CIKAQ: USING A CUSTOM INVERSE KINEMATICS MODEL TO ANIMATE A QUADRUPED ROBOT DOG GAIT MODELED AS A SET OF CONTINUOUS BEZIER CURVES VIA INTERPOLATION

MIGUEL NEPOMUCENO, University of California Berkeley, USA
DAVID KIM, University of California Berkeley, USA
TRACY XIA, University of California Berkeley, USA
ADHAM ELARABAWY, University of California Berkeley, USA

An inverse kinematics model of a 2-degree-of-freedom robot leg is presented, including a model for a single foot path of the robot leg as a set of continuous bezier curves. The foot path is interpolated to obtain intermediate foot positions, which will then be passed through the inverse kinematics model in order to compute the angles of each joint in the robot leg. Output kinematics are animated in a custom visualizer to show how the robot leg moves through the foot path.

CCS Concepts: • **Computer systems organization** → **Embedded systems**; *Redundancy*; Robotics; • **Networks** → Network reliability.

Additional Key Words and Phrases: inverse kinematics

ACM Reference Format:
Miguel Nepomuceno, David Kim, Tracy Xia, and Adham Elarabawy. 2018. CIKAQ: USING A CUSTOM INVERSE KINEMATICS MODEL TO ANIMATE A QUADRUPED ROBOT DOG GAIT MODELED AS A SET OF CONTINUOUS BEZIER CURVES VIA INTERPOLATION. In . ACM, New York, NY, USA, 2 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

We've made significant progress on creating an animation visualizer for a walking quadruped model. To recap, what we originally aimed out to produce are the following:

- A sound mathematical inverse-kinematics model of a 2d robot leg (2 joints)
- For above model, functionality should return angles the robot leg should be set to in order to achieve the desired xy position
- An environment where we can input joint angles, and see the output leg position
- The ability to draw out bezier curves to control the walking pathway for the robot dog
- Interactive GUI to modulate limb lengths, foot path speed, key foot path parameters, etc.

The following sections will cover related works and referenced projects, as well as a report on current progress for the above points, split into detailed progress on the inverse kinematics model and the graphics visualizer separately.

2 INVERSE KINEMATICS MODEL

The position of the leg at each position can be calculated by trigonometry. If the target position of the leg is reachable by the limbs in the kinematics model (i.e. distance to the desired target foot point is less than or equal to the sum of the length of the leg segments) then we can take advantage of the law of cosines to calculate the angles required to make the leg locate directly to the target point. Two solutions should arise from this method, one where the knee bends forwards, and one where the knee bends back. Since we are working with a quadruped model, we can safely assume the knee will bend backwards and come to a singular solution.

This method will be used to calculate the pathway of the foot positions as the robot dog moves. The path of each foot during a step will be set using another spline, and by interpolating over a set amount of time steps we can get discrete positions for the foot during the step. At each time step, we apply our trigonometric model to gather rthe angles required to move the dog to the set position.

If this method becomes shows itself to be insufficient for our methods, we can make use of FABRIK (Forward and Backwards Reaching Inverse Kinematics) to control the limb to a point as close as possible to the desired control point without moving the limb outside of its physically feasible range.

3 VISUALIZATION RESULTS

In order to visualize the inverse kinematics model, we are building a visualizer with PyBullet. In the visualizer, we are initializing three additional target points, which combined with the robot dog will form a Bezier curve path for the dog to follow. We are adding features such that the location of each of the target points can be adjusted using keyboard controls, and the camera can be zoomed in and out. The robot dog will begin moving in its path with the space bar command.

The visualizer also needs to be integrated with the inverse kinematics model since the robot dog currently cannot walk but is only moved through translation. Since we define a Bezier curve path for the dog, we will be calculating the target position of the dog at each time step based on this path.

We also hope to add some additional adjustable parameters such as the ability to control the dog's walking speed and being able to adjust camera angles.

Received 20 February 2007; revised 12 March 2009; accepted 5 June 2009