

Digital simulation of the movement of an atlatl

Mapia

VO Van Nghia

PHAM Tuan Kiet

Abstract

One of the special weapons rarely seen nowadays in television or newspapers is the atlatl, or simply a spear-thrower. Developed in Europe over 30 000 years ago and by the Native North-Americans tribes 12 000 years ago, the atlatl is the first true “weapon system” consisting of a projectile (the dart) and a launching device (the atlatl).

The project aims to simulate and optimize the throwing action using a robotic approach. More specifically, we illustrate the movement using Python with the help of Pinocchio. The robot model will consist only of the upper body, which includes the hip, the shoulder, the elbow, and the wrist, in 2D, where only the rotation joint is used.

The work starts with designing a stick-version of the upper body and calculating the distance, the angle of each articulation, creating a “realistic” 2D-version of the throwing action, and finally optimizing it for the longest throwing distance of the projectile. From this project, we hope to have a better simulation of the action of throwing atlatl with the dynamic relation to have further studies about this kind of weapon.

1 Introduction

Maria (Historical evidences)

2 Preliminary

In the first section, we introduce some general notions and theories that we used for modeling a movement of a robot.

2.1 Trajectory planning

2.1.1 From human to robot

The movement of a robot is captured as a series of positions, or a trajectory. For example, the modeling of a walking human is shown below.

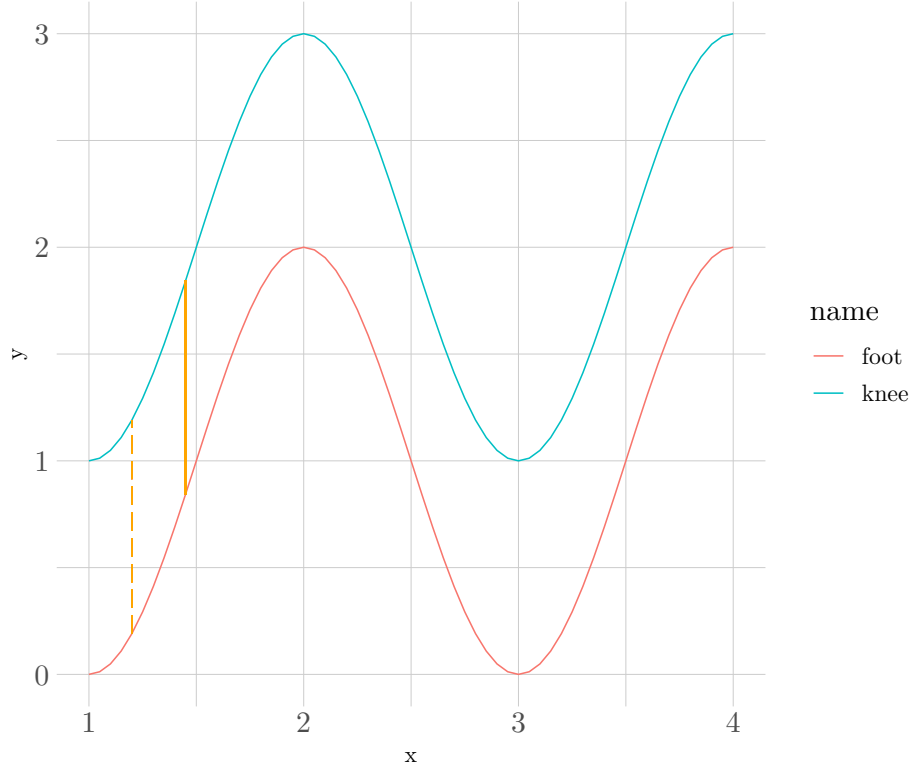


Figure 1: Knee and foot trajectory while walking

There are two curves, each represents a trajectory of our knee and foot while walking. For a better illustration, we also include an orange line which represents the leg. We could imagine a robot that is walking along the x -axis. With these trajectories, the knee and the foot of the robot will produce a human-like walking movement.

2.1.2 Trajectory interpolation

In the above example, the trajectory could be expressed as a variation of the \cos function. However, in reality, the desired trajectory is more complicated. Computing directly that function is not possible. Instead of doing that, we will only supply some positions at certain times, and interpolate the whole trajectory. To ensure a correct movement, two conditions must be satisfied:

- The interpolator must be smooth. Mathematically, we want the interpolator line is continuous and its first derivative is also continuous. This ensures that our robot will not perform any change in action too harshly.
- The monotonicity of the dataset (positions) being interpolated must be preserved. This keeps the interpolator from generating data that is out of range. (e.g the position of a foot is always greater or equal to 0).

For clarity, we will illustrate these 2 conditions below.

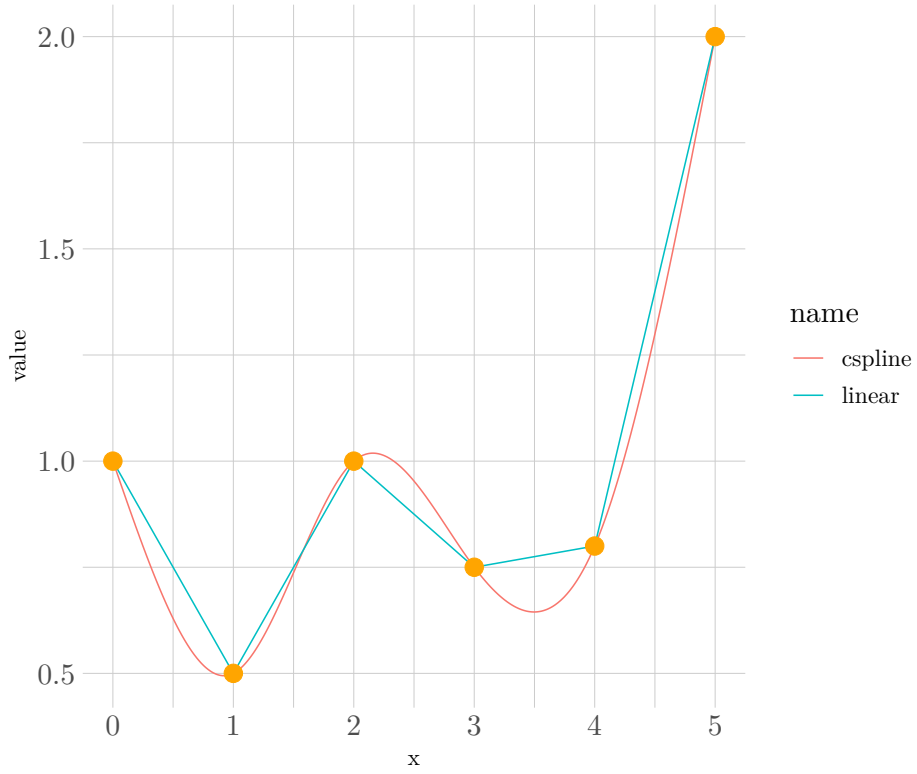


Figure 2: Smoothness of linear and cubic spline interpolation method

As in figure 2, the linear method only uses straight lines to connect the points together and has a zigzag pattern. On the other side, the cubic spline method is smooth and derivable. Since the graph of each function is also how our robot move its leg, we prefer the cubic spline method for a smooth movement.

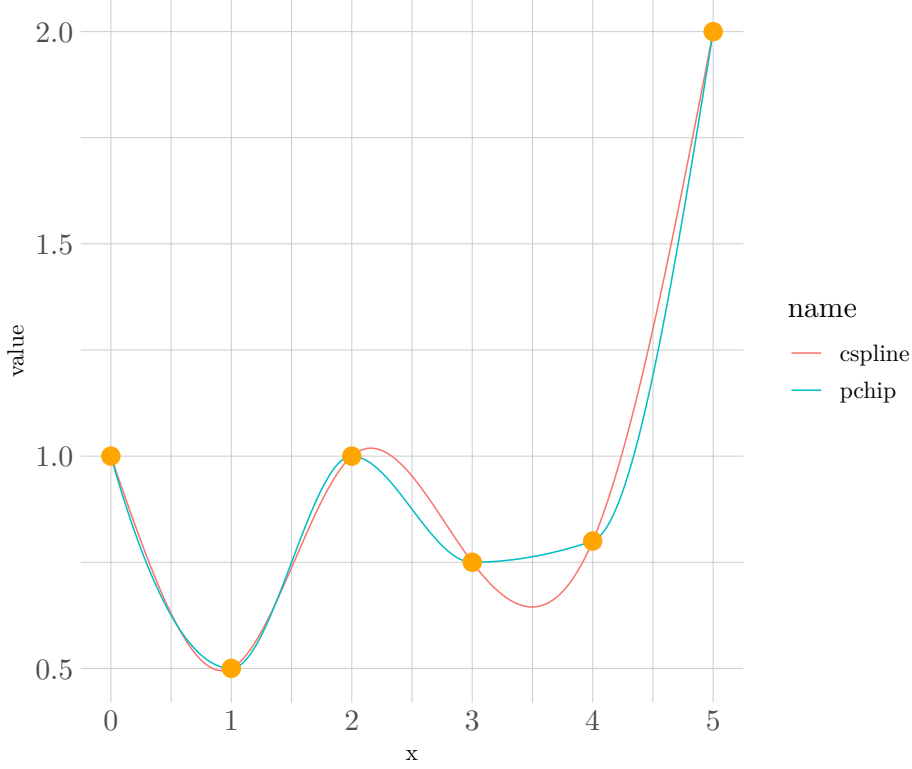


Figure 3: Monocity of cubic spline and pchip interpolation method

For the second condition, the figure 3 shows clearly that the cubic spline method does not respect the monocity between each pair of points while the pchip method (stands for Piecewise Cubic Hermite Interpolating Polynomial) does. For example, if we consider the interval between 3 and 4, we see that the pchip line is monotonically increasing while the cubic spline forms a valley.

The method that satisfy both conditions is PCHIP¹ and will be used for modeling our robot.

2.2 Configuration solver

A configuration q is a collection of parameters of a robot and the angle as well as position of each joint of our robot is tied to one unique configuration. We note that \mathbb{Q} is the set of all configurations (configuration space) of each robot, \mathbb{P} is the set of all positions of that robot. In addition, $f : \mathbb{Q} \rightarrow \mathbb{P}$ maps each configuration with its corresponding position, we have f is bijective. Given a trajectory T , we want to solve the problem below:

$$\text{For each } t, \text{ find } q \in \mathbb{Q} \text{ such that } f(q) = T(t) \quad (1)$$

Solving (1) analytically is generally impossible. Therefore, we transform it to a different optimization problem:

$$\text{For each } t, \text{ note } q_t^* \text{ the solution of (1), find } q \in \mathbb{Q} \text{ such that } \|f(q_t^*) - f(q)\| \text{ is minimized} \quad (2)$$

Solving (2) is equivalent to find a close approximation of q_t^* . We now only need to define the norm $\|\cdot\|$. Since the position of a robot is specified by the rotation and translation of its components and we might only need to control certain rotations and translations, given $\{p_1^*, \dots, p_n^*\}$ the desired positions at time t (either rotations and translations), we define the norm as follow:

¹https://en.wikipedia.org/wiki/Monotone_cubic_interpolation

$$\|f(q_t^*) - f(q)\| := \sqrt{\sum_{i=1}^n (p_i^* - p_i)^2}$$

As we do not know the analytical form of f or neither its gradient, we will use BFGS algorithm² to solve (2).

2.3 Directed angle

In order to express the translation and rotation easier, we also use directed geometry where anticlockwise angle (θ_+) are positive and the reference line is the vertical axis of the current frame.

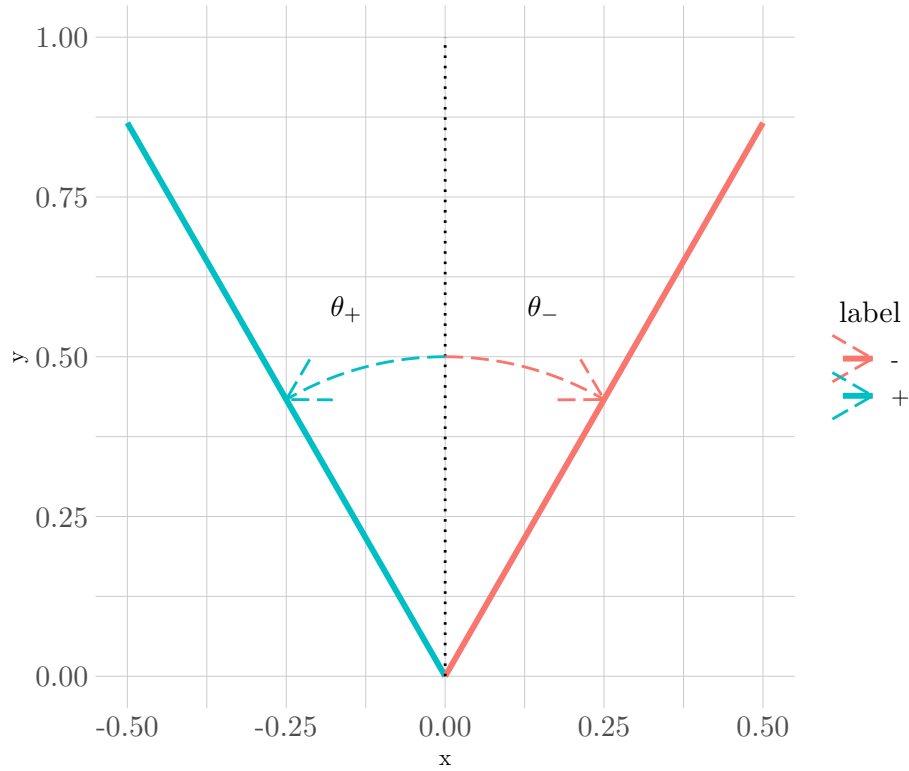


Figure 4: Directed angle

3 Modeling

In the second section, we apply previous observations and notions to our problem.

3.1 Robot structure

First, the structure of our robot is shown below.

²https://en.wikipedia.org/wiki/Broyden-Fletcher-Goldfarb-Shanno_algorithm

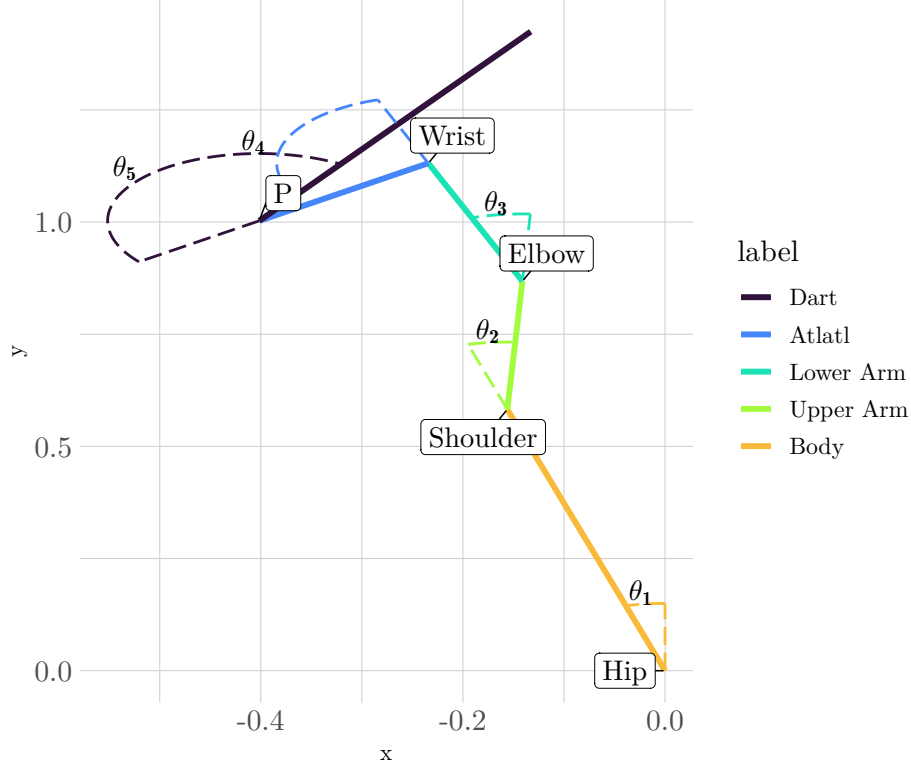


Figure 5: Structure of throwing robot

This structure models our upper body while throwing the atlatl. We have 5 rotation joints whose value are θ_i for $i \in \{1, 2, 3, 4, 5\}$.

3.2 Real life observations

In order to plan the trajectory of the robot, it is required to study the movement of an atlatl thrower in real life. The movement in real life can be divided into some stages; we first care about the position of the arm and the body, then the one of the wrist and finally, finishing the movement.

3.2.1 Arm and body

The core starts to rotate while the upper arm starts to flex at the height of the shoulder. In this way, the hand and the atlatl are moving until they are almost behind the head. The atlatl throughout this remains horizontal. The shoulder flex seems to be small at this point and the wrist must be rotated to keep the dart pointed at the target.

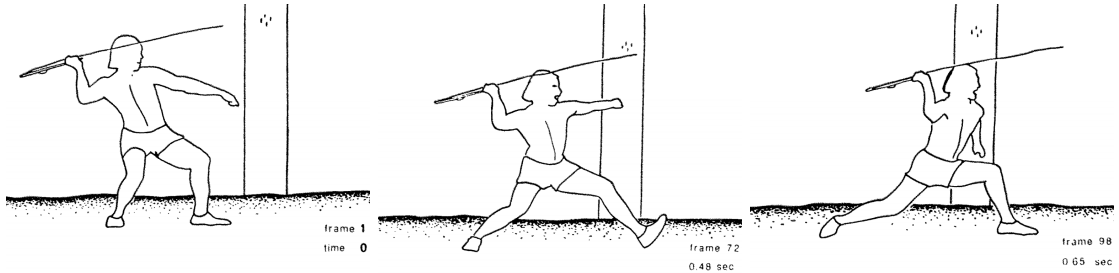


Figure 6: Position of arm and body

3.2.2 Wrist

Slightly before the hand reaches the back of the head, the hand and forearm begin to rise. Then as the hand is passing the head, the wrist should snap vigorously, giving the atlatl a vertical position and flicking the dart away. The dart will flex as the point remains aimed to the target, following a straight path. At the same time, the arm is being flexed straight out.

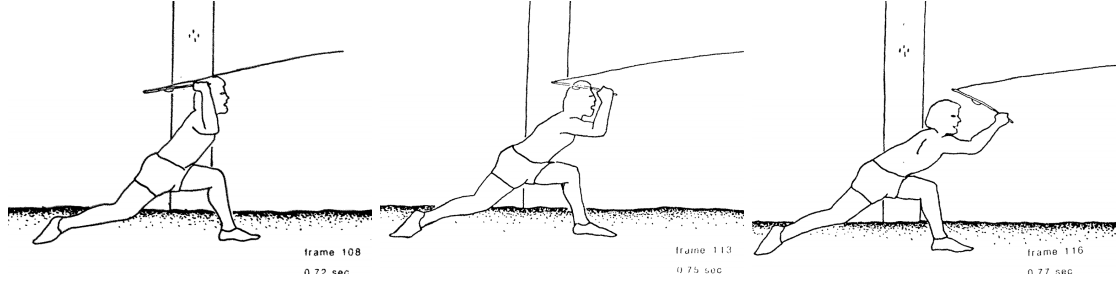


Figure 7: Position of wrist

3.2.3 Finishing

The dart is higher enough compared to the head when it is just about to leave the atlatl. The vertical position of the atlatl permits the dart to fly away. As the dart is leaving the atlatl, the thrower is leaning slightly forward. The movement finishes in a position where the arm that holds the atlatl crosses the opposite leg.

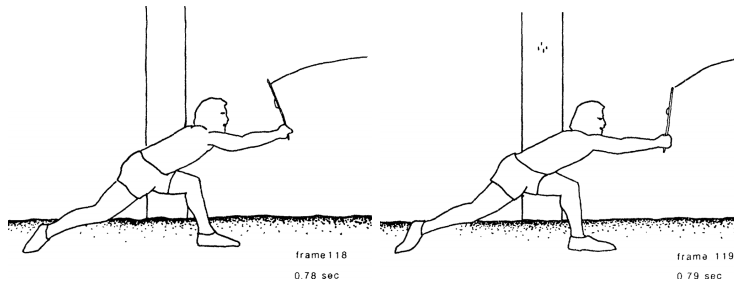


Figure 8: Finish position

3.2.4 The atlatl and the dart

Concerning the movement of the atlatl-dart system, Howard [1] states that when throwing a spear with the unaided hand it is necessary to make the hand travel in a straight line until the spear is released. When the hand nears its maximum outstretched position, the spear is then released. This allows to maintain control of the spear's line and avoid any downward motion. The same thing holds true for a dart that is thrown with the atlatl. Therefore, in addition to these observations above, we also impose another constraint: the distance between P and the ground is constant or the point P moves along a straight line.

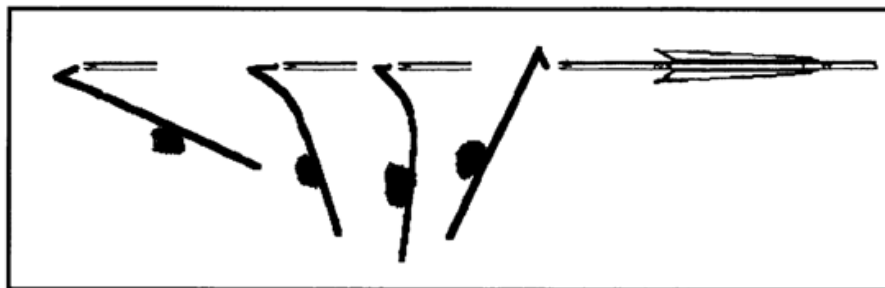


Figure 9: Atlatl straight line

3.3 Trajectory planning

3.3.1 Hip, shoulder and elbow

From the observations above, we first supply the rotations value of the first 3 joints: hip, shoulder and elbow to the interpolator and obtain the trajectory below:

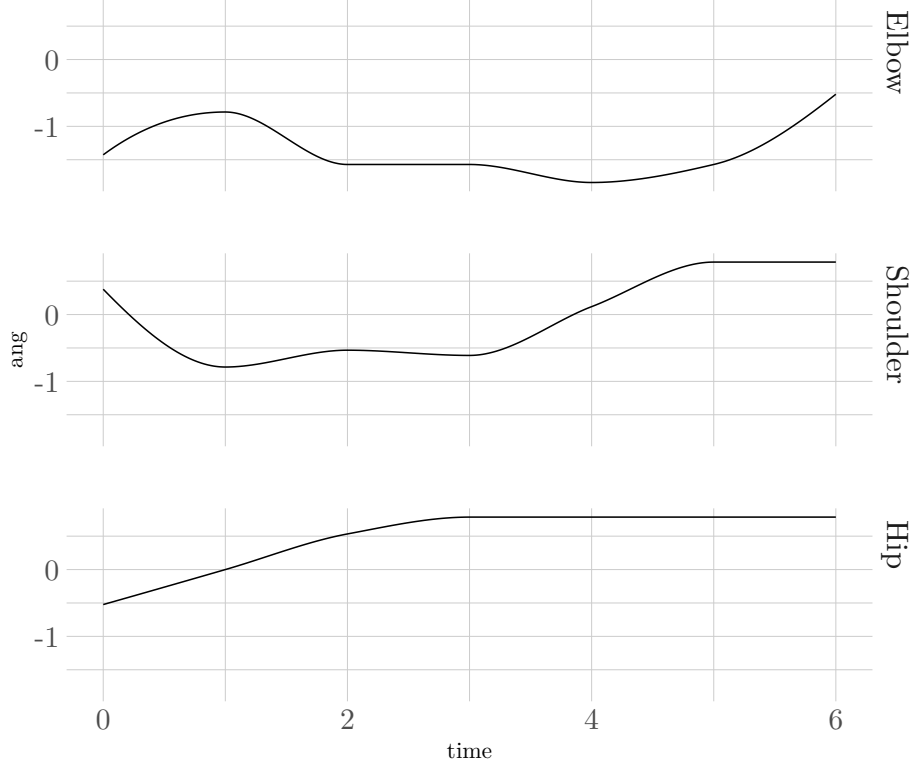


Figure 10: Trajectory of hip, shoulder and elbow

3.3.2 Wrist

For the wrist joint, we want to ensure the constant height as in the previous [section](#). If we note θ_i the angle with respect to the local frame (as in [figure 5](#)), Θ_i the angle with respect to the root frame, l_i the length of the i^{th} joint / part respectively and h_i the distance between the i^{th} joint and ground, we have the following relation:

$$h_k = \sum_{i=1}^k l_i \cos(\Theta_i) \text{ where } \Theta_j = \sum_{i=1}^j \theta_i \quad (3)$$

From [\(3\)](#), we can easily calculate θ_4 as we already know h_4 . Below is the trajectory of the wrist joint.

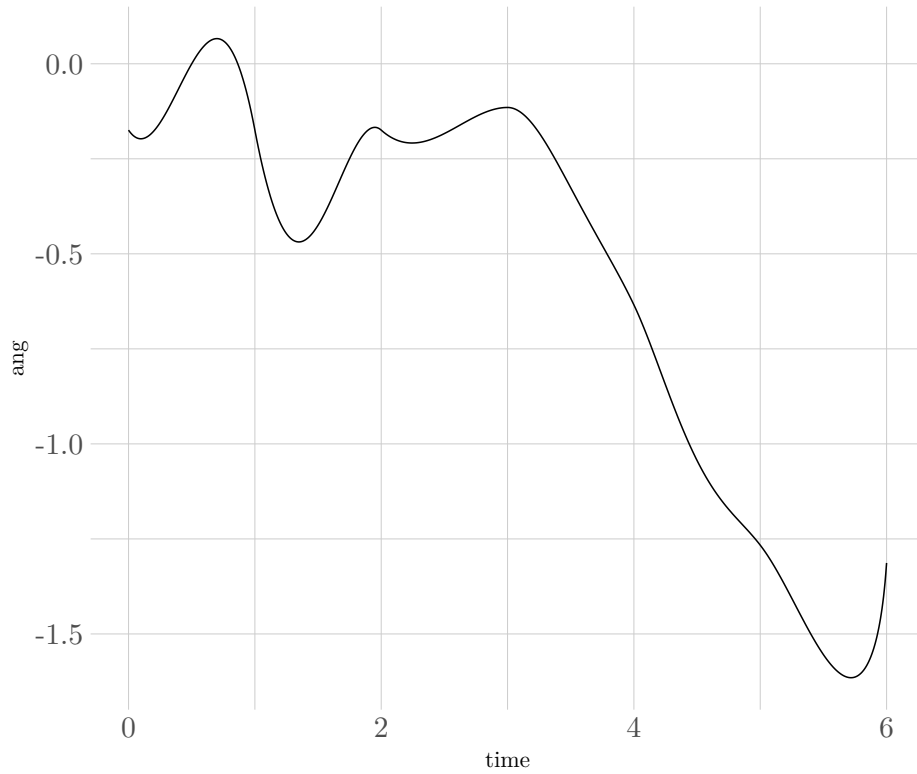


Figure 11: Trajectory of the wrist joint

3.3.3 P

Pham Tuan Kiet

3.4 Result

Vo Van Nghia & Pham Tuan Kiet

4 Improvement

Vo Van Nghia & Pham Tuan Kiet

5 Conclusion

Vo Van Nghia & Pham Tuan Kiet

References

- [1] Calvin D. Howard. “The Atlatl: Function and Performance”. In: *American Antiquity* 39.1 (1974), pp. 102–104. ISSN: 00027316. URL: <http://www.jstor.org/stable/279223>.