modeling are essential for creating robotic strummers that can replicate the precise movements required for musical expression. Kinematic modeling focuses on calculating the position and orientation of each robotic arm component based on desired strumming patterns. For a non-planar, multi-degree-of-freedom (DOF) robotic arm, forward and inverse kinematics are crucial to manage complex, three-dimensional movements across different axes, enabling the strummer to position itself accurately relative to the Krar's strings. Dynamic modeling further builds on these principles by factoring in inertia, velocity, and acceleration, accounting for the physical forces acting on the robotic strummer, such as gravitational and centrifugal forces. This allows for stable, controlled movements necessary to replicate the Krar's rhythmic strumming, where varying speeds and intensities are key to producing different tones [20][22].

In musical applications, dynamic modeling helps maintain consistent performance by compensating for force variations and managing joint dynamics. This ensures reliable sound quality and timing. Real-world implementations of robotic strummers, particularly in experimental music, demonstrate that dynamic modeling significantly enhances the replication of natural musical expressions. For a robotic Krar strummer, dynamic modeling ensures the device can handle rhythmic patterns with accuracy and adaptability, making it a valuable tool for one-handed musicians. Research by Handžić [16] and Reed on variable tension in string instruments, alongside Flor et al.[3] on educational robotics design, can inform

kinematic models tailored to the Krar, supporting the complex strumming and plucking techniques that define the instrument.

2.5. GAPS, OPPORTUNITIES, AND FUTURE DIRECTIONS

Current research in robotic strummers highlights significant gaps, especially in developing affordable, accessible devices for traditional instruments like the Krar. Although numerous robotic systems exist for popular Western instruments, limited work addresses the cultural and technical needs of traditional African instruments, which often require specialized playing techniques. Additionally, existing robotic strummers generally lack the adaptability for real-time control and are cost-prohibitive, making them inaccessible to many musicians with disabilities. Addressing these issues could significantly broaden the impact of robotic assistive devices across diverse musical traditions and user demographics [3][16].

Advances in machine learning and adaptive control systems present promising avenues to overcome these challenges. Machine learning could enable a robotic strummer to learn and replicate complex strumming patterns specific to the Krar, adjusting dynamically based on musician input. Sensors that provide real-time feedback on tempo, intensity, and rhythm accuracy could further enhance device responsiveness, creating a more interactive and expressive experience. Future research should focus on physical prototyping and user testing, especially with one-handed musicians, to refine the device's design and usability. Expanding the strumming mechanism to control multiple strings could add versatility, enabling more complex melodies and harmonies. Studies such as Hou et al.[17] on adaptive gripping and Gopinath and Weinberg's [18]. work on robotic expressivity suggest that integrating these features could help close the gap between human and robotic performances, promoting inclusivity and accessibility in music.

CHAPTER THREE

METHODOLOGY

This section details the methodology used to develop a robotic Krar strummer for one-handed individuals. The aim is to create a strumming pattern that can be useful for individuals with upper limb disability to play the Krar.

3.1. DESIGN AND MODELLING OF ROBOTIC KRAR STRUMMER

The robotic Krar strummer is initially modeled in SolidWorks, a 3D CAD software, to accurately represent its physical structure and dimensions. This 3D model is then exported to MATLAB/Simulink, where it is integrated with the control system and simulated in a virtual environment. To enable the robotic Krar to produce realistic and rhythmic strumming patterns, three samples of rhythmic audio were collected and imported into Audacity for beat time extraction. This analysis provides crucial data on the timing and tempo of the desired strumming patterns. Using the extracted beat times, a third-order polynomial rhythmic trajectory were developed in MATLAB/Simulink, defining the motion of the strumming mechanism to ensure it follows the rhythm precisely. A PID-control setup is then implemented in MATLAB/Simulink to control the robotic Krar's movements, continuously monitoring its position and adjusting its motion to maintain accurate and precise strumming as shown for software configuration and system simulation in Figure 3-1.

.

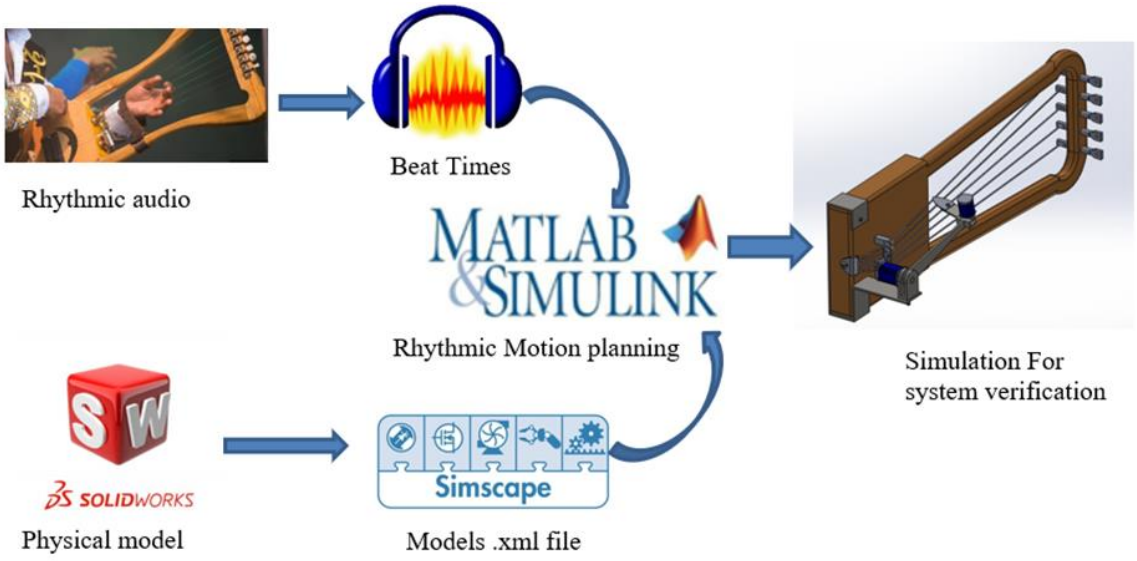


Figure 3-1: software configuration for system simulation

3.1.1. Design of robotic manipulator

The robot is a two-link, non-planar, two-degrees-of-freedom (DOF) robotic manipulator, as shown in Figure 3-2. This design is chosen for two key reasons: first, it provides comfort for the player, and second, it enables a one-handed player to achieve the required strumming speed and rhythm pattern. These features make the robot well-suited for replicating the intricate movements needed for accurate musical performance.

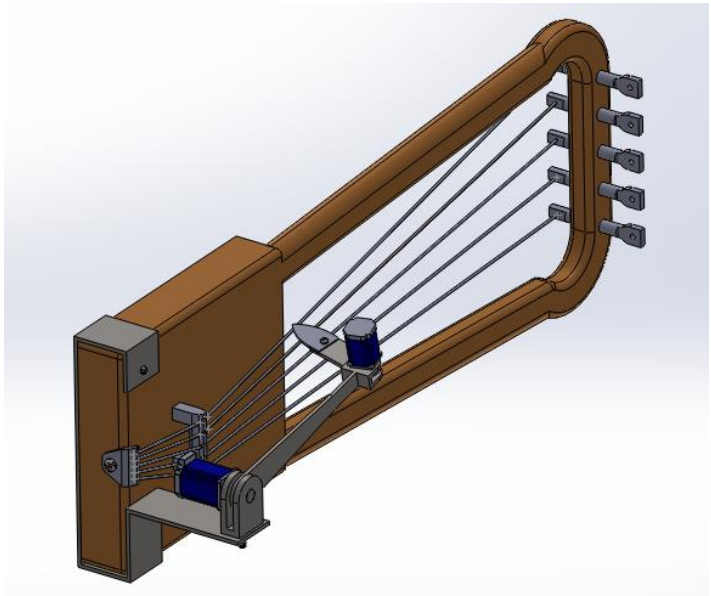


Figure 3-2: model of the robot and Krar on Solid Works

3.1.1. Robotic manipulator specification

The robotic manipulator is a two-link non-planar system designed with two degrees of freedom to effectively replicate the strumming action needed for playing the Krar. Link 1 measures 160 mm and Link 2 measures 80 mm, providing optimal reach and control. Each link's mass and moment of inertia are thoughtfully chosen to ensure a balance between stability and lightweight operation, facilitating ease of use for one-handed individuals.

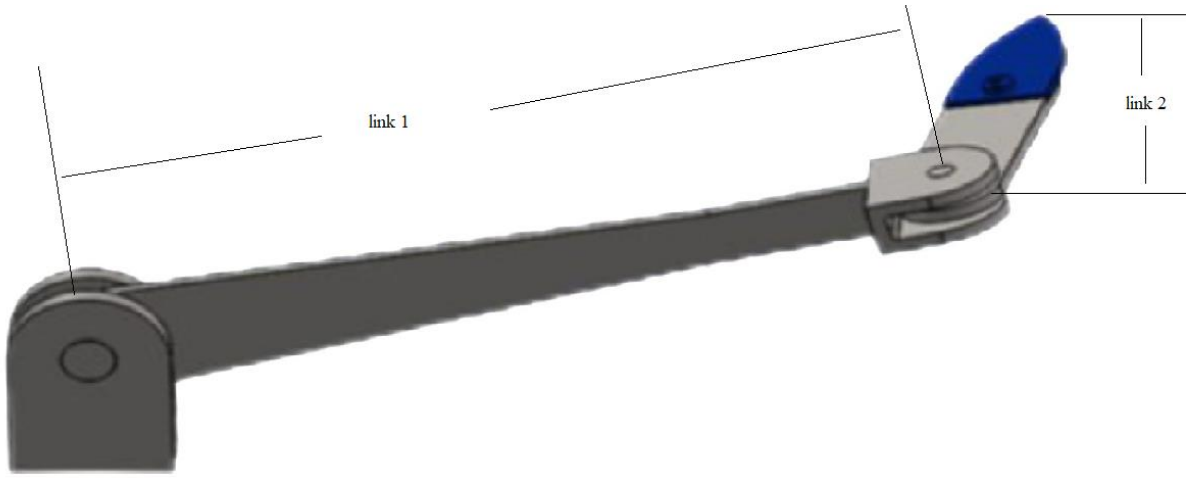


Figure 3-3 two-link non-planar robotic manipulator

As shown below in the table, are the specifications of the robotic manipulator.

Table 3-1 RKS specification

Link	Length (mm)	Mass (g)	Length from attached to the center of mass(mm)
1	160	85.32	12.35
2	80	40.15	28.15

3.2. KINEMATIC MODELING OF ROBOTIC KRAR STRUMMER

This section outlines the kinematic modeling of the robotic Krar strummer (RKS), focusing on how the robot's movements are controlled. It covers forward kinematics to determine the end-effector's position from joint angles and inverse kinematics to find joint angles for a specific position. The use of Denavit-Hartenberg parameters helps in replicating the necessary strumming actions for the Krar.

3.2.1. Forward kinematics

Forward kinematics determines the position and orientation of the end-effector of the robotic Krar strummer for a given set of joint angles (θ_1 and θ_2). By employing the Denavit-Hartenberg (DH) [1] convention, the equations are derived that relate joint angles to the end-effector's coordinates in three-dimensional space. This analysis is crucial for verifying the performance of the closed-loop control system, as it allows for accurate calculation of the robotic arm's movements necessary for executing the desired strumming actions on the Krar.

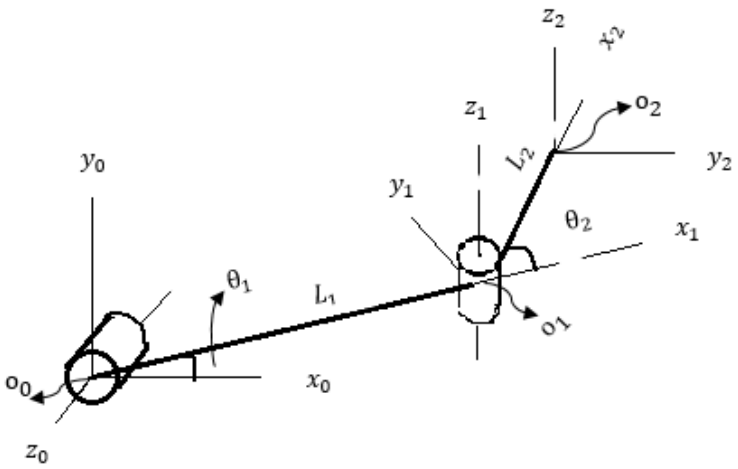


Figure 3-4 DH, two-link non-planar robotic manipulator

The Denavit-Hartenberg (DH) parameters for the two-link non-planar robotic manipulator utilized in the design of the robotic Krar strummer are outlined in Table 3-2. These parameters include link lengths (L), link twists (α), link offsets (d), and joint angles (θ), all of which are essential for establishing the kinematic relationships between the different links of the manipulator. The accurate definition of these parameters is crucial for computing the forward kinematics, thereby enabling precise positioning and orientation of the end-effector in relation to the Krar strings. Specifically, L_1 and L_2 represent the lengths of link-1 and link-2, respectively, while θ_1 and θ_2 denote the angular positions of link-1 and link-2, respectively.

Table 3-2 link DH parameters for two-link non-planar robotic manipulator

link	L_i	α_i	d_i	θ_i	0ffset
1	L_1	-90	0	θ_1	20
2	L_2	0	0	θ_2	90

$$T_i^{i-1}=\begin{bmatrix} c\theta_i & -s\theta_i c\alpha_i & s\theta_i s\alpha_i & L_i c\theta_i \\ s\theta_i & c\theta_i c\alpha_i & -c\theta_i s\alpha_i & L_i s\theta_i \\ 0 & s\alpha_i & c\alpha_i & L_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad 3\text{-}1$$

Where, T_i^{i-1} is a homogenous transformation matrix of frame "i-1" relative to frame "i" according to the DH-algorithm. Define $s_1=\sin(\theta_1+20)$, $c_1=\cos(\theta_1+20)$, $s_2=\sin(\theta_2+90)$, $c_2=\cos(\theta_2+90)$, o_0= the origin frame, o_1= frame-1 and o_2= farme-2

Substituting DH-parameters from table 2.1 into equation 2.1 gives:

$$T_1^0=\begin{bmatrix} c_1 & 0 & -s_1 & L_1 c_1 \\ s_1 & 0 & c_1 & L_1 s_1 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad 3\text{-}2$$

Position and orientation of the frame-1 relative to the origin.

$$T_2^1=\begin{bmatrix} c_2 & -s_2 & 0 & L_2 c_2 \\ s_2 & c_2 & 0 & L_2 s_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad 3\text{-}3$$

$$T_2^{0\prime} = T_1^0 * T_2^1=\begin{bmatrix} c_1 & 0 & -s_1 & l_1 c_1 \\ s_1 & 0 & c_1 & l_1 s_1 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} c_2 & -s_2 & 0 & l_2 c_2 \\ s_2 & c_2 & 0 & l_2 s_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_2^{0\prime}=\begin{bmatrix} c_1 c_2 & -c_1 s_2 & -s_1 & c_1 c_2 l_2 + c_1 l_1 \\ c_2 s_1 & -s_1 s_2 & c_1 & c_2 l_2 s_1 + l_1 s_1 \\ -s_2 & -c_2 & 0 & -l_2 s_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_2^0=R_x\ (\text{-}90)\ T_2^{0\prime}$$

Where, R_x (θx) = $\begin{bmatrix} 1 & 0 & 0 \\ 0 & c\theta x & -s\theta x \\ 0 & s\theta x & c\theta x \end{bmatrix}$

$$T_2^0 = \begin{bmatrix} c_1c_2 & -c_1s_2 & -s_1 & c_1c_2l_2 + c_1l_1 \\ l_2s_2 & -c_2 & 0 & -l_2s_2 \\ c_2s_1 & -s_1s_2 & c_1 & c_2l_2s_1 + l_1s_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3\text{-}4)$$

T_2^0 is position and orientation of the frame-2 relative to the origin. Therefore,

$$x_2 = L_1c_1 + L_2c_1c_2 \quad (3\text{-}5)$$

$$y_2 = -L_2s_2 \quad (3\text{-}6)$$

$$z_2 = L_1s_1 + L_2s_1c_2 \quad (3\text{-}7)$$

3.2.2. Inverse kinematics

Inverse kinematics involves determining the joint angles required to position the end-effector of the robotic Krar strummer at a desired location and orientation. Inverse kinematics is the opposite problem, where we aim to find the required joint angles (θ_1 and θ_2) to achieve a desired end-effector position (X, Y, Z) based on the rhythmic trajectory. Due to the non-planar nature of the manipulator and potential for multiple solutions, solving the inverse kinematics analytically might be challenging.

Taking front and top view of figure 2.3 to solve inverse kinematics of the robotic manipulator using graphical method leads to:

Taking front view of figure 3-3 or 3-4 for calculating "θ_1"

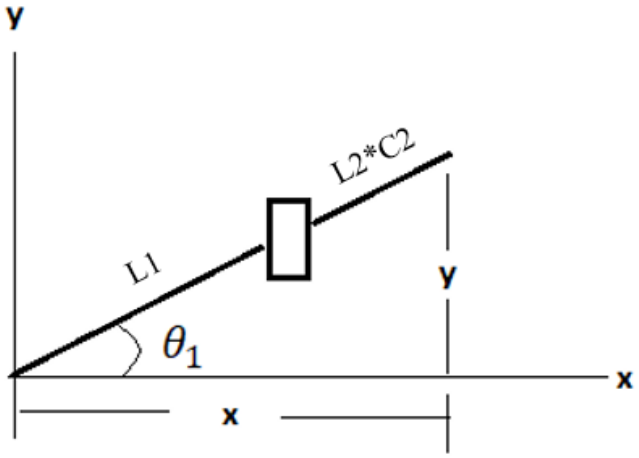


Figure 3-5: front view of RKS

$$\theta_1 = \left(\tan^{-1}\left(\frac{y}{x}\right)\right) - 20$$

Taking top view of Figure 3-3 or 3.4 for calculating " θ_2"

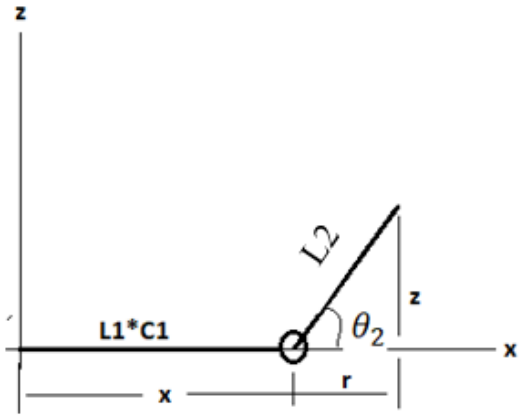


Figure 3-6: Top view of RKS

$$r=(x - L1 * C1)$$

$$\theta_2 = (\tan^{-1}(\frac{z}{r}))\text{-}90$$

3.3.3. Velocity kinematics

Velocity kinematics analyzes the relationship between the joint velocities of the robotic Krar strummer and the velocity of its end-effector. By deriving the Jacobian matrix, this analysis enables precise control during strumming actions, allowing for the optimization of the manipulator's performance. Understanding these dynamics is crucial for achieving the responsive and accurate execution of desired strumming patterns in musical performance.

$$\dot{X}=[J]\dot{\theta} \qquad 3\text{-}4$$

Where,

$\dot{X}$, is the linear velocity of the end effector or frame-2 relative to the origin

$\dot{\theta}$, is the angular velocity of the joints

$[J]$, is manipulator Jacobian matrix (Jacobian) which relates the linear velocity of the end effector with joint angular velocity

$$\dot{X}=\begin{Bmatrix}x_2\\y_2\\z_2\end{Bmatrix},\ [J]=[J_1 \quad J_2] \text{ and } \dot{\theta}=\begin{Bmatrix}\dot{\theta}_1\\\dot{\theta}_2\end{Bmatrix}$$

The Jacobian matrix for an n-links of a robotic manipulator is of the form[1]:

$$J_i=[J_1 \quad J_2 \dots\dots\dots \quad J_n]$$

Where: the i^{th} column of J_i is given by:

$$J_i=\begin{bmatrix}z_{i-1}\times(o_n-o_{i-1})\\z_{i-1}\end{bmatrix} \text{ if the joint is revolute}$$

$$J_i=\begin{bmatrix}z_{i-1}\\0\end{bmatrix} \text{ if the joint is prismatic}$$

The robot type is a two-link non-planar robotic manipulator, and because of this reason, Jacobian has two terms. Which are:

$$[J]=[J_1 \quad J_2] \qquad 3\text{-}5$$

Where,

$$J_1=\begin{bmatrix}z_0\times(o_2-o_0)\\z_0\end{bmatrix} \qquad 3\text{-}6$$

$$J_2=\begin{bmatrix}z_1\times(o_2-o_1)\\z_1\end{bmatrix} \qquad 3\text{-}7$$

z_i are the first three elements in the third column of the homogenous matrix T_i^0

o_i are the first three elements in the fourth column of the homogenous matrix T_i^0

Now, to calculate J_1 and J_2

$$J_1=\begin{bmatrix}z_0\times(o_2-o_0)\\z_0\end{bmatrix} \qquad 3\text{-}8$$

From equation 3-3 and 3-4 which are T_1^0 and T_2^0

$$z_0=\begin{bmatrix}0\\0\\1\end{bmatrix},\ z_1=\begin{bmatrix}s_1\\-c_1\\0\end{bmatrix}\ o_0=\begin{bmatrix}0\\0\\0\end{bmatrix},\ o_1=\begin{bmatrix}L_1c_1\\L_1s_1\\0\end{bmatrix} \text{ and } o_2=\begin{bmatrix}L_1c_1+L_2c_1c_2\\L_1s_1+L_2s_1c_2\\L_2s_2\end{bmatrix} \qquad 3\text{-}9$$

Substituting z_0 , z_1, o_1 and $o_2$values into equation 3-6 and 3-7 gives:

$$J_1=\begin{bmatrix} z_0 \times (o_2 - o_0) \\ z_0 \end{bmatrix}=\begin{bmatrix} \begin{bmatrix}0\\0\\1\end{bmatrix} \times \left(\begin{bmatrix} L_1c_1 + L_2c_1c_2 \\ L_1s_1 + L_2s_1c_2 \\ L_2s_2 \end{bmatrix} - \begin{bmatrix}0\\0\\0\end{bmatrix} \right) \\ \begin{bmatrix}0\\0\\1\end{bmatrix} \end{bmatrix}$$

$$J_1=\begin{bmatrix} -L_1s_1 - L_2s_1c_2 \\ L_1c_1 + L_2c_1c_2 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \qquad 3\text{-}10$$

$$J_2=\begin{bmatrix} z_1 \times (o_2 - o_1) \\ z_1 \end{bmatrix}=\begin{bmatrix} \begin{bmatrix}s_1\\-c_1\\0\end{bmatrix} \times \left(\begin{bmatrix} L_1c_1 + L_2c_1c_2 \\ L_1s_1 + L_2s_1c_2 \\ L_2s_2 \end{bmatrix} - \begin{bmatrix}L_1c_1\\L_1s_1\\0\end{bmatrix} \right) \\ \begin{bmatrix}s_1\\-c_1\\0\end{bmatrix} \end{bmatrix}$$

$$J_2=\begin{bmatrix} -L_2c_1s_2 \\ -L_2s_1s_2 \\ L_2c_2 \\ s_1 \\ -c_1 \\ 0 \end{bmatrix} \qquad 3\text{-}11$$

Where, J_1 and J_2 Jacobian of link-1 and link-2 respectively.

Then, substituting J_1 and J_2 into equation 3-5 gives the Jacobian manipulator:

$$J=\begin{bmatrix} -L_1s_1 - L_2s_1c_2 & -L_2c_1s_2 \\ L_1c_1 + L_2c_1c_2 & -L_2s_1s_2 \\ 0 & L_2c_2 \\ 0 & s_1 \\ 0 & -c_1 \\ 1 & 0 \end{bmatrix} \qquad 3\text{-}12$$

Substituting $[J]$ back into equation 3-4 gives the relationship between the linear velocity of the end-effector and angular velocity of the joints.

$$\begin{Bmatrix} x_2 \\ y_2 \\ z_2 \end{Bmatrix} = \begin{bmatrix} -L_1 s_1 - L_2 s_1 c_2 & -L_2 c_1 s_2 \\ L_1 c_1 + L_2 c_1 c_2 & -L_2 s_1 s_2 \\ 0 & L_2 c_2 \\ 0 & s_1 \\ 0 & -c_1 \\ 1 & 0 \end{bmatrix} \begin{Bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{Bmatrix} \qquad 3\text{-}13$$

The first three rows of the Jacobian matrix are linear velocity of the origin "o_2" relative to the base" o_0".

Taking the third row of equation (3-11)

$z_2 = L_2 c_2 \dot{\theta}_2$

Taking the first two rows of equation (3-11):

$$\begin{Bmatrix} \dot{x_2} \\ \dot{y_2} \end{Bmatrix} = \begin{bmatrix} -L_1 s_1 - L_2 s_1 c_2 & -L_2 c_1 s_2 \\ L_1 c_1 + L_2 c_1 c_2 & -L_2 s_1 s_2 \end{bmatrix} \begin{Bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{Bmatrix} \qquad 3\text{-}14$$

Where, $\dot{x}_2$and $\dot{y}_2$ are the linear velocity of the end effector (tip of link-2) relative to the origin.

$$\dot{\theta} = [J]^{-1} \dot{X}$$

Where, $[J]^{-1}$ is the inverse of the Jacobian matrix. considering equation 3-14, to use the advantage of the Jacobian matrix that when linear velocity of the end-effector (tip of link-2) is known, it is possible to calculate the angular velocity of the joints using the inverse of the Jacobian matrix $[J_{12}]^{-1}$.

$$\begin{Bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{Bmatrix} = [J_{12}]^{-1} \begin{Bmatrix} \dot{x_2} \\ \dot{y_2} \end{Bmatrix} \qquad 3\text{-}15$$

Where, J_{12} from equation 3-14 is equal to:

$$[J_{12}] = \begin{bmatrix} -L_1 s_1 - L_2 s_1 c_2 & -L_2 c_1 s_2 \\ L_1 c_1 + L_2 c_1 c_2 & -L_2 s_1 s_2 \end{bmatrix} \qquad 3\text{-}16$$

$$[J_{12}]^{-1} = \left(\frac{1}{L_1 L_2 s_2 + {L_2}^2 c_2 s_2}\right) \begin{bmatrix} -L_2 s_1 s_2 & L_2 c_1 s_2 \\ -L_1 c_1 - L_2 c_1 c_2 & -L_1 s_1 - L_2 s_1 c_2 \end{bmatrix}$$

$$\begin{Bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{Bmatrix} = \left(\frac{1}{L_1 L_2 s_2 + {L_2}^2 c_2 s_2}\right) \begin{bmatrix} -L_2 s_1 s_2 & L_2 c_1 s_2 \\ -L_1 c_1 - L_2 c_1 c_2 & -L_1 s_1 - L_2 s_1 c_2 \end{bmatrix} \begin{Bmatrix} \dot{x_2} \\ \dot{y_2} \end{Bmatrix} \qquad 3\text{-}17$$

Where, $\left(\frac{1}{L_1 L_2 s_2 + {L_2}^2 c_2 s_2}\right)$ is determinant of $[J_{12}]$

3.3. DYNAMICS OF THE ROBOTIC MANIPULATOR

The total kinetic and potential energy of robotic manipulator using DH-joint variables as generalized coordinates is given by [19][23]:

$$K_i = \frac{1}{2}\dot{q}^T \sum_i^n [m_i J_{vi}^T J_{vi} + J_{wi}^T I_i J_{wi}]\dot{q} \qquad 3\text{-}18$$

$$K_T = K_1 + K_2 \qquad 3\text{-}19$$

K_T is total kinetic energy

$P_i = m_i g h_i(q)$

Where,

K_i is the kinetic energy of link-i

P_i is the potential energy of link-i

$\dot{q} = \begin{Bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{Bmatrix}$ are joints variables

m_i is mass of link-i

J_{vi} is upper half Jacobian (linear Jacobian matrix)

J_{wi} is lower half Jacobian (angular Jacobian matrix)

I_i is mass moment of inertia

g is the gravitational acceleration

h_i vertical height from the centroid of link-i relative to the base

$$K_i = \frac{1}{2}\dot{q}^T \sum_i^n [m_i J_{vi}^T J_{vi} + J_{wi}^T I_i J_{wi}]\dot{q}$$

$$K_i = \frac{1}{2}\dot{q}^T D(q)\dot{q}$$

Where, $D(q) = [m_i J_{vi}^T J_{vi} + J_{wi}^T I_i J_{wi}]$ is called inertia matrix

$$K_1 = \frac{1}{2}\dot{q}^T D(q)_1(q)\dot{q}$$

$$D(q)_1 = [m_1 J_{v1}^T J_{v1} + J_{w1}^T I_1 J_{w1}] \qquad 3\text{-}20$$

$$K_2 = \frac{1}{2}\dot{q}^T D(q)_2(q)\dot{q}$$

$$D(q)_2 = [m_2 J_{v2}^T J_{v2} + J_{w2}^T I_2 J_{w2}] \qquad 3\text{-}21$$

$$D(q)=\sum(D(q)_1 + D(q)_2) \qquad 3\text{-}22$$

$$D(q)= [m_1 J_{v1}^T J_{v1} + J_{w1}^T I_1 J_{w1}]+[m_2 J_{v2}^T J_{v2} + J_{w2}^T I_2 J_{w2}]$$

$$D(q)= [m_1 J_{v1}^T J_{v1} + J_{w1}^T I_1 J_{w1} + m_2 J_{v2}^T J_{v2} + J_{w2}^T I_2 J_{w2}] \qquad 3\text{-}23$$

To calculate $D(q)_1$ and $D(q)_2$, first J_{vi} and J_{wi} has to be calculated.

$$J_i=\begin{bmatrix} z_{i-1} \times (o_n - o_{i-1}) \\ z_{i-1} \end{bmatrix}$$

$$J_{v1}=[{J_{v1}}^1 \; {J_{v1}}^2]$$

$${J_{v1}}^1= \begin{bmatrix} z_0 \times (o_1 - o_0) \\ z_0 \end{bmatrix}$$

$${J_{v1}}^1= \begin{bmatrix} z_0 \times (o_1 - o_0) \\ z_0 \end{bmatrix}$$

$${J_{v1}}^2= \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \qquad 3\text{-}24$$

Then, from the homogeneous matrix T_1^0 and T_2^0

$$z_0=\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix},\ z_1=\begin{bmatrix} \mathrm{s}_1 \\ -\mathrm{c}_1 \\ 0 \end{bmatrix}\ o_0=\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix},\ o_1=\begin{bmatrix} L_1\mathrm{c}_1 \\ L_1\mathrm{s}_1 \\ 0 \end{bmatrix} \text{ and } o_2=\begin{bmatrix} L_1\mathrm{c}_1 + L_2\mathrm{c}_1\mathrm{c}_2 \\ L_1\mathrm{s}_1 + L_2\mathrm{s}_1\mathrm{c}_2 \\ L_2\mathrm{s}_2 \end{bmatrix} \qquad 3\text{-}25$$

Substituting the values of z_0, o_1 and o_0 in to J_{v1} gives:

$$J_{v1}=\begin{bmatrix} -L_1 s_1 & 0 \\ L_1 c_1 & 0 \\ 0 & 0 \end{bmatrix}$$

$$J_{w1}= [{J_{w1}}^1 \; {J_{w1}}^2]$$

$J_{wi}= z_{i-1}$ for revolute joint

$${J_{w1}}^1= z_0= \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

$${J_{w1}}^2= \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \qquad 3\text{-}26$$

$$J_{w1}=\begin{bmatrix}0 & 0\\ 0 & 0\\ 1 & 0\end{bmatrix} \qquad 3\text{-}27$$

Where,

J_{v2} is equal to the first three rows of Jacobian manipulator [J] of equation 3-12:

$$J_{v2}=\begin{bmatrix}-L_1 s_1 - L_2 s_1 c_2 & -L_2 c_1 s_2\\ L_1 c_1 + L_2 c_1 c_2 & -L_2 s_1 s_2\\ 0 & L_2 c_2\end{bmatrix} \qquad 3\text{-}28$$

J_{w2} is equal to the last three rows of Jacobian manipulator [J] of equation 3-12:

$$J_{w2}=\begin{bmatrix}0 & s_1\\ 0 & -c_1\\ 1 & 0\end{bmatrix} \qquad 3\text{-}29$$

Substituting J_{v1} and J_{w1}into $D(q)_1$ gives:

$$D(q)_1=\begin{bmatrix}m1{L_1}^2 + I_1 & 0\\ 0 & 0\end{bmatrix} \qquad 3\text{-}30$$

Substituting J_{v2} and J_{w2}into $D(q)_2$ gives:

$$D(q)=\begin{bmatrix}m_2(L_1 + L_2 c_2)^2 + I_2 & 0\\ 0 & m_2{L_2}^2 + I_2\end{bmatrix} \qquad 3\text{-}31$$

Substituting $D(q)_1$ and $D(q)_2$ into equation 2.15 to get $D(q)$:

$$D(q)_2\begin{bmatrix}m1{L_1}^2 + m_2(L_1 + L_2 c_2)^2 + I_2 + I_1 & 0\\ 0 & m_2{L_2}^2 + I_2\end{bmatrix} \qquad 3\text{-}32$$

$$D(q)=\begin{bmatrix}d_{11} & d_{12}\\ d_{21} & d_{22}\end{bmatrix}$$

From equation 3-18:

$$d_{11}= m1{L_1}^2 + m_2(L_1 + L_2 c_2)^2 + I_2 + I_1$$

$$d_{21}= d_{12}= 0$$

$$d_{22}= m_2{L_2}^2 + I_2$$

I_1 and I_2 are inertia tensor of robotic manipulator relative to the attached frame.

The inertia tensor at the center of mass of each link is given by:

$$I_i = \begin{bmatrix} I_{xxi} & 0 & 0 \\ 0 & I_{yyi} & 0 \\ 0 & 0 & Izz_i \end{bmatrix} \qquad (3\text{-}33)$$

I_i is inertia tensor of i^{th}-link at the center of mass when the mass distribution of the body is symmetric with respect to the body attached frame.

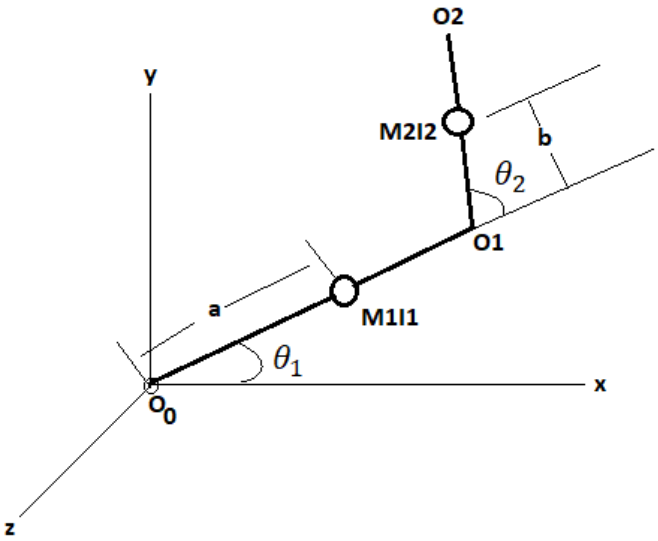


Figure 3-7: point mass of the manipulator at its center of mass

Where, a and b are length from the attached frame to the center of mass of link-1 and link-2 respectively.

$$I_i = Izz_i = m_i(x_i^2 + y_i^2) = m_i(r_i^2)$$

$$I_1 = m_1(r_1^2) \text{ where, } r_1 = a$$

$$I_2 = m_2(r_2^2) \text{ where, } r_2 = \text{b}$$

Substituting a and b, into I_1 and I_2:

$$I_1 = m_1 a^2 \qquad (3\text{-}34)$$

$$I_2 = m_2 \text{b}^2 \qquad (3\text{-}35)$$

Calculating potential energy of the links:

$$\text{P}_i = m_i g h_i(q)$$

$$P_1 = m_1 g h_1(q) \quad (3\text{-}36)$$

$$P_2 = m_2 g h_2(q) \quad (3\text{-}37)$$

From the forward kinematics:

$$h_1(q) = a \times s_1$$

$$h_2(q) = L_1 s_1 + b c_2 s_1$$

$$P_t = P_1 + P_2 \quad (3\text{-}38)$$

Substituting $h_1(q)$ and $h_2(q)$ into equation 3-19 and 3-20 respectively gives P_t.

$$P_t = a m_1 g s_1 + b m_2 g (L_1 s_1 + b c_2 s_1)$$

Where, P_t is total potential energy

$$P_t = (a m_1 + L_1 m_2) g s_1 + b m_2 g c_2 s_1$$

$$\phi_k = \frac{\partial P_t}{\partial q_k}$$

$$\phi_1 = \frac{\partial P_t}{\partial \theta_1} = (a m_1 + L_1 m_2) g c_1 + b m_2 g c_2 c_1 \quad (3\text{-}39)$$

$$\phi_2 = \frac{\partial P_t}{\partial \theta_2} = -b m_2 g s_2 s_1) \quad (3\text{-}40)$$

Using Euler-Lagrange formulation:

$\mathcal{L}_i = K_i - P_i$ where, $\mathcal{L}_i$ is Lagrangian of the robotic manipulator

$$\mathcal{L}_i = K_i - P_i = \frac{1}{2} \dot{q}^T D(q) \dot{q} - P(q) \quad (3\text{-}41)$$

$$\mathcal{L}_i = \frac{1}{2} \dot{q}^T D(q) \dot{q} -- P(q)$$

$$\mathcal{L}_i = \frac{1}{2} \sum_{i,j}^{n} d_{ij}(q)\, \dot{q}_i \dot{q}_j - P(q)$$

$$\tau_i = \frac{d}{dt} \left(\frac{\partial \mathcal{L}_i}{\partial q_i'}\right) - \frac{\partial \mathcal{L}_i}{\partial q_i}$$

where, τ_i denotes Euler- Lagrangian dynamic equation which is torque or force (for prismatic joint) applied on the i^{th} joint.

$K_i = \frac{1}{2}\dot{q}^T D(q)\dot{q} = \frac{1}{2}\sum_{i,j}^{n} d_{ij}(q)\,\dot{q}_i\dot{q}_j$ where, d_{ij} is an element of inertia matrix D(q) and K_i denotes kinetic energy of i^{th} link.

$$\tau_k = \sum d_{kj}(q)\ddot{q}_j + \sum_{ij} C_{ijk}(q)\dot{q}_i\dot{q}_j + \phi_k \qquad 3\text{-}42$$

Where,

$\phi_k = \frac{\partial P_t}{\partial q_k}$

$C_{ijk}(q) = \frac{1}{2}\left\{\frac{d_{kj}}{\partial q_i} + \frac{\partial d_{ki}}{\partial q_j} - \frac{\partial d_{ij}}{\partial q_k}\right\}$ is called Christoffel symbols

When the inertia matrix is diagonal and independent of "q_i" (constant) all of $C_{ijk}(q)$ are zero.

$$C_{111} = \frac{1}{2}\left\{\frac{d_{11}}{\partial\theta_1} + \frac{\partial d_{11}}{\partial\theta_1} - \frac{\partial d_{11}}{\partial\theta_1}\right\} = 0$$

$$C_{121} = C_{211} = \frac{1}{2}\frac{d_{11}}{\partial\theta_2} = -m_2 s_2(L_1 L_2 + {L_2}^2 c_2)$$

$$C_{221} = 0$$

$$C_{112} = -\frac{1}{2}\frac{d_{11}}{\partial\theta_2} = m_2 s_2(L_1 L_2 + {L_2}^2 c_2)$$

$$C_{122} = C_{212} = C_{222} = 0$$

$$\tau_k = \sum d_{kj}(q)\ddot{q}_j + \sum_{ij} C_{ijk}(q)\dot{q}_i\dot{q}_j + \phi_k$$

$$\tau_1 = d_{11}\ddot{\theta}_1 + d_{12}\ddot{\theta}_2 + C_{121}\ddot{\theta}_1\ddot{\theta}_2 + C_{211}\ddot{\theta}_1\ddot{\theta}_2 + \phi_1 \qquad 3\text{-}43$$

$$\tau_2 = d_{21}\ddot{\theta}_1 + d_{22}\ddot{\theta}_2 + C_{112}\ddot{\theta}_1\ddot{\theta}_2 + \phi_2 \qquad 3\text{-}44$$

Writing the dynamic equations in a matrix form:

$$\begin{bmatrix}\tau_1\\ \tau_2\end{bmatrix} = \begin{bmatrix}d_{11} & d_{12}\\ d_{21} & d_{22}\end{bmatrix}\begin{Bmatrix}\ddot{\theta}_1\\ \ddot{\theta}_2\end{Bmatrix} + \begin{bmatrix}-2C_{121}\dot{\theta}_2 & 0\\ C_{112}\dot{\theta}_1 & 0\end{bmatrix}\begin{Bmatrix}\dot{\theta}_1\\ \dot{\theta}_2\end{Bmatrix} + \begin{bmatrix}\phi_1\\ \phi_2\end{bmatrix} \qquad 3\text{-}45$$

$$\begin{Bmatrix}\tau_1\\ \tau_2\end{Bmatrix} = [D(\theta_i)]\begin{Bmatrix}\ddot{\theta}_1\\ \ddot{\theta}_2\end{Bmatrix} + C(\theta,\dot{\theta})\begin{Bmatrix}\dot{\theta}_1\\ \dot{\theta}_2\end{Bmatrix} + \begin{Bmatrix}\phi_1\\ \phi_2\end{Bmatrix} \qquad 3\text{-}46$$

Where:

$$[D(\theta_i)] = \begin{bmatrix}d_{11} & d_{12}\\ d_{21} & d_{22}\end{bmatrix} \qquad 3\text{-}47$$

$$C(\theta,\dot{\theta}) = \begin{bmatrix} -2C_{121}\dot{\theta}_2 & 0 \\ C_{112}\dot{\theta}_1 & 0 \end{bmatrix} \qquad 3\text{-}48$$

To accurately replicate the physical behaviour of the manipulator, the dynamic model is virtually prototyped in SOLIDWORKS and integrated into MATLAB/Simulink through the Simscape Multibody plug-in, as illustrated Figure 3-8. This integration allows for a more realistic simulation of the manipulator's dynamics within the virtual environment.

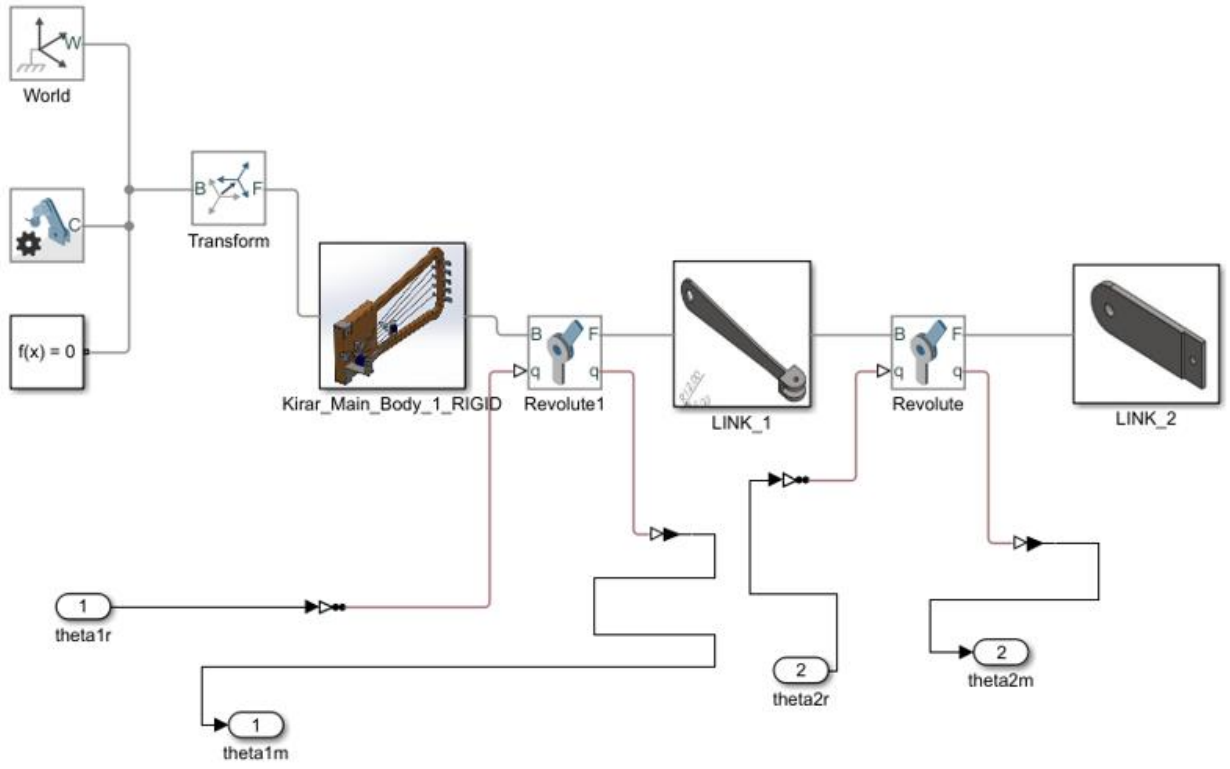


Figure 3-8: Dynamic model of 2-DOF non-planner robotic manipulator (RKS)

3.4. CONTROLLER DESIGN AND SIMULATION SETUP

3.4.1. Joint limitations of RKS

In this system, joint one is constrained to move within a range of -20 degrees to 20 degrees, ensuring that its motion remains within these limits to facilitate the formation of the desired rhythmic trajectory, as represented in the top-view labeled as θ_{1a}. Joint two is similarly restricted, with its movement confined to a range of 60 degrees to 90 degrees, ensuring it does not exceed these angles during the trajectory formation process, as also labeled as θ_{2a}. The home position of joint one is set at $\theta_1 = 20$ degrees, while the home position of joint two is set as $\theta_2 = 90$ degrees labeled as θ_{1h} and θ_{2h} as shown in figure 3-11

and figure 3-12 respectively. These joint limitations are critical for achieving the intended rhythmic motion of the RKS.

3.4.2. Joint requirements of RKS

The Krar string width, labeled as "Sw" in the schematic of joint one shown in Figure 3-11, has a width of 80 mm. To achieve the desired rhythmic strumming pattern, joint one must have a minimum positional range of (-14.5,14.5) degrees labeled as θ_{1r} . Similarly, joint two must maintain a minimum positional requirement of 85.5 degrees to 90 degrees labeled as θ_{2r} in Figure 3-12. The maximum allowable deviation for joint one is 5.5 degrees, calculated as the difference between its maximum limit (20 degrees) and the minimum required position (14.5 degrees). For joint two, the maximum allowable error is 3.5 degrees, determined by the difference between its maximum limit (90 degrees) and the minimum required position (85.5 degrees). These constraints are essential for ensuring accurate rhythmic strumming within the system.

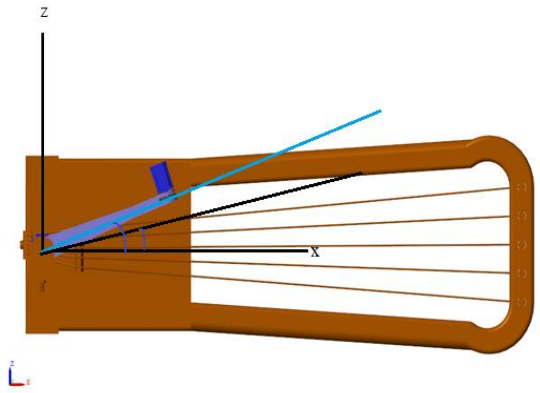


Figure 3-9: home position RKS from front view

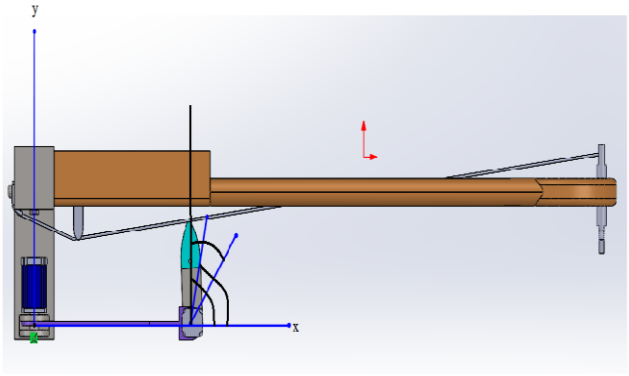


Figure 3-10: home position RKS from top view

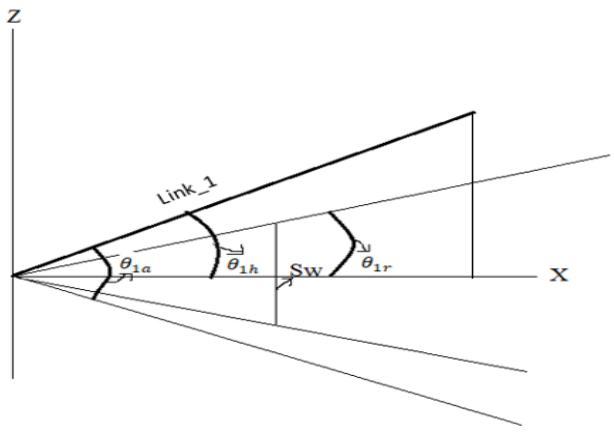


Figure 3-11: schematic of joint_1 from the front view

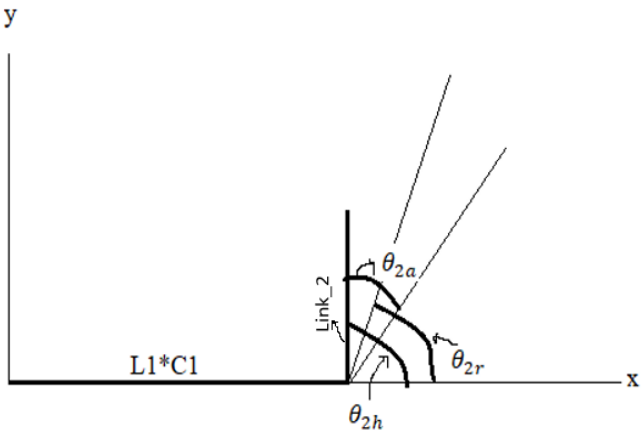


Figure 3-12: schematic of joint_2 from the top view

3.4.3. Developing rhythmic trajectory

Three samples of Krar rhythms were selected and analyzed to develop rhythmic trajectories for a robotic strummer. Each of the selected rhythms had a constant tempo, and their beat times were precisely analyzed using Audacity, a tool for audio editing and signal analysis. The strumming pattern from each sample, including the timing of up and down strum, was extracted to determine key moments for strumming actions. These moments were mapped to specific waypoints in the trajectory.

To generate the rhythmic trajectories in MATLAB/Simulink, six waypoints were chosen, each with known angles and corresponding end-effector coordinates (pick) for the robotic manipulator. These waypoints were carefully aligned with the beat times derived from the audio analysis. Three different cubic polynomial trajectories were then generated, each corresponding to the rhythmic patterns of the Krar audio samples. By using the beat times as the temporal references for the waypoints, the rhythmic trajectories were designed to replicate the strumming pattern, ensuring that the robotic strummer's movement followed the audio rhythm with precision. The result was a set of cubic rhythmic trajectories that used to control the end-effector of the robotic manipulator in, allowing it to strum the Krar in sync with the selected rhythmic patterns. Here in Table 3-3, are the way points with their corresponding times:

Table 3-3: way points with their corresponding joint angles

Way points	Joint angles (Deg.)		End effector position (way points) in mm		
	θ_1	θ_2	x	y	z
1	0	0	150.4	-80	54.72
2	-20	0	160	80	0
3	-40	0	150.4	-80	-54.72
4	-40	-30	187.9	-69.28	-68.4
5	-20	-30	200	-69.28	0
6	0	-30	187.9	-80	68.4

3.4.4. Rhythm beat times Extraction and mapping to waypoints

The Krar strings are muted using the one hand of the Krar player to isolate the percussive sound. This method minimizes interference from string vibrations while capturing the rhythmic pattern only. The recording environment was controlled with minimal background noise to ensure accurate audio data.

The captured audio files are imported into Audacity, a free and open-source audio editing software. Audacity's spectrogram and visual inspection of the rhythmic audio wave view is used to visually identify the strong rhythmic transients (peaks) corresponding to the Krar strums. By placing markers at these transients, the software will automatically calculate the inter-onset intervals (IOIs), which represent the time between each strum.

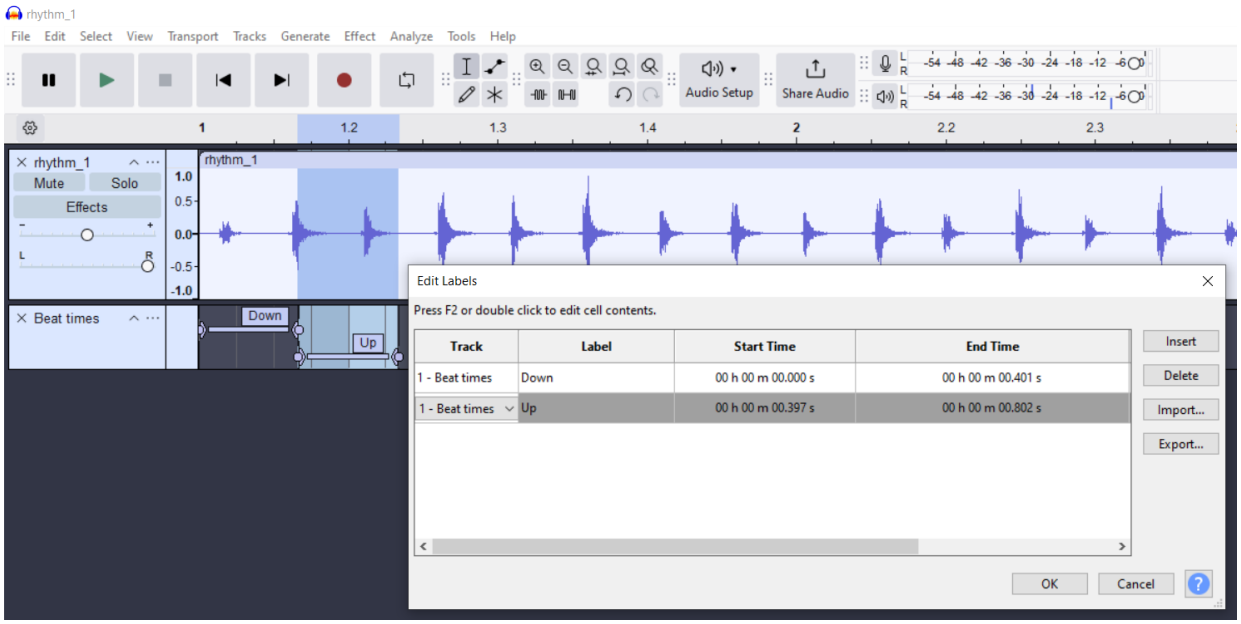


Figure 3-13: Beat times of rhythm one using Audacity

As presented in the figure 3-13, the down strum takes 0.4 seconds, while the up strum also requires 0.4 seconds, resulting in a total time of 0.8 seconds to complete the entire down-up pattern.

Table 3-4: mapping waypoints with beat times to develop rhythmic trajectory one

points	Rhythmic pattern	Joint angles		End effector position (way points) (mm)			Way Times(sec)
		θ_1	θ_2	x	y	z	
1	Down	0	0	150.4	-80	54.72	0
2		-20	0	160	80	0	0.2
3		-40	0	150.4	-80	-54.72	0.4
4	Up	-20	0	160	-80	0	0.6
5		0	0	150.4	-80	54.72	0.8

The beat times, which were extracted from the rhythmic audio using Audacity, are aligned with the specified waypoints to generate the down-up strumming pattern, as shown in the Table 3-4.

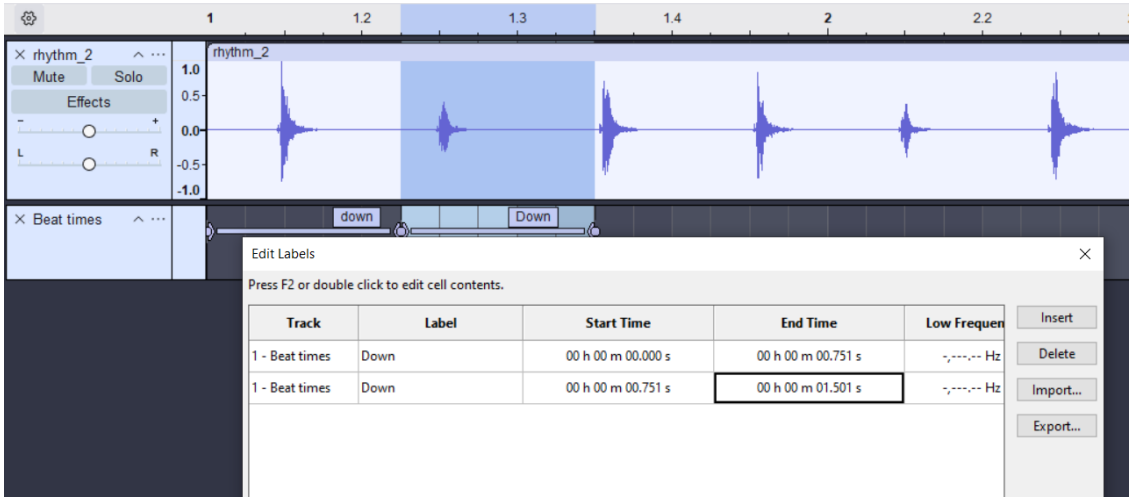


Figure 3-14: Beat times of rhythm two from Audacity

As illustrated in the Figure 3-14, the time required for each down strum in the down-down pattern is 0.75 seconds, resulting in a total time of 1.5 seconds to complete the entire pattern. The beat times extracted from the rhythmic audio using Audacity are aligned with the waypoints to generate the down-down strumming pattern.

Table 3-5: mapping waypoints with beat times to develop rhythmic trajectory two

points	Rhythmic pattern	Joint angles (Deg)		End effector position (mm)			Way Times (sec)
		θ_1	θ_2	x	y	z	
1		0	0	150.4	-80	54.72	0
2		-20	0	160	80	0	0.25
3	Down	-40	0	150.4	-80	-54.72	0.75
4		-40	-30	187.9	-69.28	-68.4	0.8
5	Down	-20	-30	200	-69.28	0	0.98
6		0	-30	187.9	-80	68.4	1.28
7		0	0	150.4	-80	54.72	1.5

The beat times, extracted from the rhythmic audio using Audacity, are aligned with the waypoints to form the down-down pattern, as detailed in the Table 3-5.

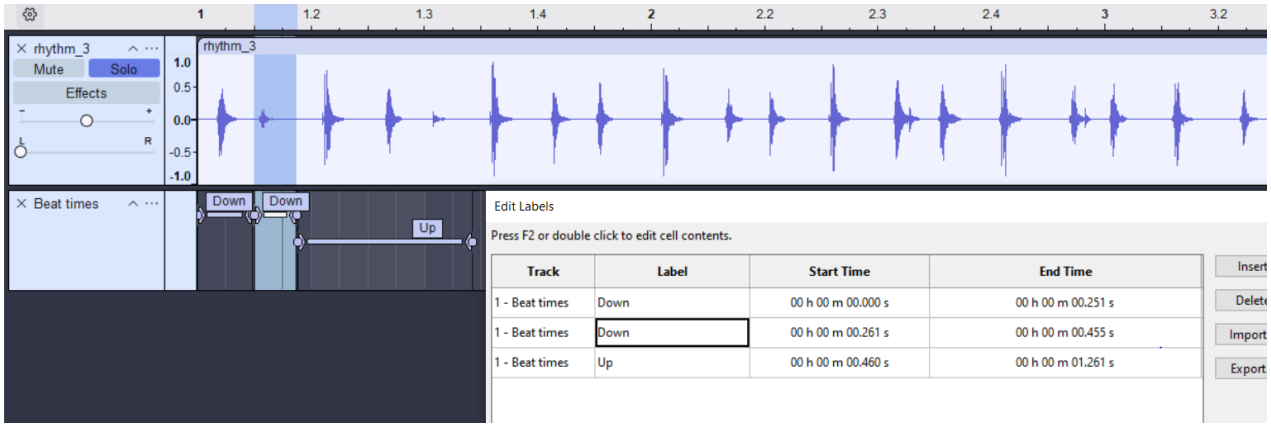


Figure 3-15: Beat times of rhythm two three Audacity

As illustrated in the Figure 3-15, the first down strum occurs at 0.25 seconds, the second down strum is completed at 0.455 seconds, and the up strum finishes at 1.26 seconds. This sequence defines one cycle, which spans from 0 to 1.26 seconds. The beat times, extracted from the rhythmic audio using Audacity, are aligned with the waypoints to form the down-down-up pattern, as detailed in the Table 3-6. Using the waypoints and the corresponding beat times (point times), a rhythmic trajectory is developed in MATLAB/Simulink.

Table 3-6: mapping waypoints with beat times to develop rhythmic trajectory three

Way points	Rhythmic pattern	Joint angles		End effector position (mm)			Way Times
		θ_1	θ_2	x	y	z	
1		0	0	150.4	-80	54.72	0
2		-20	0	160	80	0	0.12
3	Down	-40	0	150.4	-80	-54.72	0.25
4		-40	-30	187.9	-69.28	-68.4	0.36
5		-20	-30	200	-69.28	0	0.48
6		0	-30	187.9	-80	68.4	0.6
7	Down	0	0	150.4	-80	54.72	0.72
8		-20	0	160	80	0	0.84
9		-40	0	150.4	-80	-54.72	0.96
10	Up	-20	0	160	-80	0	1.12
11		0	0	150.4	-80	54.72	1.26

3.4.5. Rhythmic Trajectory Planning

Polynomial trajectories were developed to represent the movement of the robotic manipulator during strumming. Polynomial functions offer a balance between smooth motion control and ease of mathematical manipulation. A third-order polynomial is used to represent desired strumming patterns between points. The number of strings is strummed simultaneously and variations in speed strumming is considered when defining the trajectory.

The extracted beat times from each Krar rhythm audio is used as a time reference among waypoints for the polynomial trajectory as mapped in Table 3-4, Table 3-5, and Table 3-6. These waypoints and, the precise timing and duration of each strum define the rhythmic structure. This ensures the robotic strummer accurately replicates the chosen Krar rhythms.

MATLAB/SIMULINK is used to develop (generate) a rhythmic trajectory using the time data extracted and the waypoints mapped. The Rhythm Selector block is utilized to enable the user to select one of the three sample rhythms. The Multiport Switch is employed to switch the output based on the input from the Rhythm Selector, ensuring that the desired rhythm is accurately reflected in the system's output. As illustrated in Figure 3-16, these blocks collectively facilitate the development and selection of the desired rhythmic trajectory. The complete MATLAB code supporting this implementation is provided in Appendix A.

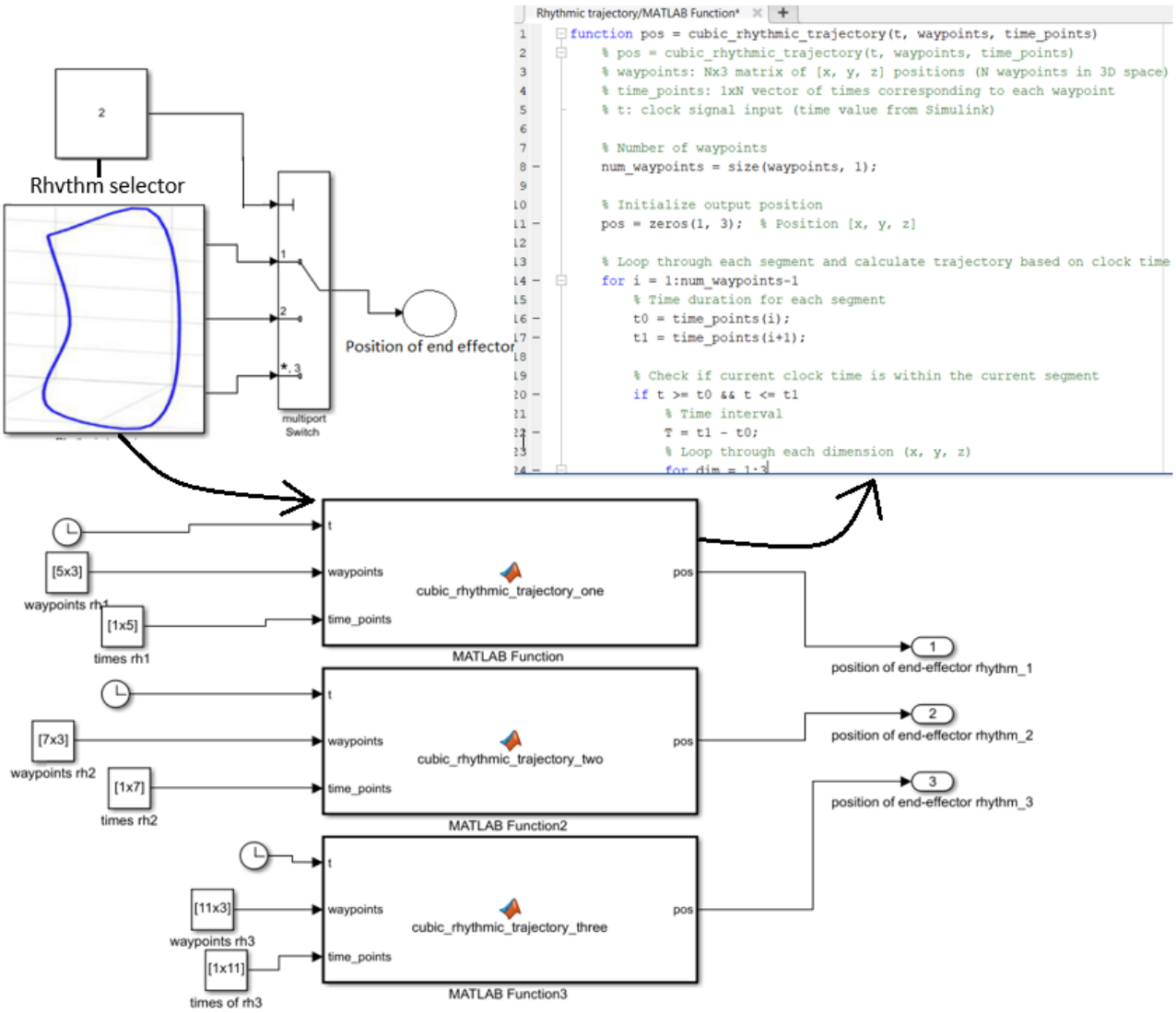


Figure 3-16: Rhythmic trajectory input generator Matlab/Simulink block

The rhythmic trajectory input is shown in Figure 3-17, Figure 3-18 and Figure 3-19 were developed using the MATLAB/Simulink block explained in figure 3-16, using the waypoints and extracted beat times mapped Table 3-4, Table 3-5, and Table 3-6 respectively.

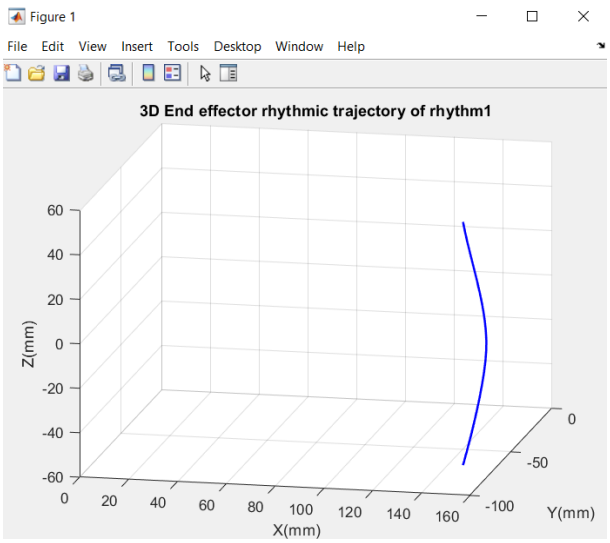


Figure 3-17: Rhythmic trajectory of rhythm one

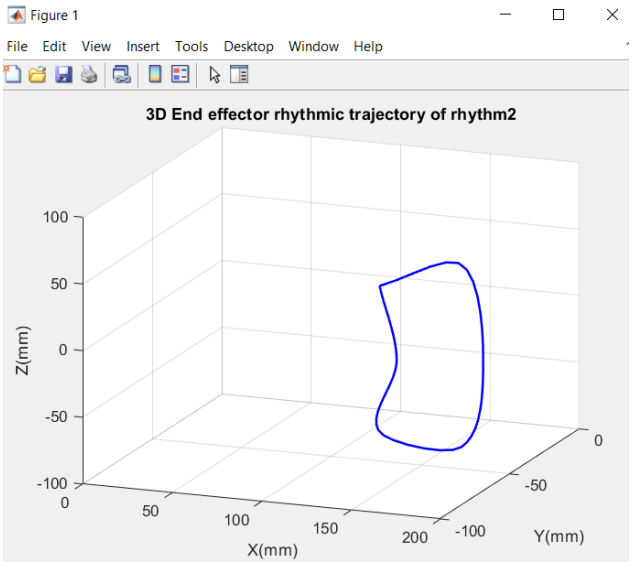


Figure 3-18: Rhythmic trajectory of rhythm two

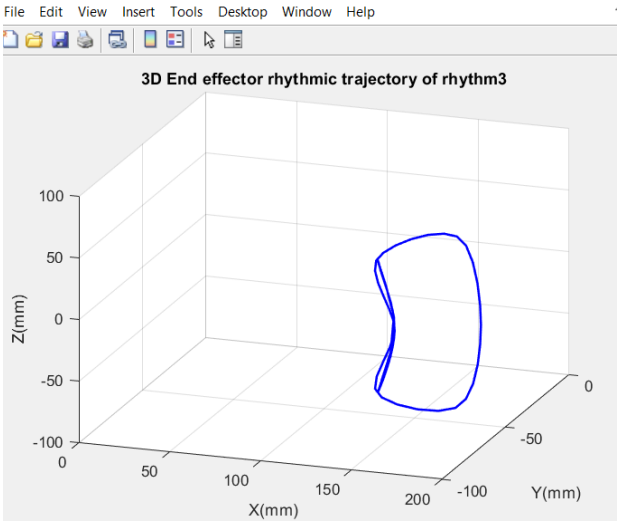


Figure 3-19: Rhythmic trajectory of rhythm three

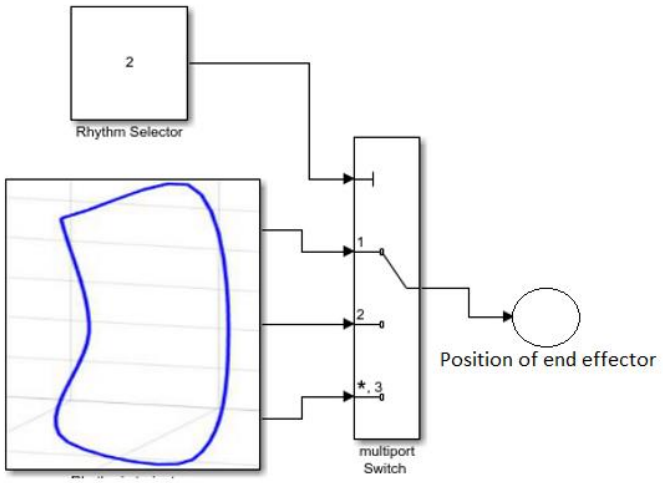


Figure 3-20: MATLAB/Simulink that generate cubic rhythmic trajectory

3.4.6. Control Strategies

SolidWorks, a 3D computer-aided design (CAD) software, is utilized to create a detailed model of the robotic manipulator. The design prioritizes functionality and ease of use for one-handed players, featuring a two-link non-planar structure that balances the necessary range of motion for strumming with a simple, efficient mechanism. Key components include a base unit securely attached to the Krar body for comfortable gripping, two lightweight links connected by revolute joints to provide the desired degrees of freedom (DOF) for strumming, and a mechanism designed for effective contact with the Krar strings as shown in Figure 3-2.

MATLAB/Simulink, a suite for numerical computing and model-based design, is used to simulate the behavior of the robotic manipulator. The SolidWorks model is imported into Simulink via Simscape Multibody, creating a virtual representation of the manipulator within the simulation environment. The developed polynomial trajectories for each Krar rhythm serve as control signal inputs, facilitating the simulation of strumming motion and verification of the manipulator's ability to reproduce the chosen rhythms accurately. This simulation phase allows for the evaluation and optimization of the robotic strummer’s performance before any physical prototyping, ensuring that various control parameters can be tested and the design refined for optimal functionality. Mathematical equation of the manipulator is modeled using Denavit-Hartenberg (DH) parameters to relate joint angles to the end effector's position and orientation.

A closed-loop Proportional-Integral-Derivative (PID) control strategy is implemented to ensure accurate tracking of the desired rhythmic trajectory. Separate PID control loops are established for each joint (θ_1 and θ_2), utilizing the proportional(P) term for current error, the integral(I) term to eliminate steady-state errors, and the derivative(D)S term to enhance responsiveness.

Throughout the control process, the inverse kinematic equations is used to calculate the actual joint angles position based on the rhythmic trajectory inputs, allowing for performance assessment and PID parameter adjustments. This comprehensive approach combines kinematic analysis with closed-loop control to enable RKS effectively play rhythmic pattern of the Krar.

3.4.7. Simulation setup

The robotic manipulator is a 2-DOF (Degrees of Freedom) non-planar manipulator with two links. This design was imported into the MATLAB/Simulink environment, where it serves as the basis for simulating the RKS.This model was subsequently integrated into MATLAB/Simulink for kinematic

analysis and dynamic simulations. The kinematic analysis allows for the precise calculation of joint angles and end-effector positions, which are critical for ensuring the accurate strumming of the Krar strings.

The control system for the RKS is established through a closed-loop framework in MATLAB/Simulink. The system begins with the rhythmic trajectory input, which is fed into the inverse kinematics block to convert task space coordinates (x, y, z) into joint space coordinates. These calculated joint angles are sent to the PID controller block, which minimizes the error between the desired and actual joint angles by adjusting the system's response. The output from the PID block is then processed through the forward kinematics block, which calculates the position of the end-effector (pick) in task space. This output is then compared with the input trajectory, and the error in end-effector position is analyzed for further correction as shown in Figure 3-21. The MATLAB code for the 3D model, inverse kinematics, forward kinematics, and rhythmic trajectory generation is provided in Appendix-A.

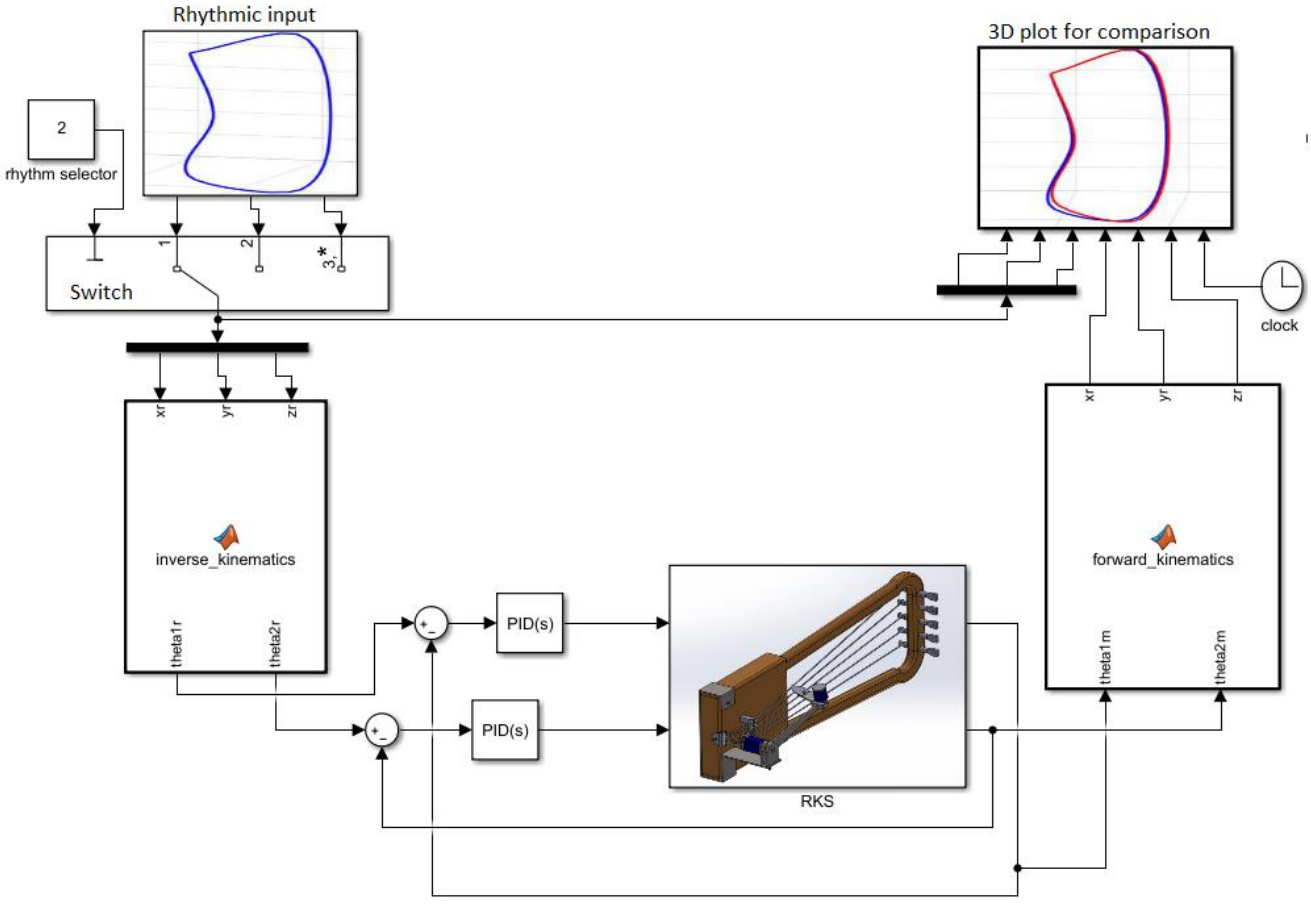


Figure 3-21: MATLAB/Simulink block of joint space control loop

The PID controllers are the most commonly used controllers in industrial robots due to their easy implementation and versatility. Besides, researches prefer PID controllers because of flexibility and ease of use in both simulation and hardware systems. Therefore, a PID controller with position error tracking strategy has been designed to control the rhythmic trajectory input.

A PID controller is preferred among P, PI or PD due to the system behaviors. The rhythmic trajectory involves periodic motion and positional accuracy, the PID controller offers the best balance: P will provide immediate corrections, I ensure no drift in rhythm accuracy over time and D smooths the controller's response to rhythmic transitions, reducing overshoot and oscillation. Therefore, PID controller is recommended because it provides the most comprehensive control over the dynamic and periodic behavior of the robotic Krar strummer, ensuring precise tracking of the cubic rhythmic trajectory.

Automated Method is used to determine the PID parameter values. This approach leverages a MATLAB built-in function PID control law. The transfer function of the system is analyzed, and MATLAB's automated PID tuning capabilities are utilized to calculate the Proportional (P), Integral (I), and Derivative (D) gains. This automated method streamlines the PID parameter selection process, ensuring effective control of the robotic Krar strummer. Because, in real world application, the basic (ideal) PID control law causes control instability and chattering in the output and also challenging in digital control implementation. Therefore, the PID parameters used as shown Table 3-7 are for virtual simulation, and for testing on prototype level retuning the parameters may require.

Table 3-7: PID parameters

PID block	PID parameters	
Joint 1 controller	112	P1
	56	I1
	0.05	D1
Joint 2 controller	116	P2
	46	I2
	0.08	D2

CHAPTER FOURS

RESULTS AND DISCUSSION

The key results obtained from the research highlight the robotic Krar strummer's accuracy in replicating rhythmic patterns. Specifically, joint space trajectory errors were minimal, with Joint 1 showing a maximum deviation of 0.091 degrees and Joint 2 a maximum deviation of 1.56 degrees without disturbance. With disturbance present, Joint 1 showed a maximum deviation of 1.7 degrees and Joint 2 a maximum deviation of 3 degrees. These results, detailed in chapter four, demonstrate the system's ability to closely follow desired trajectories and maintain stability even with disturbances. Following this overview, the discussion delves into the implications of these findings, contextualizing them within existing literature and examining how they address the research questions posed in the introduction.

4.1. DESIGN AND MODELING OF THE ROBOTIC KRAR STRUMMER

The design and modeling of the Krar Strummer served as the foundation for dynamic simulations and control testing. Each link and joint were carefully designed to capture key structural parameters, such as link lengths, masses, and moments of inertia. The design process involved defining critical components that would enable the robot to perform its intended operations, with particular emphasis on maintaining a balance between rigidity and mobility as shown in Figure 3-2. Moreover, link configurations were optimized to achieve a smooth and accurate range of motion across the robot’s workspace. By using SolidWorks, mechanical parts and assemblies were created with high precision, ensuring realistic representations of the physical properties essential for further analysis

The detailed model allowed for accurate extraction of mass and inertia properties. These properties were directly exported to MATLAB/Simulink for integration with the SimMechanics plugin, facilitating the transfer of physical parameters into the dynamic simulation environment. The SolidWorks model provided a comprehensive and reliable representation of the Krar Strummer, enabling seamless data transfer for subsequent kinematic and dynamic analysis. This foundational stage confirmed the feasibility of the robot’s design and established the groundwork for the simulation and control phases of the project.

4.2. KINEMATIC AND DYNAMIC MODELING WITH TRAJECTORY DESIGN

The second phase of the project focused on developing the kinematic and dynamic models of the robotic Krar Strummer, followed by a sample rhythmic trajectory design for verification purposes. These models provided a mathematical framework to describe the robot's motion and analyze its response to different input trajectories. The kinematic model was developed using the DH convention, which provided a systematic approach for defining joint transformations in robotic systems. Using the DH parameters, transformation matrices were derived for each joint, establishing the sequential transformations needed to reach the end-effector's position in the global coordinate frame as shown Figure 4-1, and stated in equation 3-4.

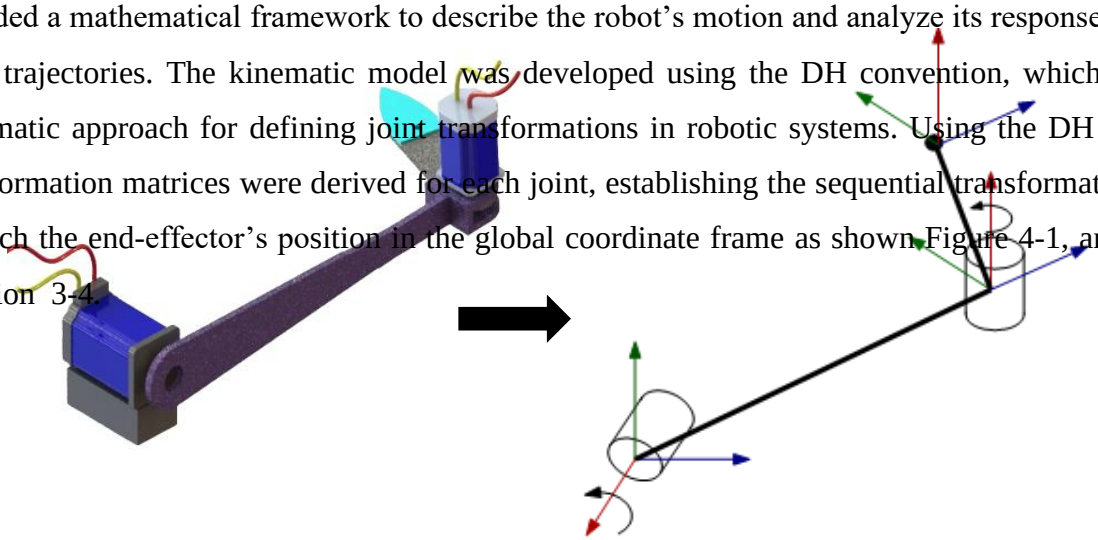


Figure 4-1: DH, two-link non-planar robotic manipulator

The inverse kinematic model provides the joint configurations required to achieve a specific position and orientation of the end-effector. The development of these kinematic models provided a reliable framework for position and orientation control, confirming that the Krar Strummer could be programmed to achieve specific tasks with high precision. To do so a sample cubic polynomial rhythmic trajectory was developed to evaluate the accuracy of the kinematic models, and the effectiveness of the control system. Verification tests, *Figure 4-2*, confirmed the consistency of the kinematic models under various operating conditions, demonstrating the robot's capability to execute complex tasks.

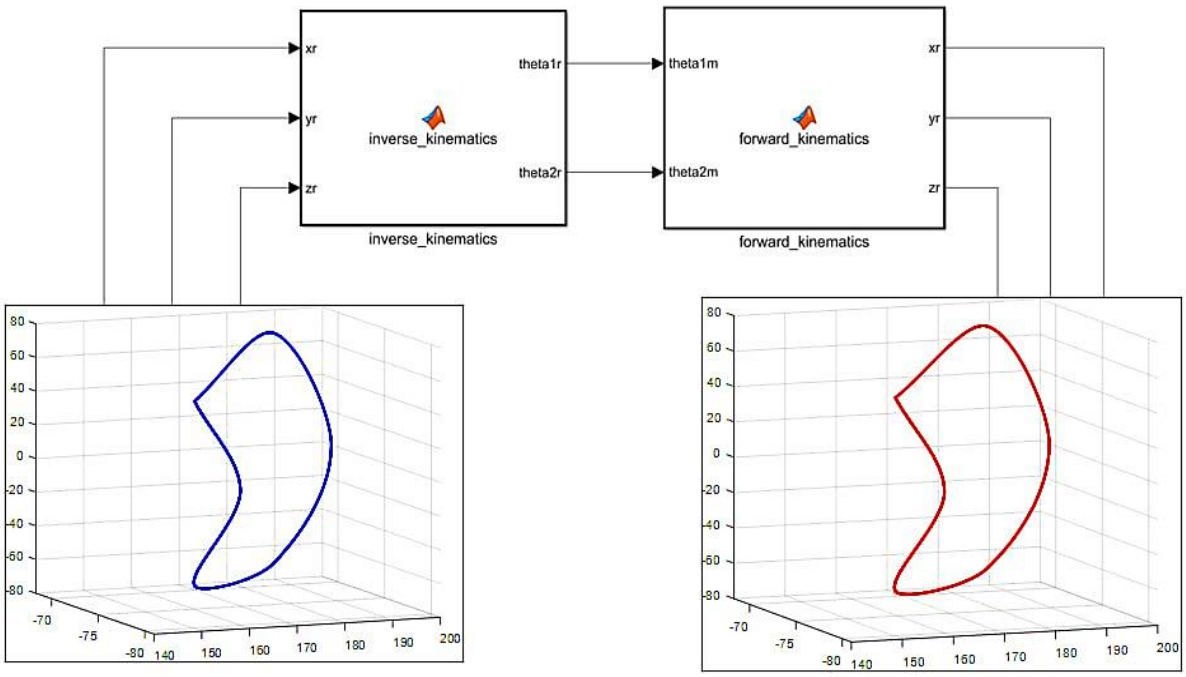


Figure 4-2: Forward and Inverse Kinematics verification result using MATLAB/Simulink

In addition to kinematic modeling, a dynamic model was developed to account for the forces and torques acting on each link of the robot. This model incorporated parameters such as link masses, centers of mass, and moments of inertia, which were essential for realistic simulations of the robot's motion. The integration of SolidWorks and MATLAB/Simulink using the SimMechanics plugin facilitated the direct import of physical properties into the dynamic simulation environment, improving model accuracy and reducing data processing time of the model for control applications. The dynamic model enabled the analysis of the robot's response to external forces and torques, providing insights into stability and control requirements under operational conditions

4.3. CONTROLLER DESIGN AND SIMULATION RESULTS

The final stage of the thesis involved the design and testing of a proportional-integral-derivative (PID) controller to ensure that the robotic Krar Strummer could follow the specified trajectory accurately. The controller was implemented in MATLAB/Simulink, and its performance was evaluated under various test scenarios to determine its effectiveness in maintaining stability and accuracy. The controller's

parameters were tuned to achieve an optimal balance between response speed and stability, allowing the robot to track the desired trajectory with minimal error.

4.3.1. Time response analysis results of the RKS under no disturbance

The three-input rhythmic trajectories, without disturbances, were simulated in the MATLAB/Simulink environment, with the maximum error occurring in rhythmic trajectory two. For the purpose of discussing the results, rhythmic trajectory two is selected, as it shows the highest error of 2.8 mm among the three rhythmic inputs. Furthermore, the results of this trajectory is representative and inclusive of the other two rhythmic inputs (trajectory one and three). The simulation results for the maximum end-effector (pick) error and joint error for rhythmic trajectory are presented in Table 4-1, while the simulation results for the other two rhythmic trajectories are provided in Appendix A.

Table 4-1: maximum error simulation result without disturbance

Maximum Error(mm) of end effector	Rhythm_1	Rhythm_2	Rhythm_3		Joint_1	Joint_2	Maximum error(degree) joint space
e_x	0.0493	1.7	1.8		0.011	0.032	Rhythm_1
e_y	0.0122	2.8	2.42		0.091	1.56	Rhythm_2
e_z	0.0259	0.6416	0.65		0.02	1.46	Rhythm_3

To assess the time response of the robotic Krar Strummer's Joint 1, the joint space controller was tested with a reference input based on the desired joint trajectory, calculated from the end-effector(pick) trajectory using the inverse kinematics model developed in Chapter Three. The simulation revealed a maximum deviation of 0.091 degrees from the desired path, Figure 4-3. This level of error is within the acceptable tolerance range for the application and has a negligible impact on the rhythmic pattern of the input trajectory, which is essential for smooth and consistent movement. Consequently, the controller demonstrated its effectiveness in maintaining precise alignment with the desired trajectory, ensuring stable operation without disrupting the robot's motion.

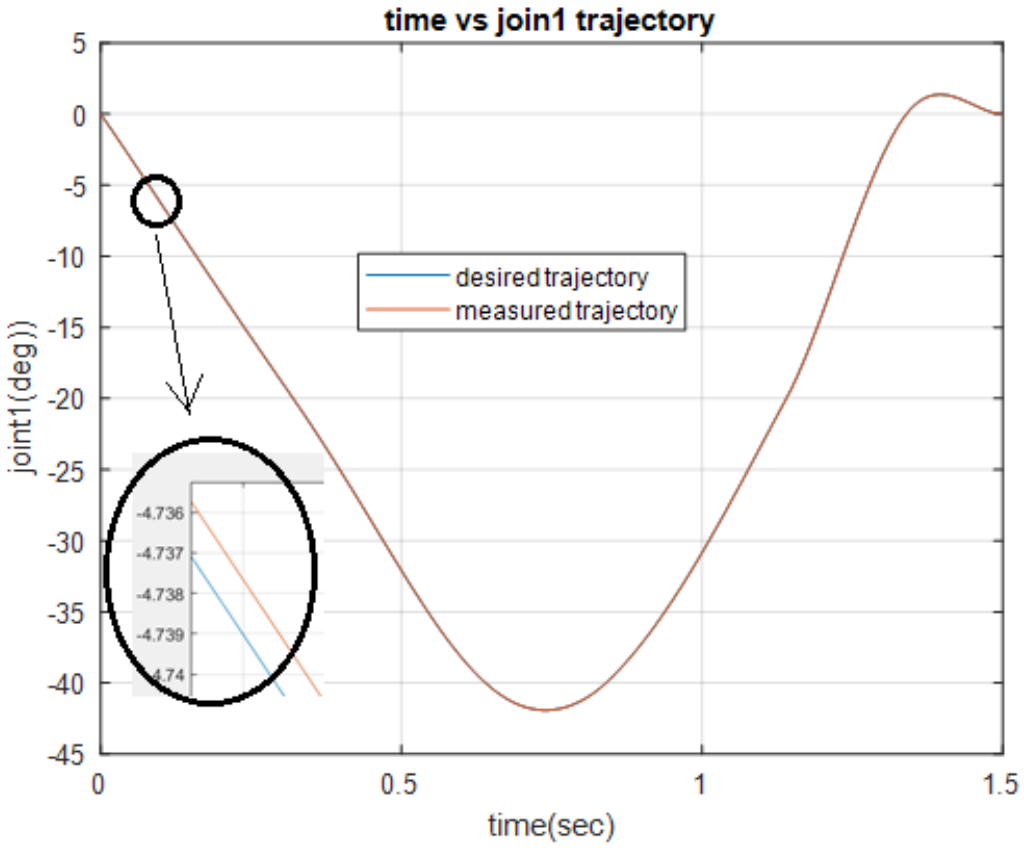


Figure 4-3: joint_1 joint space trajectory of rhythmic input one without disturbance

Similarly, the time response of Joint 2's controller for the robotic Krar strummer was assessed by providing a reference input based on the desired joint trajectory, which was calculated from the end-effector (pick) path using the inverse kinematics model developed in Chapter Three. The simulation results indicated a maximum deviation of 1.56 degrees from the desired trajectory Figure 4-4. This deviation is within the acceptable tolerance range for the application and has minimal effect on the rhythmic pattern of the input trajectory. Hence, the controller effectively maintained the alignment of Joint 2 with the desired motion path, ensuring smooth and consistent performance without compromising the overall operation of the robot.

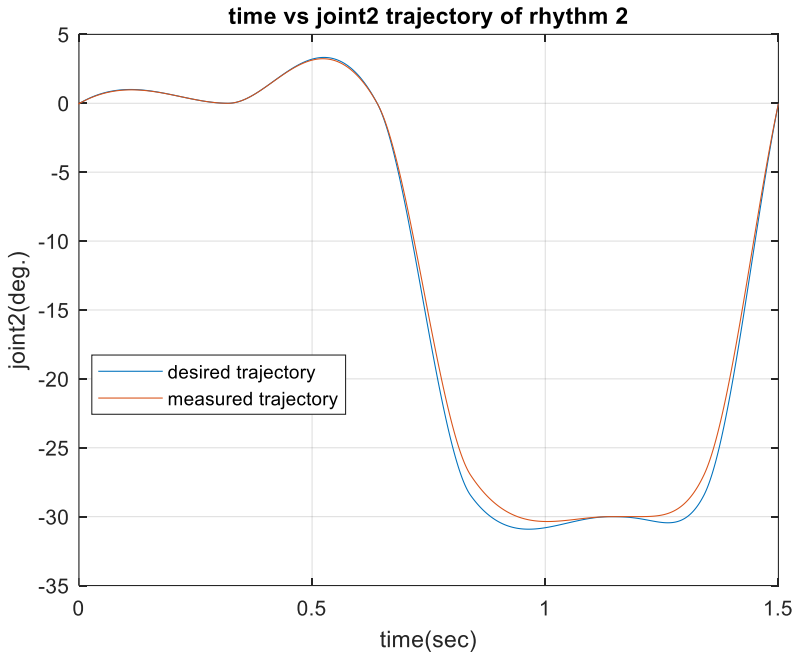


Figure 4-4: joint_2 joint space trajectory of rhythmic input two without disturbance

The trajectory tracking of the end-effector, which is directly influenced by the effectiveness and accuracy of the joint space controllers of the robotic Krar Strummer, was evaluated by feeding the robot with three different rhythmic trajectories as discussed in in chapter three. The robot followed these trajectories with high accuracy, demonstrating the capability of the joint controllers to maintain precise end-effector. As shown in Figure 4-5, the maximum error for rhythmic trajectory two is 2.8mm in the y-directions.

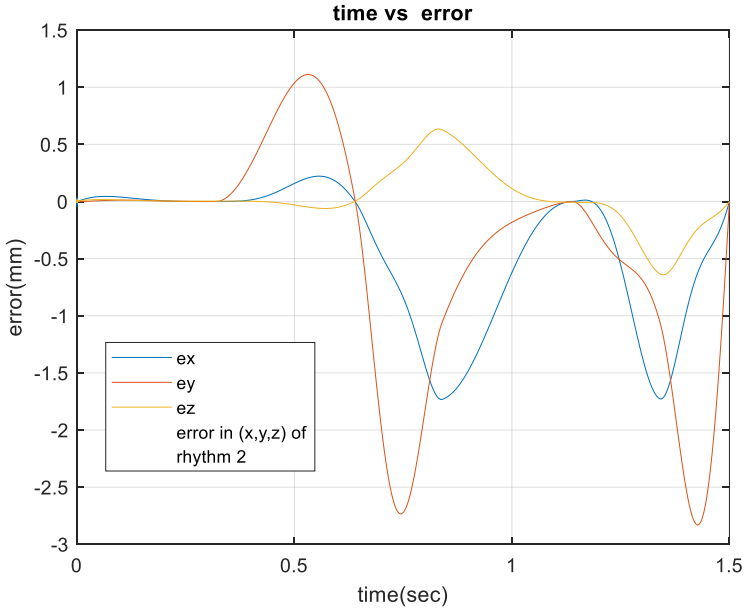


Figure 4-5: maximum end effector error of rhythmic trajectory two without disturbance

The maximum errors observed were 1.7 mm along the x-axis, 2.8mm along the y-axis, and 0.65mm along the z-axis, which were encountered in the rhythm trajectory sample 2 as shown Figure 4-6. These errors are considered insignificant in terms of the rhythmic pattern, as they have minimal impact on the overall trajectory execution. The joint space controllers, therefore, effectively ensured that the end-effector adhered to the desired rhythmic motion, confirming the PID controllers' ability to perform consistent and reliable movements.

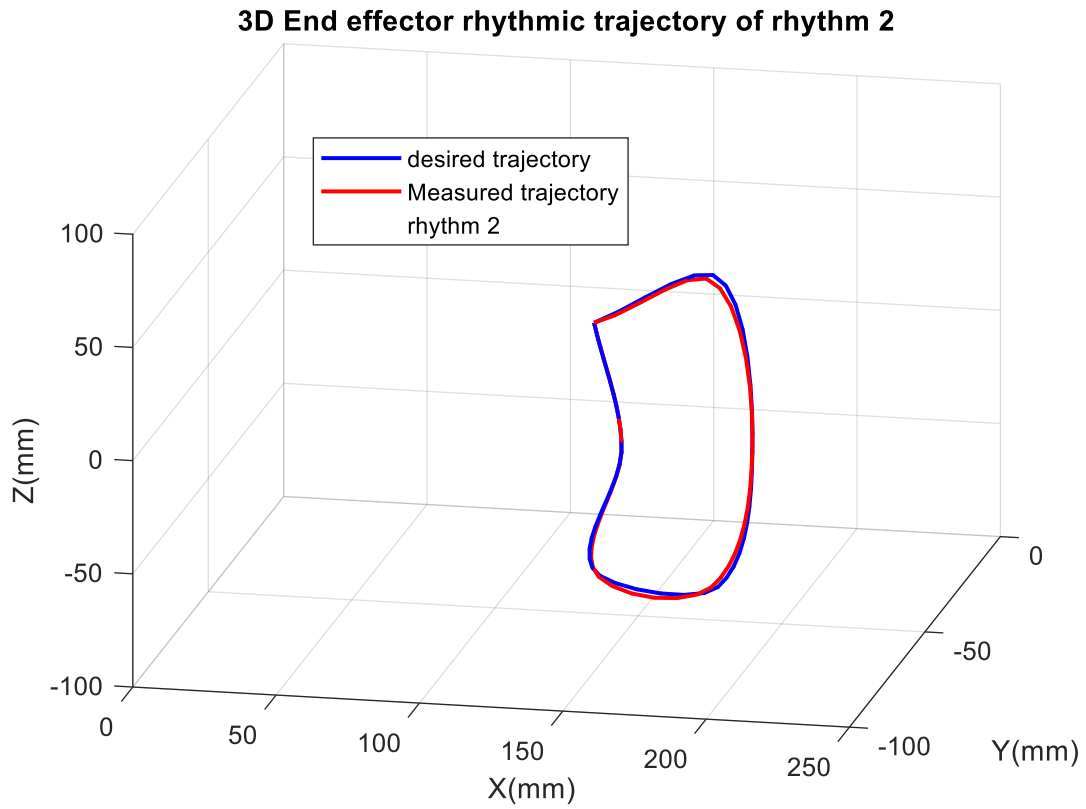


Figure 4-6: Desired and measured end-effector rhythmic trajectory of rhythm two

4.3.2. Time response analysis results of the RKS under disturbance

The behavior of RKS is oscillatory motion due to repetitive pattern of the rhythmic trajectory. Therefore, a sinusoidal disturbance can simulate rhythmic variations or oscillations that may occur during the strumming process, such as variations in the force applied to the strings or small vibrations from the Krar body. Testing how the PID controller handles such oscillatory disturbances can give insight into how well it performs in maintaining the desired strumming trajectory. A sinusoidal disturbance was added the position input of the PID controller, with a frequency of 2 Hz (two beats per second which is equal with

tempo of the rhythmic audio) and amplitude of 1.5mm that represent realistic strumming variations shown In Figure 4-7.

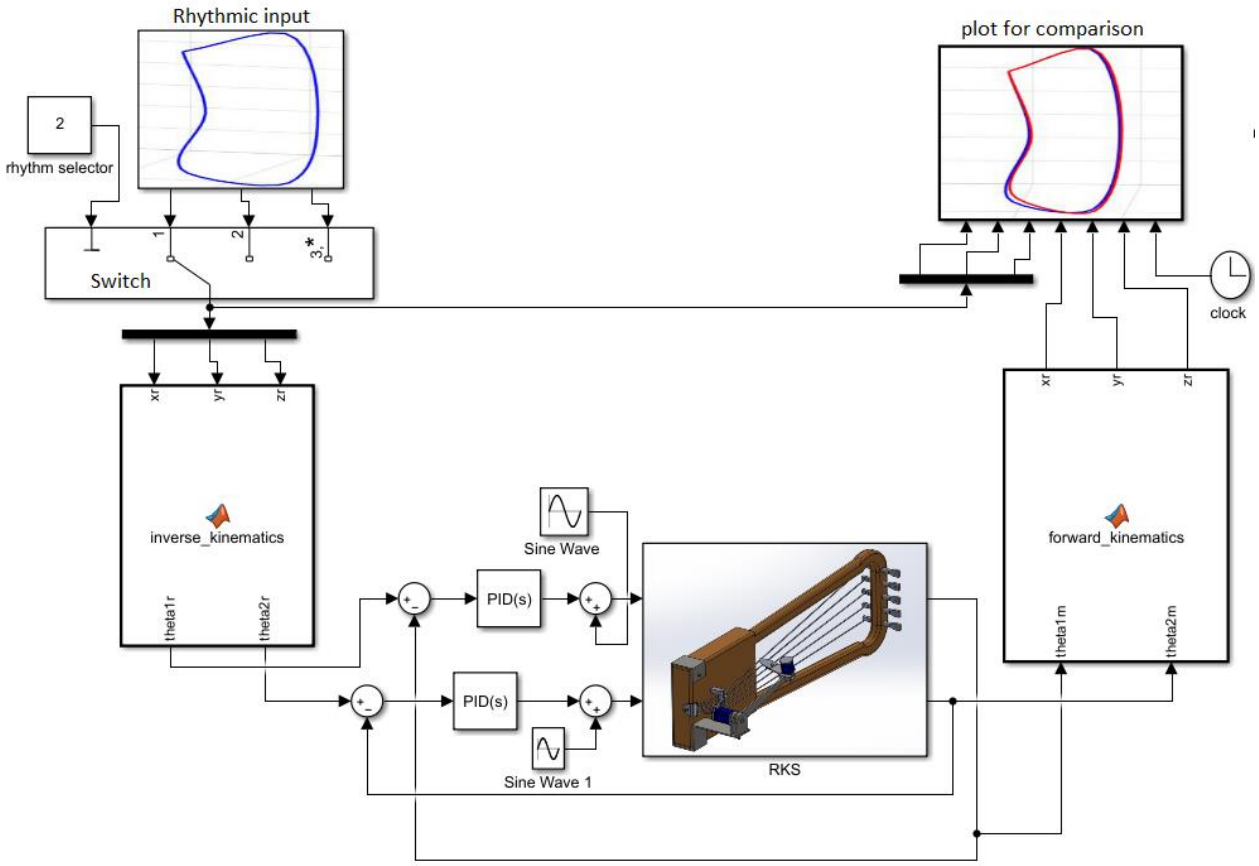


Figure 4-7: closed loop control of RKS under disturbance

Similarly, the three-input rhythmic trajectories under disturbance were simulated in the MATLAB/Simulink environment, with the maximum error occurring in rhythmic trajectory two. For the purpose of result discussion, rhythmic trajectory two under disturbance is selected, as it has the highest error of 6 mm compared to the other two rhythmic inputs. Additionally, the results for rhythmic trajectory input two is representative of the other two rhythmic inputs (trajectory one and three). The simulation results for the maximum end-effector error for rhythmic trajectory two under disturbance are presented in Table 4-2. while the simulation results for the other two rhythmic trajectories are provided in Appendix A.

Table 4-3: maximum error simulation result under disturbance

Maximum Error(mm) of end effector	Rhythm_1	Rhythm_2	Rhythm_3		Joint_1	Joint_2	Maximum error(degree) joint space
e_x	3.43	3.3	4.1		1.45	1.6	Rhythm_1
e_y	0.07	3.37	3.2		1.6	3	Rhythm_2
e_z	3.8	6	4		1.5	2.75	Rhythm_3

To evaluate the time response of the robotic Krar Strummer's Joint 1, the joint space controller was tested using a reference input derived from the desired joint trajectory, which was computed from rhythmic trajectory input using the inverse kinematics model discussed in Chapter Three. The simulation indicated a maximum deviation of 1.6 degrees from the intended path. This error falls within the acceptable tolerance range for the application and has a minimal effect on the rhythmic pattern of the input trajectory, ensuring smooth and consistent movement. As a result, the controller proved effective in maintaining precise alignment with the desired trajectory, ensuring stable operation without disrupting the robot's motion.

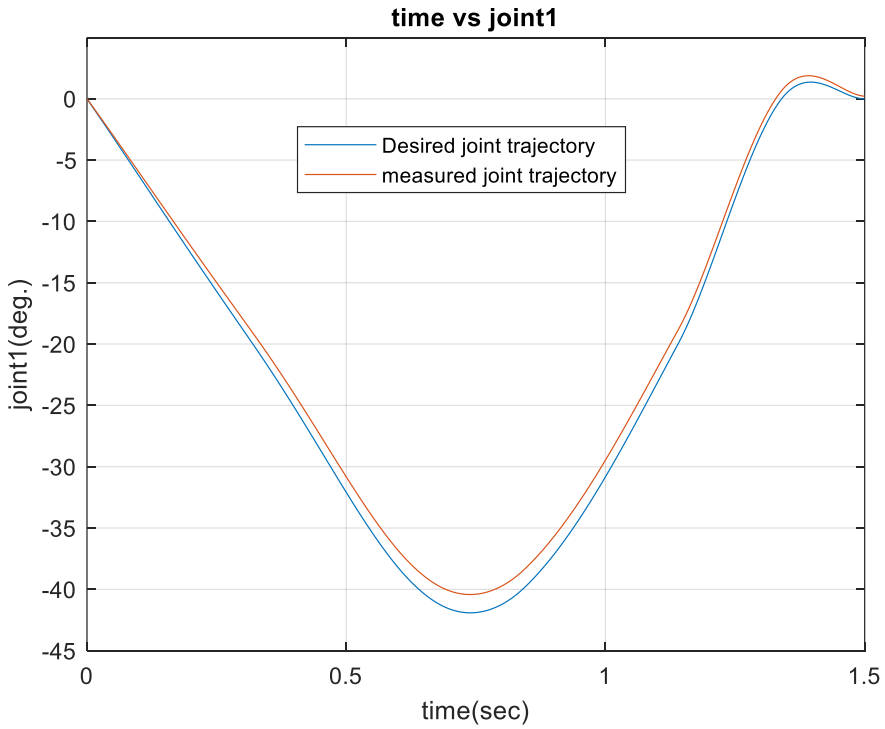


Figure 4-8: joint_1 joint space trajectory of rhythmic input two under disturbance

Similarly, the time response of the controller for Joint 2 of the robotic Krar strummer was evaluated by applying a reference input based on the desired joint trajectory, which was derived from the rhythmic trajectory input using the inverse kinematics model developed in Chapter Three. The simulation results showed a maximum deviation of 3 degrees from the intended trajectory. This deviation falls within the acceptable tolerance range for the application and has little impact on the rhythmic pattern of the input trajectory. Therefore, the controller successfully maintained Joint 2's alignment with the desired motion path, ensuring smooth and consistent performance without affecting the overall operation of the RKS.

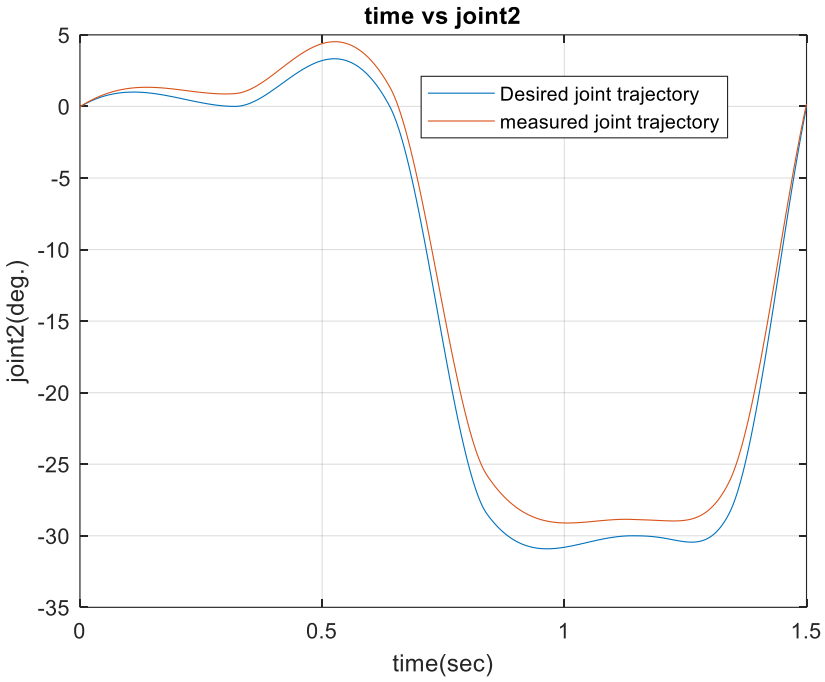


Figure 4-9: joint_2 joint space trajectory of rhythmic input two under disturbance

The trajectory tracking of the end-effector under disturbance, which is directly influenced by the accuracy and performance of the joint space controllers of the robotic Krar Strummer, was evaluated using three distinct rhythmic trajectories under disturbance. The robotic Krar Strummer (RKS) demonstrated high accuracy in following these trajectories, highlighting the effectiveness of the joint controllers in maintaining precise end-effector motion even under disturbance. As shown inFig. Figure 4-10, the maximum error for rhythmic trajectory two was recorded as 6 mm in the z-direction.

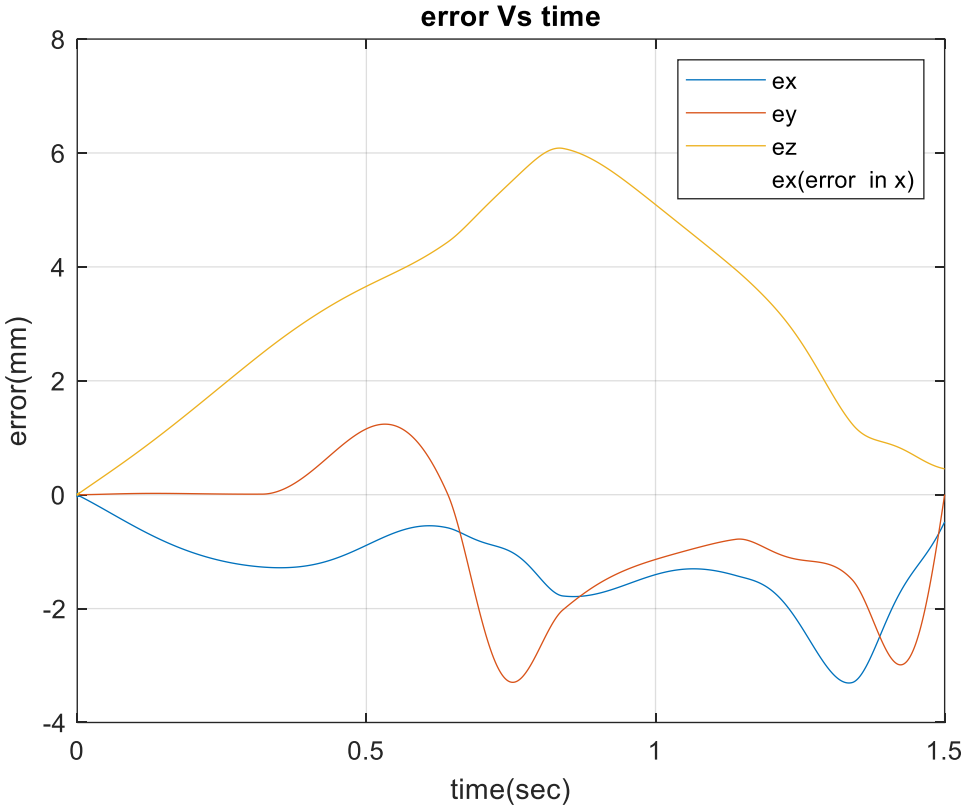


Fig. Figure 4-10: maximum error end effector of rhythmic trajectory two under disturbance

The maximum errors recorded were 3.3 mm along the x-axis, 3.29 mm along the y-axis, and 6 mm along the z-axis, all occurring in rhythmic trajectory sample 2 as shown Figure: Figure 4-11. These errors are considered minimal with respect to the rhythmic pattern, as they have a negligible effect on the overall execution of the trajectory. The joint space controllers effectively maintained the end-effector's adherence to the desired rhythmic trajectories motion, demonstrating the PID controllers' capability to deliver consistent and reliable performance.

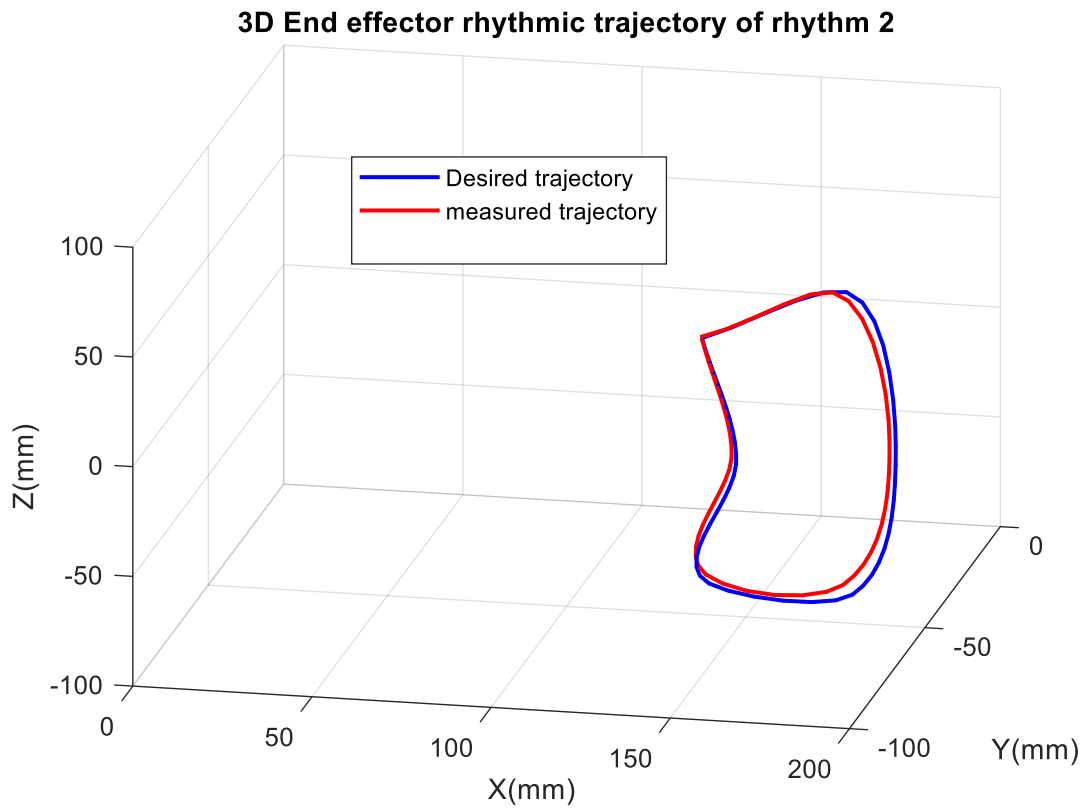


Figure: Figure 4-11: Desired and measured end-effector rhythmic trajectory of rhythm two under disturbance

The simulation results confirmed that the PID controller effectively reduced tracking errors, allowing the end-effector to closely follow the cubic polynomial rhythmic trajectory. The controller demonstrated high accuracy in maintaining the desired path, ensuring that deviations were kept to a minimum. Additionally, the PID controller provided a stable response, even when external sinusoidal disturbances were introduced. This robustness highlights the controller's ability to adapt to changing conditions without compromising performance. Overall, the results emphasize the reliability of the PID controller in achieving precise and stable trajectory tracking, making it well-suited for the rhythmic tasks required by the robotic Krar Strummer.

CHAPTER FIVE

CONCLUSION AND RECOMMENDATION

5.1. CONCLUSION

This thesis successfully presents the design, modelling, and simulation of a robotic Krar strummer for one-handed individuals, developing a kinematic and dynamic model that enables control through inverse kinematics and PID controller. A rhythmic trajectory generated from the analyzed rhythmic audio was implemented in the MATLAB/Simulink environment, where simulations using a PID controller demonstrated the feasibility robotic Krar strummer's ability to follow the desired rhythmic trajectory. We successfully developed a simulated robotic Krar strummer model, utilizing SolidWorks for 3D modelling and MATLAB/Simulink for dynamic behavior simulation and control testing.

The RKS mathematically model to function as a 2-DOF non-planner robotic manipulator, it is modeled kinematically using the standard DH convention and dynamically modeled using Euler-Lagrange equations. To accurately capture nonlinearities, coupling effects, and emulate real-world behavior, the RKS's dynamics were represented using Simscape Multibody System in MATLAB-Simulink. This was achieved by exporting the physical 3D CAD model from SOLIDWORKS into MATLAB/Simulink using the Simscape Multibody Plug-in. Based on its dynamics, two PID controllers are independently tuned for each joint.

The user experience is a crucial consideration, highlighting the need for future research to involve one-handed Krar players in user testing and feedback sessions to ensure that the final design meets their specific needs and preferences, thus promoting user adoption and maximizing the assistive value of the robotic strummer.

In summary, this thesis contributes to the field of assistive technology by proposing a novel solution for one-handed individuals to enjoy playing the traditional Krar instrument. The robotic Krar strummer has the potential to enhance musical expression and inclusivity. This thesis lay the groundwork for ground-breaking assistive technology for traditional musical instrument such as Krar, enhancing accessibility for one-handed individuals and taking a significant step toward inclusivity in music creation. This research also opens doors for exploring robotic assistants for other string instruments, enriching the musical landscape for people with disabilities.

5.2. RECOMMENDATION

In this thesis, the control system for the robotic Krar strummer has been designed and simulated using a PID controller to regulate the strumming motion, demonstrating effective control in achieving desired strumming position and timing. Future development includes building a physical prototype based on the developed model and testing its performance with real-world Krar instruments, as well as investigating alternative control methods, such as artificial neural network control that allows the RKS to learn, and enable more enhanced playing techniques.

However, for actual deployment in real-world conditions, it is recommended to implement a PID controller with gravity compensation to enhance performance. Gravity compensation is crucial for managing the dynamic forces acting on the robotic arm or strumming mechanism, especially when playing the Krar at various angles and orientations. While the current simulation does not explicitly consider gravitational forces, focusing instead on the controller's functionality in an idealized environment, real-world implementation may introduce unwanted torque and load that could affect the accuracy of the strumming motion.

Therefore, future work should focus on incorporating a gravity compensator with the existing PID control system to counteract gravitational forces when the robotic arm is positioned at different angles. This integration will facilitate smoother and more precise control of the strumming mechanism, particularly during changes in the Krar's position or orientation. Additionally, once the gravity compensator is integrated, the system should be tested with a physical prototype to validate the controller's ability to maintain consistent strumming patterns under real-world conditions, including a variety of Krar orientations and playing environments. Furthermore, the PID controller will require re-tuning in the presence of the gravity compensator to ensure optimal performance; fine-tuning the proportional, integral and derivative gains will improve responsiveness and stability against gravitational effects.

REFERENCE

[1] I. Handžić and K. B. Reed, "The musical kinetic shape: A variable tension string instrument," *Appl. Acoust.*, vol. 85, pp. 143–149, 2014, doi: https://doi.org/10.1016/j.apacoust.2014.04.010.

[2] S. Ferretti, "On the modeling of musical solos as complex networks," *Inf. Sci. (Ny).*, vol. 375, pp. 271–295, 2017, doi: https://doi.org/10.1016/j.ins.2016.10.007.

[3] O. Flor, M. Fuentes, and C. Toapanta, "Criteria for the design of an educational robotics platform," *Athenea*, vol. 1, pp. 29–40, Sep. 2020, doi: 10.47460/athenea.v1i1.4.

[4] Y. Yoshitake, A. Ikeda, and M. Shinohara, "Robotic finger perturbation training improves finger postural steadiness and hand dexterity," *J. Electromyogr. Kinesiol.*, vol. 38, pp. 208–214, 2018, doi: https://doi.org/10.1016/j.jelekin.2017.11.011.

[5] S. Ueki, H. Kawasaki, and T. Mouri, "Adaptive Control of Multi-fingered Robot Hand Using Quaternion," *IFAC Proc. Vol.*, vol. 41, no. 2, pp. 6757–6762, 2008, doi: https://doi.org/10.3182/20080706-5-KR-1001.01145.

[6] H. Park, B. Lee, and D. Kim, "Violin Musical Tone Analysis Using Robot Finger," *Procedia Comput. Sci.*, vol. 94, pp. 398–403, 2016, doi: https://doi.org/10.1016/j.procs.2016.08.061.

[7] H. Feng, M. Mahoor, and F. Dino, *A Music-Therapy Robotic Platform for Children with Autism: A Pilot Study*. 2022.

[8] E. Frid and A. Ilsar, *Reimagining (Accessible) Digital Musical Instruments: A Survey on Electronic Music-Making Tools*. 2021.

[9] A. S. Prabuwono, K. Allehaibi, and K. Kurnianingsih, "Assistive Robotic Technology: A Review," *Comput. Eng. Appl. J.*, vol. 6, pp. 71–78, Jul. 2017, doi: 10.18495/comengapp.v6i2.203.

[10] E. Singer, K. Larke, and D. Bianciardi, *LEMUR GuitarBot: MIDI Robotic String Instrument*. 2003.

[11] C. Lopez-Franco, D. Diaz, J. Hernandez-Barragan, N. Arana-Daniel, and M. Lopez-Franco, "A Metaheuristic Optimization Approach for Trajectory Tracking of Robot Manipulators," *Mathematics*, vol. 10, p. 1051, Mar. 2022, doi: 10.3390/math10071051.

[12] K. M. Dogan, T. Yucelen, and J. Muse, *Adaptive Control of Dynamical Systems with Unstructured Uncertainty and Unmodeled Dynamics*. 2019.

[13] A. Cemgil *et al.*, "On Tempo Tracking: Tempogram Representation and Kalman Filtering," *J. New Music Res.*, vol. 28, p. 259, Jan. 2001.

[14] A. Nagchaudhuri and D. P. Garg, "Adaptive control strategies for coordination and control of mutliple robots," *Am. Soc. Mech. Eng. Dyn. Syst. Control Div. DSC*, vol. 70, pp. 925–933, Jan. 2002.

[15] M. Khan, M. Ul Islam, and J. Iqbal, *Control strategies for robotic manipulators*. 2012.

[16] I. Handzic and K. Reed, "The musical kinetic shape: A variable tension string instrument," *Appl. Acoust.*, vol. 85, pp. 143–149, Nov. 2014, doi: 10.1016/j.apacoust.2014.04.010.

[17] B. Suthar, Y. Abdulrahman, and Y. Zweiri, "Robotic Fingers: Advancements, Challenges, and Future Directions- A Comprehensive Review," *IEEE Access*, vol. PP, p. 1, Jan. 2024, doi:

10.1109/ACCESS.2024.3440007.

[18] N. Yang, R. Savery, R. Sankaranarayanan, L. Zahray, and G. Weinberg, *Mechatronics-Driven Musical Expressivity for Robotic Percussionists*. 2020.

[19] J. J. Craig, *Introduction to Robotics: Mechanics and Control*, 3rd ed. London: Pearson Education, Inc., 2005.

[20] E. M. R. and Q. Gan, "Forward and Inverse Kinematics Models for a 5-dof Pioneer 2 Robot Arm," Tech. Rep. CSM-413, 2002, [Online].

Available:https://www.researchgate.net/publication/244270445_Forward_and_Inverse_Kinematics_Models_for_a_5-dof_Pioneer_2_Robot_Arm

[21] S. Kucuk and Z. Bingul, "Robot Kinematics: Forward and Inverse Kinematics," in *Industrial Robotics: Theory, Modelling and Control*, no. December, 2006. doi: 10.5772/5015.

[22] S. You, "Dynamics and controls for robot manipulators with open and closed kinematic chain mechanisms," p. 165, 1994, [Online].

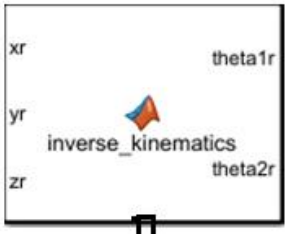
Available: https://dr.lib.iastate.edu/entities/publication/8f1ab766-f6de-4786-b439-c1c96efeaf2a

[23] M. V., Mark W. Spong, and Seth Hutchinson, *Robot Dynamics and Control*, 2nd ed. Wiley India Pvt. Limited, 2004.

APPENDICES

APPENDIX-A: MATLAB CODE

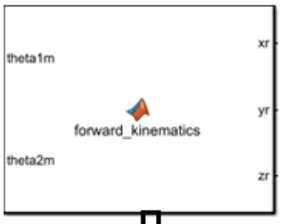
MATLAB code of inverse kinematics:



inverse_kinematics

```
function [theta1r,theta2r] = inverse_kinematics(xr,yr,zr)
 L1=160;
theta1r=(atan2d(zr,xr))-20;
theta2r=(atan2d(-yr,(xr-L1*cosd(theta1r+20))))-90;
```

MATLAB code of forward kinematics:



forward_kinematics

```
function [xr,yr,zr] = forward_kinematics(theta1m,theta2m)
 L1=160;
 L2=80;
s1=sind(theta1m+20);
c1=cosd(theta1m+20);
s2=sind(theta2m+90);
c2=cosd(theta2m+90);
xr= abs((L1*c1)+(L2*c1*c2));
zr= (L1*s1)+(L2*s1*c2);
yr= -L2*s2;
```

MATLAB code of rhythmic input generator:

```
function pos = cubic_trajectory_3D(t, waypoints, timepoints)
    % pos = cubic_trajectory_3D_clock(t, waypoints, timepoints)
    % waypoints: Nx3 matrix of [x, y, z] positions (N waypoints in 3D space)
    % time_points: 1xN vector of times corresponding to each waypoint
    % t: clock signal input (time value from Simulink or function call)
    % Number of waypoints
    num_waypoints = size(waypoints, 1);
    % Initialize output position
    pos = zeros(1, 3);  % Position [x, y, z]
    % Loop through each segment and calculate trajectory based on clock time
    for i = 1:num_waypoints-1
        % Time duration for each segment
        t0 = time_points(i);
        t1 = time_points(i+1);
        % Check if current clock time is within the current segment
        if t >= t0 && t <= t1
            % Time interval
            T = t1 - t0;
            % Loop through each dimension (x, y, z)
            for dim = 1:3
                q0 = waypoints(i, dim);     % Starting point of segment in this dimension
                q1 = waypoints(i+1, dim);   % End point of segment in this dimension
                % Compute velocities for smooth transition (finite differences)
                if i == 1
                    v0 = (waypoints(i+1, dim) - waypoints(i, dim)) / (time_points(i+1) - time_points(i));
                else
                    v0 = (waypoints(i+1, dim) - waypoints(i-1, dim)) / (time_points(i+1) - time_points(i-1));
                end
                if i == num_waypoints-1
                    v1 = (waypoints(i+1, dim) - waypoints(i, dim)) / (time_points(i+1) - time_points(i));
                else
                    v1 = (waypoints(i+2, dim) - waypoints(i, dim)) / (time_points(i+2) - time_points(i));
                end
```

```
            % Solve for the coefficients of the cubic polynomial
            A = [1, 0, 0, 0;
                1, T, T^2, T^3;
                0, 1, 0, 0;
                0, 1, 2*T, 3*T^2];

            B = [q0; q1; v0; v1];

            coeffs = A \ B;  % Solve for coefficients a0, a1, a2, a3

            % Calculate position for this dimension at time t
            pos(dim) = coeffs(1) + coeffs(2)*(t-t0) + coeffs(3)*(t-t0)^2 + coeffs(4)*(t-t0)^3;
        end
        break;  % Exit the loop once the segment is found
    end
  end
end
```

APPENDIX-B: SIMULATION RESULT OF RHYTHMIC INPUT TRAJECTORY ONE AND THREE

a. Simulation result of rhythmic input trajectory one with no disturbance:

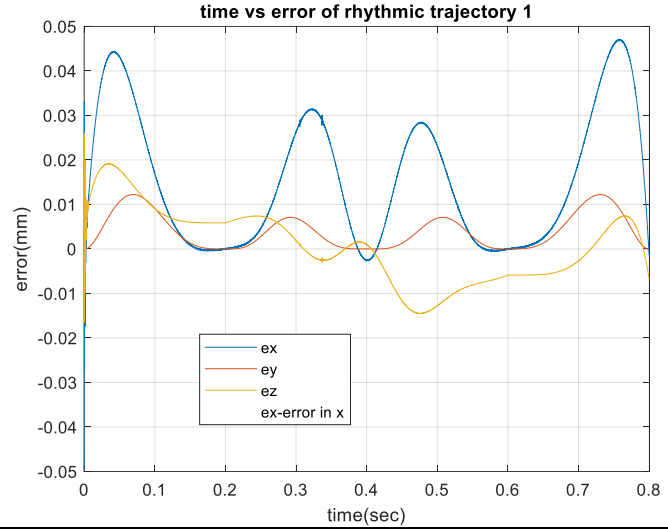
error		Error maximum
ex		0.05
ey		0.012
ez		0.026

Figure a-1: time vs error of rhythm one

The maximum errors of the end-effector (pick) were recorded as 0.05 mm in the x-direction, 0.012 mm in the y-direction, and 0.026 mm in the z-direction. These errors are negligible and do not significantly impact the overall performance.

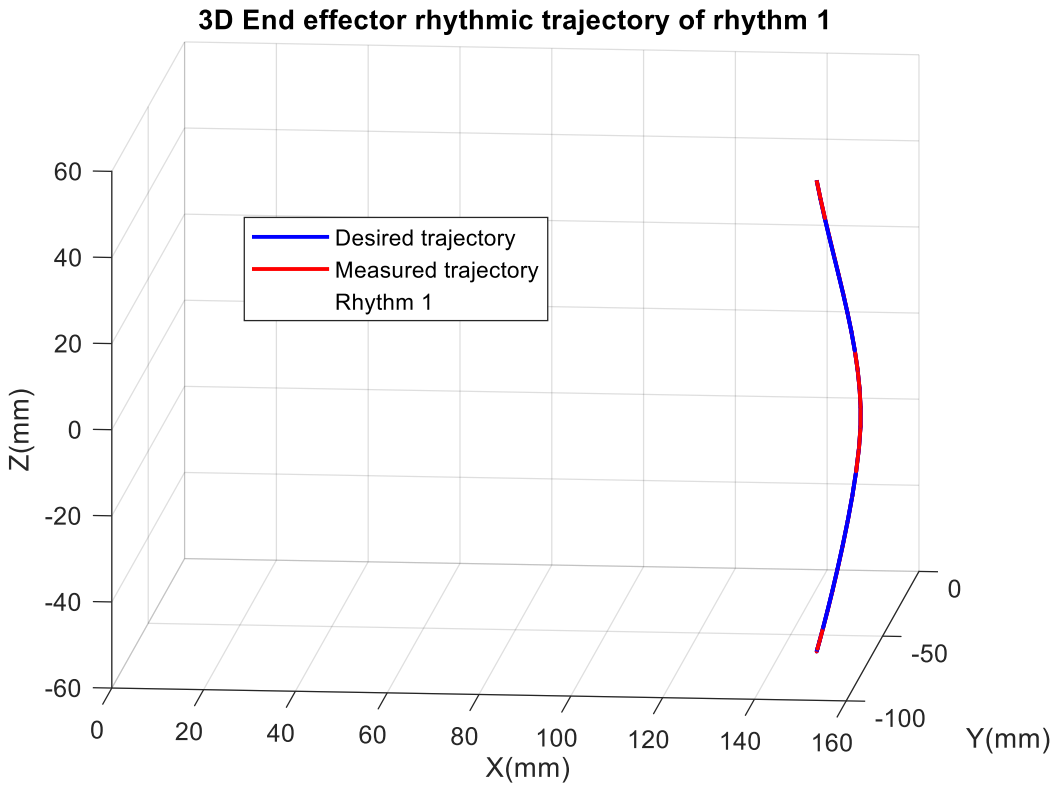


Figure a-2: end effector rhythmic trajectory one

b. Simulation result of end effector rhythmic input trajectory one: under disturbance

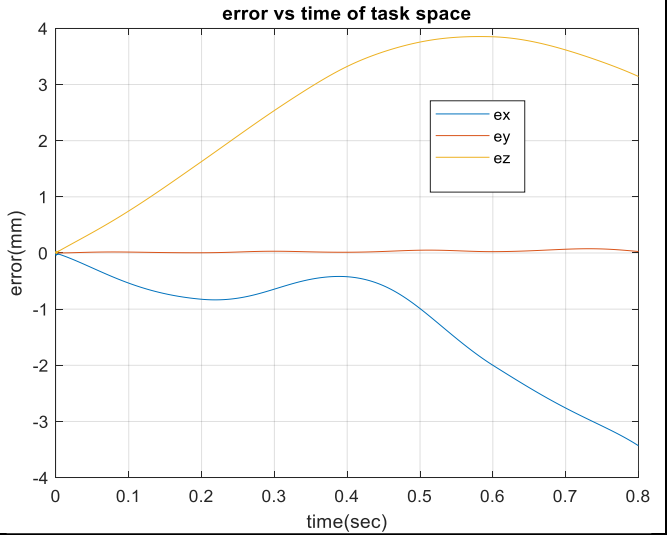
error		Error maximum
ex		3.4
ey		0.07
ez		3.8

Figure b-1: time vs error of rhythm one under disturbance

The maximum errors of the end-effector (pick) under disturbance were 3.4 mm in the x-direction, 0.07 mm in the y-direction, and 3.8 mm in the z-direction. While these errors are minimal, they fall within the acceptable range of error for the system's performance.

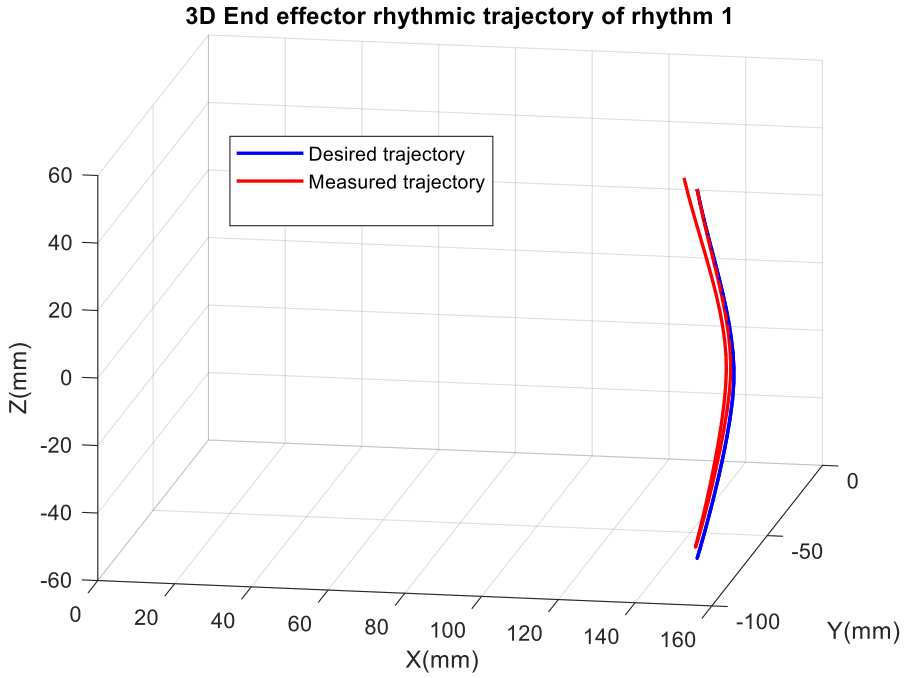
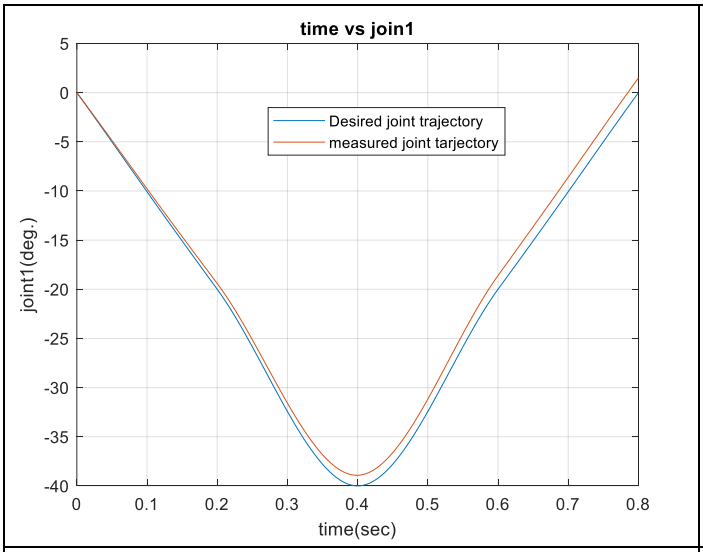
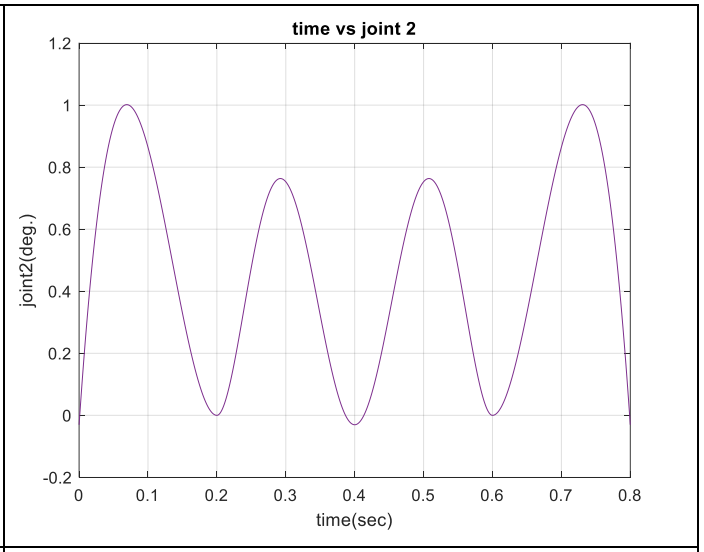


Figure b-2: end effector rhythmic trajectory one under disturbance

c. Simulation result of joint space input rhythmic trajectory one: under disturbance



Joint_1: joint space trajectory



Joint_2: joint space trajectory

The joint space trajectories of Joint 1 and Joint 2 for Rhythm 1 exhibited errors of 1.5 degrees and 1.6 degrees, respectively, under disturbance. These errors are within the acceptable range for the system's performance.

d. Simulation result of end effector rhythmic input trajectory three: with no disturbance

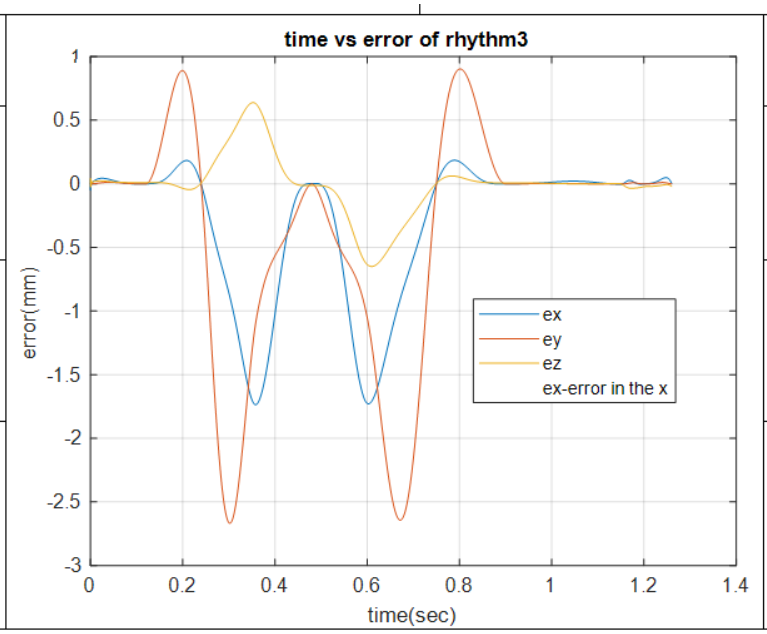
error		Error maximum
ex		1.8
ey		2.4
ez		0.65

Figure c-1: time vs error of rhythm three under disturbance

The maximum errors of the end-effector (pick) for rhythmic input three were recorded as 1.8 mm in the x-direction, 2.4 mm in the y-direction, and 3.8 mm in the z-direction. While these errors are minimal, they remain within the acceptable range for the system's performance.

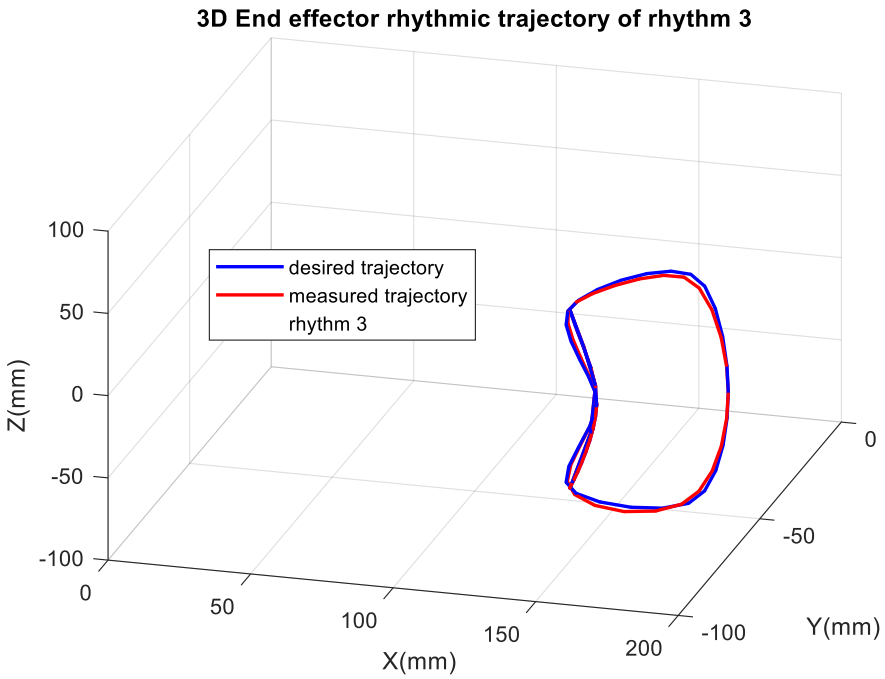
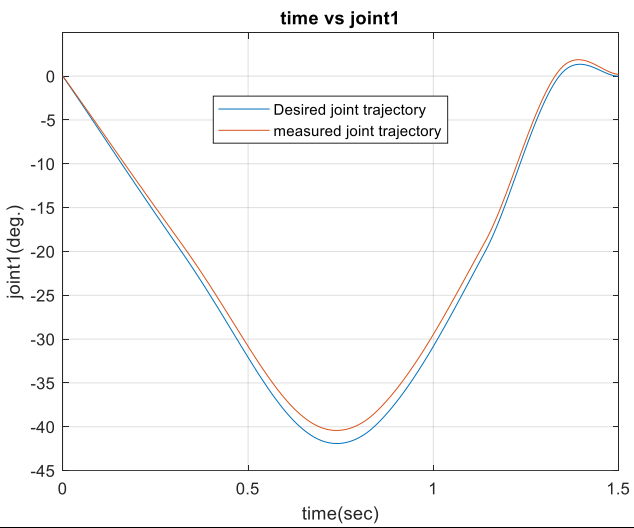
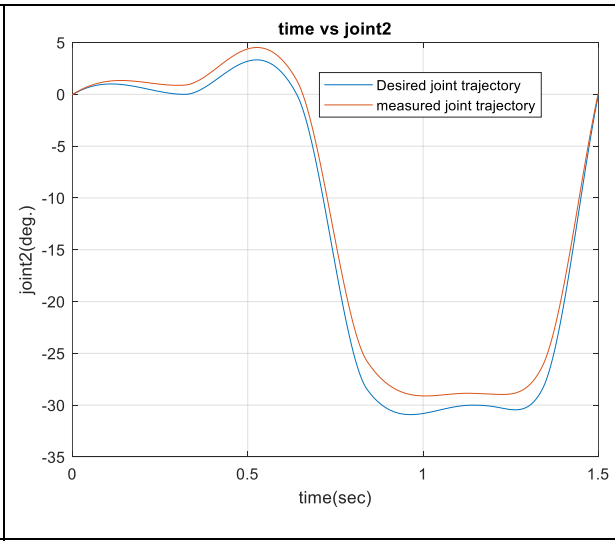


Figure b-2: end effector rhythmic trajectory three under disturbance

e. Simulation result of joint space input rhythmic trajectory three: under disturbance



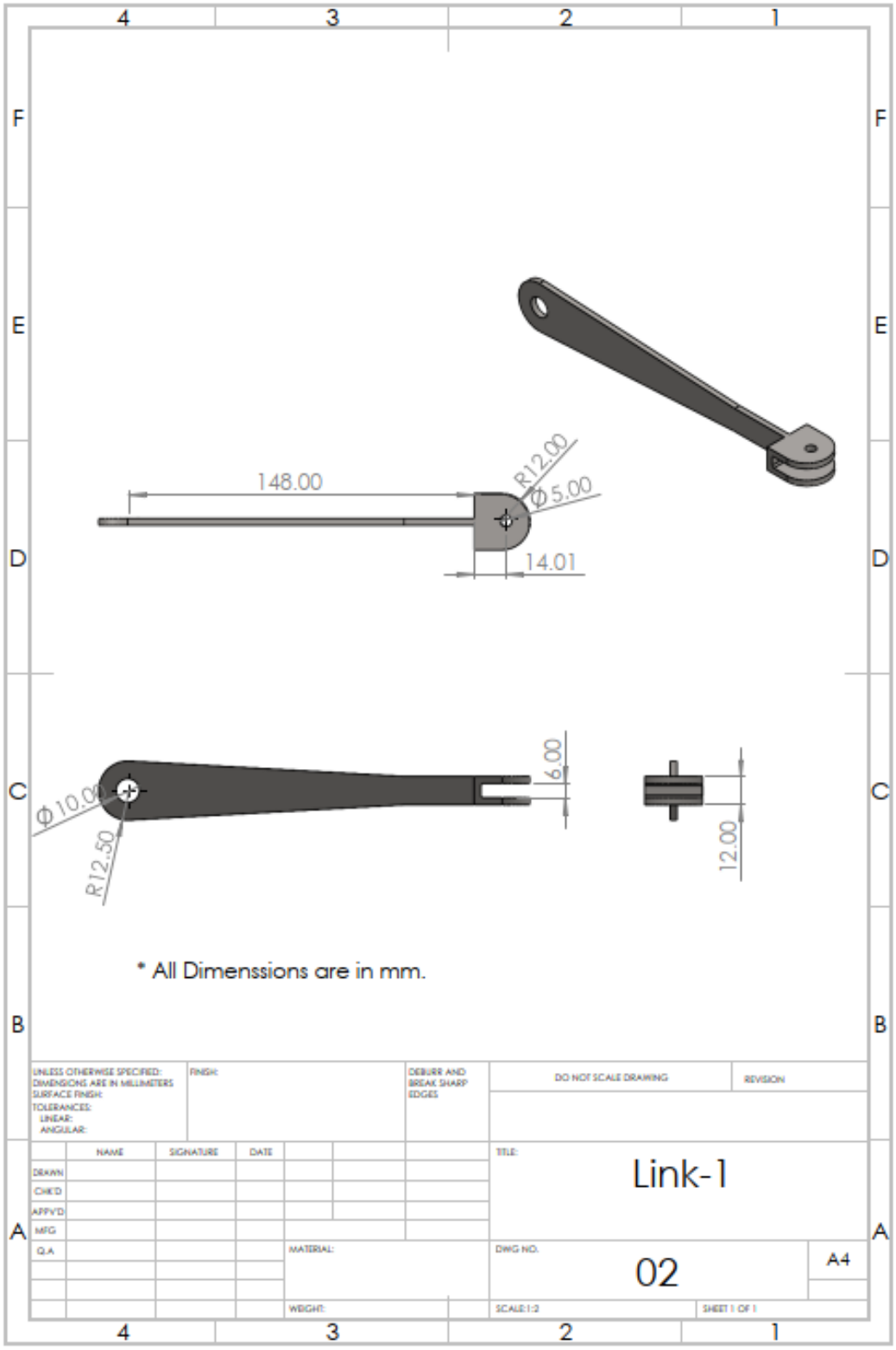
Joint_1: joint space trajectory rhythm three

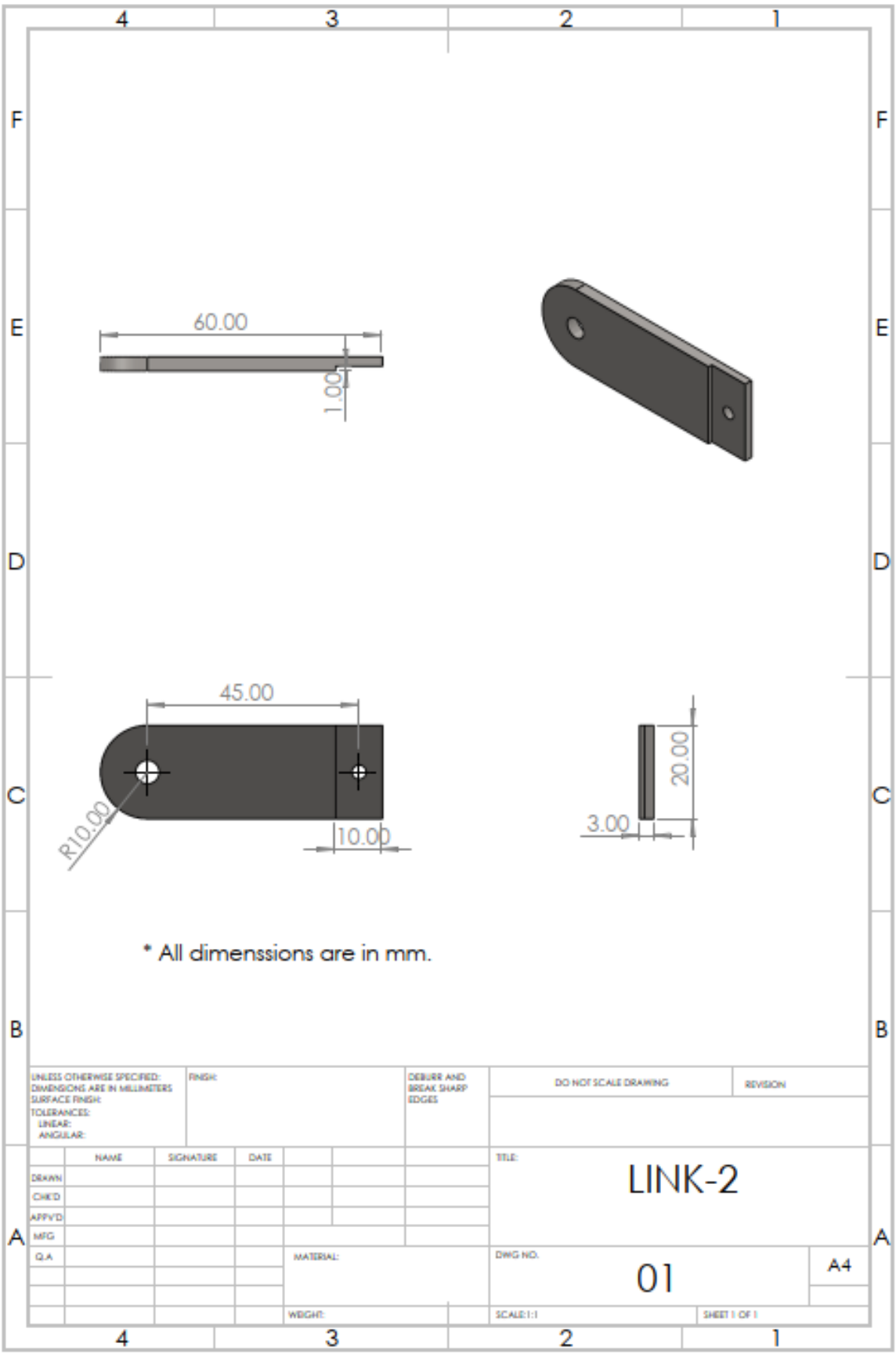


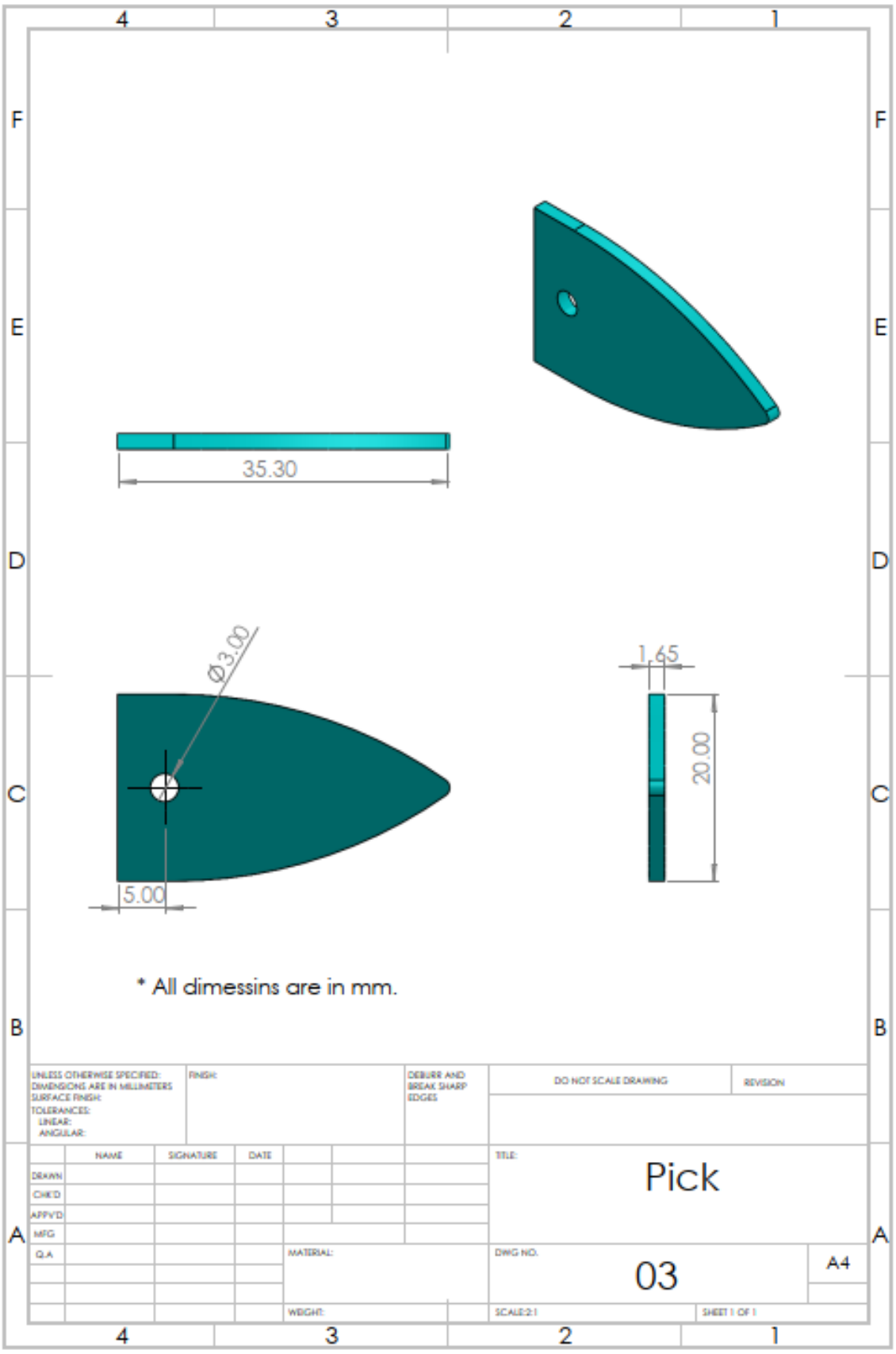
Joint_2: joint space trajectory rhythm three

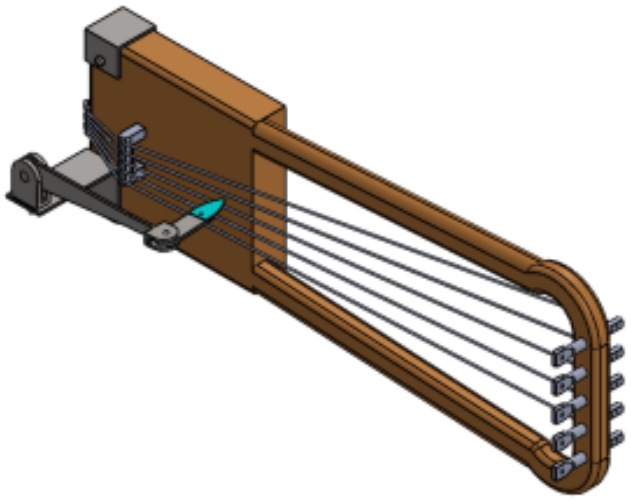
The joint space trajectories of Joint 1 and Joint 2 for Rhythm 3 exhibited errors of 1.5 degrees and 2.75 degrees, respectively, under disturbance. These errors are within the acceptable range for the system's performance.

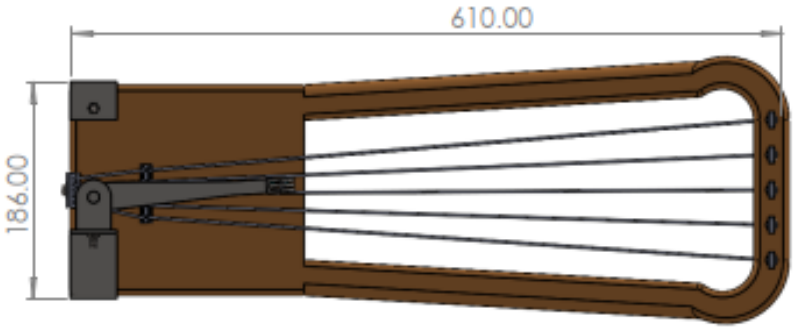
APPENDIX-C: DRAWING




4
3
2
1
F
E
D
C
B
A
60.00
1.00
45.00
R10.00
10.00
20.00
3.00
* All dimenssions are in mm.
UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN MILLIMETERS
SURFACE FINISH:
TOLERANCES:
LINEAR:
ANGULAR:
FINISH:
DEBURR AND BREAK SHARP EDGES
DO NOT SCALE DRAWING
REVISION
NAME
SIGNATURE
DATE
DRAWN
CHK'D
APPV'D
MFG
Q.A
MATERIAL:
WEIGHT:
TITLE:
LINK-2
DWG NO.
01
A4
SCALE:1:1
SHEET 1 OF 1


4
3
2
1
F
E
D
C
B
A
35.30
Ø3.00
5.00
1.65
20.00
* All dimessins are in mm.
UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN MILLIMETERS
SURFACE FINISH:
TOLERANCES:
LINEAR:
ANGULAR:
FINISH:
DEBURR AND BREAK SHARP EDGES
DO NOT SCALE DRAWING
REVISION
NAME
SIGNATURE
DATE
DRAWN
CHK'D
APPV'D
MFG
Q.A
MATERIAL:
WEIGHT:
TITLE:
Pick
DWG NO.
03
A4
SCALE:2:1
SHEET 1 OF 1





* All Dimessions are in mm.

UNLESS OTHERWISE SPECIFIED: DIMENSIONS ARE IN MILLIMETERS SURFACE FINISH: TOLERANCES: LINEAR: ANGULAR:	FINISH:		DEBURR AND BREAK SHARP EDGES	DO NOT SCALE DRAWING	REVISION

	NAME	SIGNATURE	DATE				TITLE:
DRAWN							RKS ASSEMBLY
CHK'D							
APPV'D							
MFG							
Q.A				MATERIAL:			DWG NO. 04 A4
				WEIGHT:			SCALE:1:10 SHEET 1 OF 2