 Volume 1 (1982), Number 1

CSC 473 Final Paper

Tai Kilpatrick

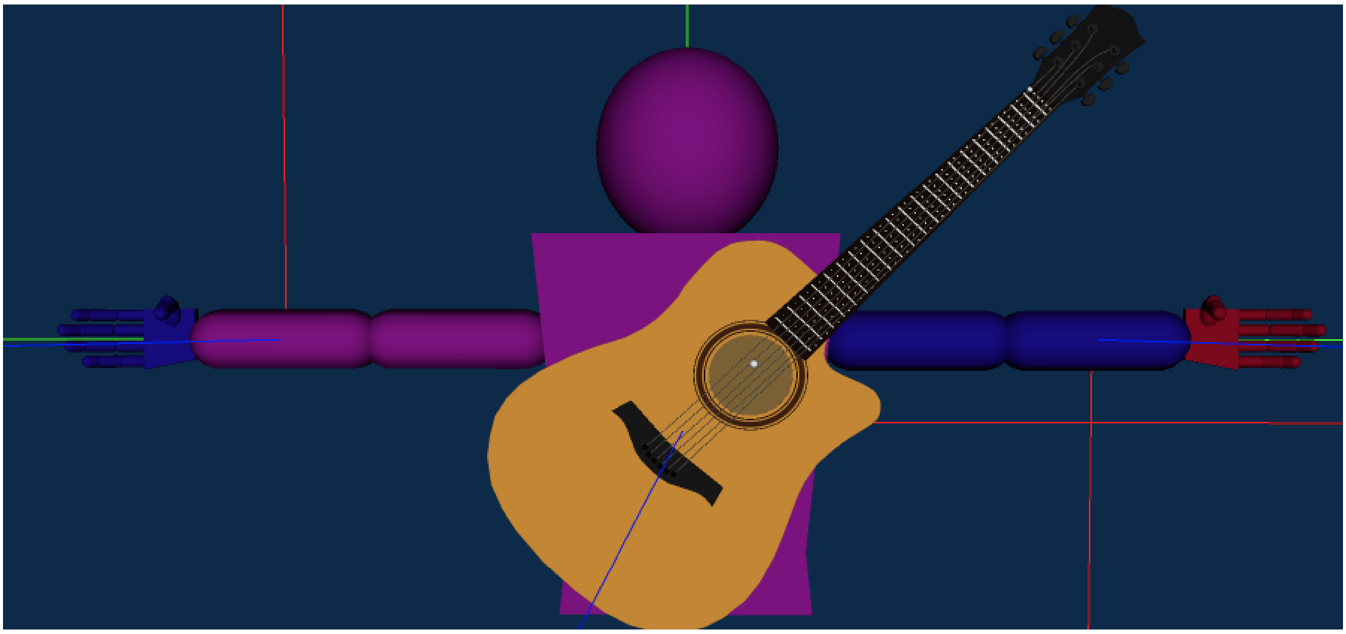

Figure 1: *A view of the guitarist*

Abstract

The "Animating Guitar" project aims to simulate realistic guitar playing through lifelike hand motions. Using Forward and Backward Reaching Inverse Kinematics (FABRIK), the system maps hand positions on a guitar's strings and frets. The method allows for precise articulation control, optimizing finger placement to reproduce music accurately. Evaluation focuses on the system's ability to recreate music complexity, from basic melodies to chord sequences. Based on previous research on hand animation and IK solvers, the project integrates findings from Handrix and FABRIK papers. The project uses a shortest path and cost minimization optimization algorithm to determine the movement needed from the fingers, hand and arm in order for the player to the inputted notes. The final deliverable will be a demonstration video showcasing the project "playing" multiple different musical pieces.

CCS Concepts

• ***Computing methodologies*** $\rightarrow$ ***Animation;***

1. Introduction

The motions of the human hand are incredibly complex, from playing musical instruments to typing on a keyboard, the complex movements of the hand is an important topic for researchers in animation and related fields. This fascination drives ongoing efforts to create animation systems that realistically recreate the nuances of hand motions.

The "Animating Guitar" seeks to contribute to this domain, focusing on accurately simulating the hand movements of a guitar player. Modeling a player, hands, and guitar is the first step towards recreating this musical performance, where the guitar's fretboard can be conceptualized as a 21x6 grid of note positions. To animate the player's movements with precision, Inverse Kinematics (IK) methodologies are crucial.

Amongst various IK algorithms, the Forward and Backward Reaching Inverse Kinematics (FABRIK) method stands out due to its computational efficiency, rapid convergence, and intuitive approach. FABRIK determines new joint positions by tracing a path along a line, a principle that aligns well with the natural arc of the fingers. Furthermore, FABRIK's ability to incorporate rotational constraints is particularly advantageous for modeling the human hand, where finger abduction (separation) is naturally limited. Its tendency to follow the shortest path between an origin position and a target position further mimics the efficiency of human movement. When animating hand motions, particularly those involving multiple end effectors (like individual fingers), the FABRIK system offers distinct advantages. When completed the project will be able to simulate a real guitar player, by realistically selecting which fingers to move and when to move them, along with taking realistic joint angles and movement paths.

2. Related Works

The "Animating Guitar" project draws significantly from the work of George ElKoura and Karan Singh in their paper, "Handrix: Animating the Human Hand" [ES03]. Their research aimed to create a system that realistically replicates guitar playing animations, specifically targeted at modeling music learners. Several key contributions from Handrix provide inspiration for this project.

Hand Modeling: ElKoura and Singh developed a comprehensive hand model designed to capture the intricate interdependence between fingers. While this project adopts a simplified model their research provides foundational knowledge on the complexities of hand articulation.

Optimal Finger Selection: Handrix includes algorithms for determining the most suitable finger to play a particular note on the guitar. This aligns directly with the decision-making process the "Animating Guitar" project needs to emulate for realistic playing simulations. Handrix also provides different models to simulate varying perceived skill of the animated guitar player.

Geometric Cost Function: Optimizing the player's hand movements provides a way to evaluate potential paths and gestures. The concept of a geometric cost function, as introduced by ElKoura and Singh, provides a framework for quantifying the efficiency and realism of different movement options. This algorithm allows for the smooth transfer between individual fretted notes and chords, along with aiding complicated transitions that require the optimal position for multiple fingers.

Beyond the Handrix project, the paper "FABRIK: A Fast, Iterative Solver for the Inverse Kinematics Problem" by Aristidou and Lasenby [AL11] offers crucial insights and directly influences the choice of IK algorithm for "Animating Guitar". Let's break down why FABRIK is particularly well-suited:

Multiple End Effectors: The FABRIK algorithm handles IK problems with multiple end effectors, mirroring the individual control we aim to achieve over each finger in the hand model.

Rotational Constraints: FABRIK's ability to incorporate rotational limits is invaluable when simulating the human hand, where joints have specific ranges of motion.

Visual Realism: An explicit goal of the FABRIK algorithm is to produce visually convincing end poses. This focus on aesthetic results benefits the "Animating Guitar" project, where the believability of the animation is crucial.

Optimization: FABRIK provides guidance on how to optimize the IK system for computational efficiency, a consideration for potential real-time applications of the project.

This project builds upon the foundational research explored in the Handrix and FABRIK papers. While specific implementations or models might be adapted to suit the project's scope, these works provide a robust framework and address many of the core challenges inherent in simulating realistic hand movements for guitar playing.

3. Overview

The main components of the project are hand modelling, inverse kinematics, the shortest path algorithm. The first to be defined is the hand and player modelling, where the components to be considered are the shoulder, forearm, wrist and fingers. Each arm component needed to be set up with its full degrees of freedom in order for the IK to be implemented. The IK is used to animate the finger and arm movements of the player, and the shortest path algorithm is the main task which will recreate the motions of a real player playing guitar, by optimizing the finger movements. The main simulator which will drive the animation is a simple music simulator, which takes in a sequence of notes and their values, and map the notes onto the fret board to input into the IK system. Once the sequence of notes has been given, the shortest path algorithm runs using the current positional data of the hands and compute the optimal fingering pattern and sequence. The length of the note is used as a timer in between the switching of notes.

3.1. Modelling

The modelling of the hands is quite complicated as was done from fingers down to the body of the player. The following joints needed to be considered.

Fingers. Each hand has 5 fingers with three knuckles each, which are all capable of flexion/extension and abduction/adduction, granting 2 degrees of freedom per finger. As adduction and abduction are only available to the bottom joint of the finger, the second and third joints are only subject to flexion and extension. In order to more realistically model the hand, a simplified interdependence model is introduced. As when you try to flex a single finger, the others are all affected and flex slightly, this can be represented as the joint angle constraint $\theta_{\text{DIP}} = \frac{2}{3}\theta_{\text{PIP}}$. Where DIP is the third joint in the finger, and PIP is the middle joint. This constraint helps keep the range of poses in the scope of reality. An illustrated version of rotations is displayed in Figure 2.

Wrist. The wrist has a total of **3 degrees of freedom**, with extension/flexion, supination/pronation, and ulnar/radial deviation.

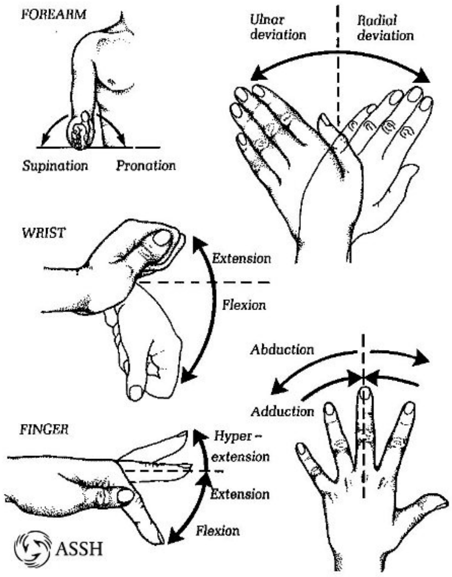

Figure 2: *An anatomical reference for hand movement.*

```
e|----------------------------------------------------------------------------------------------------|
B|------------------------------------8-8-----------------------------------------------------8b9b10-|
G|-----5-7-7b------5-7-7b-----5-7-7b------7p5------------5------5-7-7b------5-7-7b-----5-7-7b--------|
D|-7-7---------7-7---------7-7---------------7p5-----5-7----7-7---------7-7--------7-7---------------|
A|-----------------------------------------------7p5--------------------------------------------------|
E|----------------------------------------------------------------------------------------------------|
```

Figure 3: *A sample guitar tab.*

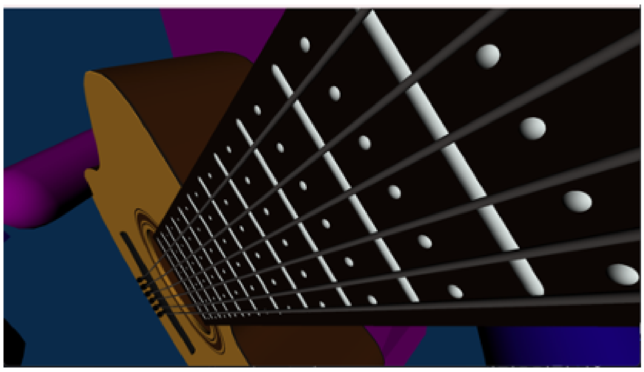

Figure 4: *A view of the guitar neck.*

Supination and pronation is technically a forearm movement, but it has been defined with the hand for this case.

Shoulder and Forearm. The shoulder and forearm share the same definitions in degrees of freedom, with each being subject to flexion/extension, adduction/abduction and internal rotation for a total of 3 degrees of freedom each.

3.1.1. The Guitar

Before we define the playing methods, we will first define the layout of the guitar. The strings are indexed from 1 (representing the highest pitch and lightest gauge) to 6 (representing the lowest pitch and heaviest gauge). To produce different notes, the musician presses down on the strings at specific positions along the neck, known as frets (see Figure 4). These frets are numbered sequentially starting from the nut, with the first fret being labeled as fret 0 and increasing in order along the neck. In terms of playing, the dominant hand(the right hand in this case) is responsible for strumming or plucking the strings, while the fretting hand changes the pitch by pressing the strings against the frets.

Tablature is one of the most straightforward ways of representing playing notes on guitar(see Figure). In this notation, six horizontal lines represent the guitar strings (numbered 1 to 6 from top to bottom). Notes are played sequentially from left to right along these lines. The fret numbers are positioned on the lines to indicate which fret on the corresponding string should be pressed. As guitar tabs do not state the length of each note, a separate sequence will be inputted, marking the length of each note in the sequence.

3.2. Inverse Kinematics

In order to simplify the IK of the system, the IK between the shoulder and forearm will be treated independently from that of the wrist and hand. Where movement of the fingers will be prioritized, and the player will only adjust the arm if the frets are out of reach from the fingers and wrist. As the fret board positions are easily mapped, it was easy to determine the end effector for the left hand. For the players right hand, the hand mostly stays in place while the fingers pluck individual strings, with the thumb playing the two lowest strings, and the other fingers playing the remaining strings. FABRIK is used to handle the inverse kinematics computation, which iteratively converges on the new position given the joint positions, target positions and distance between the joints. This is done by following a point along the line by moving the end effector towards the target position. In each iteration the joints are adjusted to reduce the distance between consecutive joints. This forward reaching process continues until the end effector reaches the target position or until a certain tolerance level is reached. After the forward reaching phase, FABRIK reverses the direction of movement and adjusts the joint positions from the root towards the end effector. This process ensures that the entire structure remains consistent with the desired end effector position while maintaining the constraints imposed by the joint limits and other factors. The entire process is illustrated in Algorithm 1

3.3. Finger Selection

The process of optimally using fingers is the most important element in regard to recreating how a human would play guitar or any instrument. This process has been greatly expanded from [ES03] where they propose a number of algorithms and equations. The first is the algorithm to decide which finger to use for a given fretting. This function can be adjusted by α_i and β_i to adjust the weighting for each finger. The natural arc of the hand makes it such that the index and middle fingers can reach forwards easier than can the ring and little finger. Setting $\alpha_1 < \beta_1$ would thus indicate that the index can move forwards easier than it can move backwards along the neck of the guitar. The choice for α_i's and β_i's is arbitrary, al-

Algorithm 1 A full iteration of the FABRIK algorithm

Input: The joint positions p_i for $i = 1, \ldots, n$, the target position t, and the distances between each joint $d_i = |\mathbf{p_{i+1}} - \mathbf{p_i}|$ for $i = 1, \ldots, n-1$.
Output: The new joint positions p_i for $i = 1, \ldots, n-1$.

```
% The distance between root and target
dist ← |p_{i+1} − p_i|
% Check whether the target is within reach
if dist > d_1 + d_2 + ... + d_{n-1} then
    % The target is unreachable
    for i = 1 to n − 1 do
        % Find the distance r_i between the target t and the joint position p_i
        r_i ← |t − p_i|
        λ ← d_i / r_i
        % Find the new joint positions p_i
        p_{i+1} ← (1 − λ_i) p_i + λ_i t
    end for
else
    % The target is reachable
    b ← p_1
    difA ← |p_n − t|
    while difA > tol do
        Stage 1: Forward Reaching
        p_n ← t
        for i = n − 1 to 1 do
            r_i ← |p_{i+1} − p_i|
            λ_i ← d_i / r_i
            p_{i+1} ← (1 − λ_i) p_{i+1} + λ_i p_i
        end for
        Stage 2: Backward Reaching
        % Set the root p_1 to its original position
        p_1 ← b
        for i = 1 to n − 1 do
            r_i ← |p_{i+1} − p_i|
            λ_i ← d_i / r_i
            p_{i+1} ← (1 − λ_i) p_i + λ_i p_{i+1}
        end for
        difA ← |p_n − t|
    end while
end if
```

though to replicate an average hand's finger strength, the following constraints were used: $\alpha_1 < \beta_1$, $\alpha_2 < \beta_2$, $\alpha_3 > \beta_3$ and $\alpha_4 > \beta_4$.

$$f_i(\mathbf{c}) = \begin{cases} \lambda_i \alpha_i ||\mathbf{d}||^{m_i} & \text{if } d_y \geq 0, \\ \lambda_i \beta_i ||\mathbf{d}||^{m_i} & \text{if } d_y < 0 \end{cases} \tag{1}$$

This algorithm is used to calculate the cost of the moving the finger i by the displacement vector $\mathbf{d} = \mathbf{c} - \mathbf{p_i}$, where $\mathbf{c}$ is the target position and $\mathbf{p_i}$ is the current position of the finger. Note that this equation is in the reference frame where the x-axis is parallel to the frets, the y-axis is along the guitar neck, and the z-axis is the normal of the fret board. The parameter λ_i allows the user to prioritize specific fingers in the algorithm. A lower λ_i for a given finger indicates a preference for its use when multiple solutions exist. For instance, setting $\lambda_1 < \lambda_2$ would favor the index finger over the middle finger. The exponents m_i control the rate at which the cost function increases as a finger moves further from its comfortable or natural fret position. This allows for fine-tuning the penalty associated with extending each finger. Similar to the finger cost function, the wrist cost function f_{wrist} reflects the inherent cost associated with moving the wrist away from its natural position. This factor influences the algorithm's overall hand posture optimization. These customizable cost functions are integrated into the core cost-minimization algorithm, which determines the most efficient finger and wrist configuration to play a given musical input. Equation 1 is used in the *resolveFinger* algorithm (Algorithm 2) to determine the optimal finger, although with the following modification:

$$g_i = \begin{cases} \text{L} & \text{if finger } i \text{ is used,} \\ f_i & \text{otherwise} \end{cases} \tag{2}$$

Where L is a very large cost, preventing the finger from being chosen and allowing a less optimal finger to be used when performing actions such as playing chords.

Tying these equations together in an algorithm, we have our final cost-minimizing algorithm. Which determines the position $\mathbf{c}$ that minimizes:

$$\chi(\mathbf{c}) = f_{wrist}(\mathbf{c} + \sum_{t=T}^{T+W} resolveFingers(\mathbf{c}, t)) \tag{3}$$

This equation is define over a moving window W which is a user-definable variable determining how far ahead the player is reading ahead in the music. The minimum cost is solved for at each time step T, and is easily solved as there is only a limited number of fret positions on a guitar. Function 2 is then used in Algorithm 2 which iterates through each string to determine the best finger and position to use.

Algorithm 2 *resolveFingers*(c, t)

```
Initialize finger positions
totalCost ← 0
for each string s from high to low do
    if n > 0 then
        cost ← min(g_i(c))
        i ← argmin(g_i(c))
        if i is not used then
            play note at s with finger i
            mark i as used
            totalCost ← totalCost + cost
        end if
    end if
end for

return totalCost
```

The initial step involves setting up the finger positions, by flexing them approximately one third of their full range. While this initial

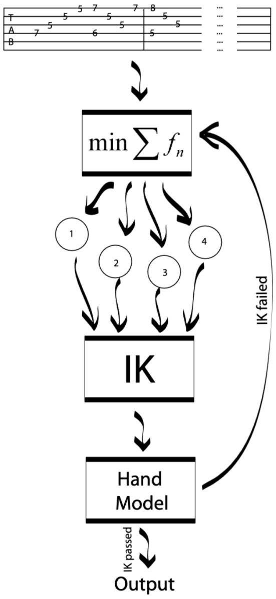

Figure 5: *Full system architecture[ES03].*

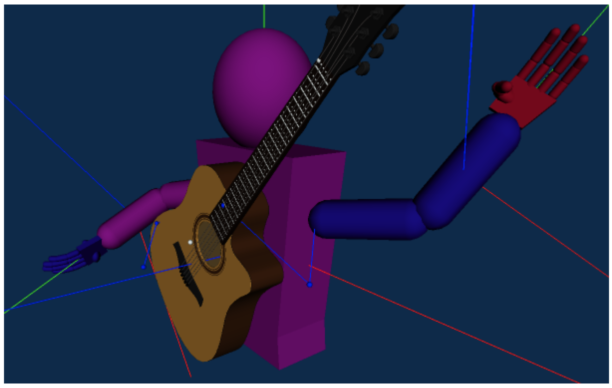

Figure 6: *A view of the attempted IK implementation, the line between the players armpit and guitar is the points output from FABRIK, and represents the proper angles that should have been taken.*

positioning can be adjusted by the user, it often feels the most natural and comfortable for the fingers. Subsequently, we address two critical factors that greatly influence the realism of the outcome: wrist posture and potential finger collisions. These aspects are corrected once the finger positions have been established.

To determine wrist extension and flexion, a heuristic approach is used. It's assumed that the wrist extends when playing high strings and flexes when playing low strings. We identify the highest and lowest strings played at each timestep, denoted as s_{hi} and s_{lo}, respectively, during the execution of Algorithm 1. User-provided minimum and maximum extension/flexion angles for the wrist, denoted as θ_{min} and θ_{max}, are then blended using a computed blend factor $b = \frac{s_{hi}+s_{lo}}{12b}$. A similar approach can be adopted for handling wrist pronation/supination. In the case of a collision between fingers, a check is run after the fingers are placed to determine if there are any overlapping radii, in which case the finger on the lower string is adjusted so that the radii are not overlapping. After this process, the data is then fed into the FABRIK algorithm in order for the playing to be animated. See Figure 5 for the full process.

4. Evaluation

While a fully functional implementation of inverse kinematics presented challenges, the "Animating Guitar" project provides insights into the complexities of simulating realistic hand movements for music playing. This section discusses the key evaluation metrics. Unfortunately the system is unable to be properly evaluated as it is not fully functional. After a long period of attempts a successful implementation of inverse kinematics could not be properly realized. The issue came from the change in coordinate systems of the arms from the world co-ordinate system, and a proper conversion of co-ordinates and angles could not be found.

Figure 6 shows a snapshot of the inverse kinematics and its issues. While the positions were being correctly calculated, the angles between those positions were not properly being applied to the 3D model, resulting in a bottleneck in development, and not providing enough time to implement the third stage of the project which involves the shortest path and finger optimization.

To begin with, a core metric would be musical accuracy. This would involve determining the system's ability to correctly position the fretting hand on the guitar fretboard to play the desired notes and chords. A sequence of simple melodies and chords would serve as input, followed by a visual analysis of whether the simulated fingers accurately land on the correct frets and strings. A success rate could be calculated based on the percentage of correctly played elements, and whether or not the hands were able to properly reach the notes in time. However, without a full IK implementation, a qualitative visual analysis is be the primary approach. Regardless, this metric directly measures how effectively the system replicates the core functionality of a guitar player, pinpointing areas for refinement.

Furthermore, assessing the visual realism of the simulated hand and arm animations is crucial. Creating test animations of common guitar playing patterns (e.g., single-note scales, arpeggios, simple chord changes) and comparing the movement of the system vs that of a real human playing guitar is the primary metric to assess this factor. Feedback would focus on finger movement fluidity, overall hand posture, and the transitions between different hand positions. Subjectivity in evaluation is a potential challenge, as methods of playing vary between people and can be largely dependent on their hand size and strength. However, this metric is essential for crafting visually appealing and believable animations for applications in music and entertainment.

Lastly, evaluating the system's computational efficiency would assess its suitability for potential real-time usage. Implementing profiling tools to measure computation time for IK calculations, hand model updates, and rendering is crucial. The performance needs to be analyzed under varying levels of complexity to determine if the system can achieve frame rates sufficient for real-

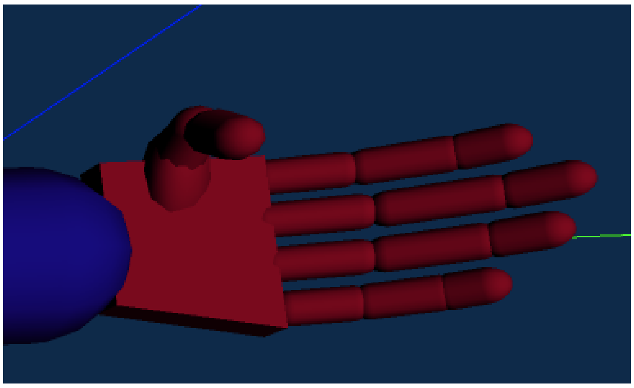

Figure 7: *A closeup view of the hand model.*

time interaction or if further optimization is needed. Even without a complete IK solution, profiling specific aspects can provide valuable insights into potential performance bottlenecks. Efficiency directly impacts the practicality of the system in applications like interactive music learning tools or virtual performances.

4.1. Discussion

This project displayed the intricate challenges of replicating the fine motor control of human hands within the context of guitar playing. Even though the incomplete IK implementation limits proper evaluation, the effort and development process uncovered valuable insights and highlighted several key areas for future development.

Hand Model Complexity: Initial assumptions about a simplified hand model with linear interdependence between fingers might have proved insufficient for replicating the subtle nuances of a real guitarist's hand. Researching more sophisticated hand models, potentially incorporating insights from anatomical studies or motion capture data, could yield significant improvements in the realism of the simulated movements. Figure 7 provides a look at the hand model in detail. The hand model had varying issues including a lack of actual anatomical data in average hand dimensions.

Qualitative Feedback: The importance of visual assessment, despite the incomplete IK implementation, emphasizes the need for qualitative feedback mechanisms. Collaborating with experienced guitarists would provide invaluable insights into nuances of hand posture and finger transitions that go beyond purely quantitative measurement.

4.2. Future Work

This project sought to follow after some previous developments in the animation of musical instruments. Some potential directions for further development are:

Advanced Hand Modeling: Investigating more complex hand models with greater degrees of freedom and realistic interdependence between fingers would enhance the simulation's ability to represent the skill levels of different guitar players.

Integrating Strumming/Picking Hand: Extending the model to include both the fretting and strumming/picking hands is essential for simulating a complete guitar performance. This demands the development of another IK system and synchronization between the two hands.

Music-Driven Animation: Incorporating audio analysis of music pieces would allow for better quantitative evaluation, as the system could only output the proper sound if the correct IK target is hit, better simulating the real life instrument.

Applications to other tasks: While the motions of playing a guitar are relatively unique, if the system was implemented correctly, it would allow for easy transitions to other tasks which focus on dexterity, such as playing a different musical instrument or typing on a keyboard as once the positions for each action are mapped, the system would then be able to perform the task with relatively little accommodation needed.

This project revealed the inherent challenges of achieving realistic hand animation for complex tasks like playing a musical instrument. The obstacles encountered highlight the need for sophisticated hand models, robust coordinate system handling, and a further understanding of IK implementations. Despite its current limitations, it allows for many future developments.

5. Conclusion

The "Animating Guitar" project explored the challenges of accurately simulating guitar playing through realistic hand motions. Drawing inspiration from research in human hand animation and inverse kinematics (IK) techniques, the project integrated findings from the Handrix and FABRIK papers. Employing the Forward and Backward Reaching Inverse Kinematics (FABRIK) method, the system successfully mapped hand positions to specific frets and strings on a guitar. Additionally, a shortest path and cost-minimization algorithm was developed to optimize finger placement, hand position, and arm movements to recreate the nuances of a real guitarist.

While the project demonstrated promise in simulating the motions of guitar playing, certain limitations existed. The simplified interdependence model for finger movements might benefit from greater complexity to reduce the robotic look of the movement. Furthermore, the focus on fretting hand motions could be extended to include the strumming/picking hand for complete guitar playing simulation.

Future research could investigate more advanced hand models to increase the range of achievable articulations. Exploring techniques for accurately transitioning between chords and finger-picking patterns would expand the system's musical capabilities. In addition, integrating the system with real-time audio playback or music synthesis tools would provide for better assessment of the simulation, to determine if the simulation is hitting the correct positions on the guitar.

In conclusion, the "Animating Guitar" project provided a solid foundation for guitar animation. By addressing the identified limitations and expanding the scope of the system, there is vast potential for applications in music education, virtual performance, and the development of engaging interactive music experiences. In future development of the project, a simplified modelling method will be used to avoid the co-ordinate system difficulties that are currently present, which will open up development to the later parts of the proposed project.

References

[AL11] ARISTIDOU A., LASENBY J.: Fabrik: A fast, iterative solver for the inverse kinematics problem. *Graphical Models 73*, 5 (2011), 243–260.

[ES03] ELKOURA G., SINGH K.: Handrix: animating the human hand. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation* (2003), pp. 110–119.