Functional Specification

Year: 2024 Semester: Spring Team: 05 Project: DodgeBot

Creation Date: January, 2024 Last Modified: January 20, 2024

Member1: Ayman Motoda Email: aelnamr@purdue.edu

Member2: Maximilian Drach Email: mdrach@purdue.edu

Member3: Yusuf Jarada Email: yjarada@purdue.edu

Member4: Carlos Cotera Email: ccoteraj@purdue.edu

Assignment Evaluation: See Rubric on Brightspace Assignment

1.0 Functional Description

PunchBot is designed to make physical training and reaction time improvement fun for children by simulating a dynamic environment where the robot responds to incoming punches. The device consists of a single punchable target mounted on an axle, allowing movement along a fixed base. The primary functionalities of the dodging robot are displayed in the functional block diagrams in appendices 2 & 3 but can be described as follows:

Punch Detection:

Utilizes a camera system to capture the surroundings in real time.

Employs a color detection algorithm to identify brightly colored gloves worn by the user.

Object Tracking:

Implements an object detection and tracking algorithm to monitor the position and movement of the detected gloves continuously.

Dodging Strategy:

Executes a dodging strategy based on the tracked position of the gloves.

Analyze and predict the trajectory of the gloves to enable the robot to move along the axle and avoid incoming punches.

Motion Control:

Utilizes motors and an axle system for physical movement. Two motors and a differential allow the main axle to tilt forward as well as rotate itself in the same way a boxer would in real life. We will implement a control algorithm to ensure swift and accurate dodging movements based on real-time information from the camera system.

2.0 Theory of Operation

Computer Vision System:

The computer vision system of the robot relies on a top-down birds-eye camera, which functions as the eyes of the robot by capturing real-time feed of the surrounding environment. This camera input is then processed by the computer vision system to detect and track the movement of objects, with a specific focus on monitoring the trajectory of incoming punches. The system's ability to analyze the visual data enables the robot to respond effectively to dynamic changes in its surroundings [4]. Look at Appendix 1.

Within the computer vision system, additional techniques have been employed for enhanced image processing. The system leverages the HSV color space and employs k-means clustering to segment the image effectively. This segmentation process enables the system to distinguish and identify pixels associated with the punch, particularly those corresponding to colored gloves. By utilizing the distinct properties of the HSV color space and the clustering capabilities of k-means, the computer vision system enhances its ability to accurately detect and isolate relevant elements in the image, contributing to the overall precision of punch detection [5,6].

3.0 Expected Usage Case

The Dodgebot project is designed to operate within indoor environments, requiring approximately 2 to 3 m2 of clean space for optimal functionality. The system is configured to track punches and the robot uses a single camera, necessitating a controlled environment to ensure precise monitoring. Intended for a stationary setting, once the motors are set up and calibrated with the camera, the system remains in place. While players have the flexibility to move around the robot during interaction, the overall setup is expected to be stationary. Each instance of the project is tailored for a single user at a time in its prototype phase. The target user is ideally over 4 feet tall, with the primary requirement being the ability to throw a punch at the designated target. Minimal restrictions are placed on the user's nature, emphasizing physical capability over specific age or technical literacy considerations.

4.0 Design Constraints

4.1 Computational Constraints

The biggest computational constraint on this process will be the inverse kinematics equations required to move DodgeBot away from a punch. We will need to use object tracking to calculate the distance of the gloves relative to the head of the robot and determine how to move the robot to evade the attack most effectively. Our first attempt at object tracking will be a simple K-Means on marked gloves approach to tracking the position of everything. Next, we will have to establish certain rules on the system so that the robot can automatically evade punches, the distance from a punch must be greater than or equal to 5 inches for example. Finally, we will need to calculate the actual position of all the motors to move the robot to its desired position. This final calculation will be an inverse kinematics problem and will likely be the biggest bottleneck in terms of computational speed.

4.2 Electronics Constraints

The major external components of this project include a few key things. The ELP OV4689 camera is going to be the backbone of the operation of this robot, so it has a few parameters that limit our operation. The frame rate of the camera will decide how many frames we get in a certain time, so having a higher frame rate will increase how many points we can read before giving the decision algorithm its input data. Currently, the camera runs at a native 60FPS, so we may need to increase it if the algorithm does not get a valid input quickly enough. This interfaces through USB, but the Jetson already has this connection covered. The second main part external to the main computation system is the motor controller. Based on what motor controller we picked, we have a few options for communication, but currently, we will try to set up a connection over EtherCAT back to the STM32 which uses an ethernet port physically. This will require some special configuration since EtherCAT is not a natively supported interface in the STM manual. The last interface that we will need to set up is UART. This is to set up communication between the Jetson and the STM32 to send motor commands. This will not have a large challenge of impedance matching since the traces are going to be relatively short on our PCB.

4.3 Thermal/Power Constraints

Luckily, the power consumption of the computation and communication part of the electronics stack is going to be relatively low compared to the rest of the system. The Jetson draws a maximum of 4A at 5V, and the STM32 draws a maximum of 0.258A at 3.3V[1,2]. Since we will be using switching-based supply units of those voltages, the heat that is going to be dissipated is going to be relatively low. The Jetson also has a heatsink on board with the support of using a cooling fan and has a maximum temperature threshold of 97°C. The STM32 has a temperature threshold of 125°C [1,2]. Since the smaller peripherals are connected to power through one of these microcontrollers, their power usage is put together with them. The larger power constraint is going to come from the motors. The AKM2G motors that are being used have a supply voltage of 220V AC to the controller, and a single motor can draw a maximum of 37.6A [3]. We must ensure the circuits we plug the motor controllers into are rated for this current. Cooling is not a huge constraint since the movements we make are going to be so short, that the motors will not have enough time to heat up from these bursts of movement.

4.4 Mechanical Constraints

The device must be light enough to be carried around without extensive equipment. This may be unrealistic due to the size of the motors we are using. We also hope to keep all the electronics contained within the chassis of the device to preserve a clean aesthetic and safe design. Furthermore, regarding safety, the moving head joint must be fully constrained and secure to ensure that it does not move beyond its intended range. The head of DodgeBot is strong enough to be able to take a punch from a user and not break. This means that the device needs to be secured to the ground via weight or screws so it can handle that force, along with the force required to move the head of the robot fast enough to dodge punches. Finally, we want the project to last a long time with little maintenance so all mechanical components must be tested and rated for many uses.

4.5 Economic Constraints

This project has a $350 budget however the main cost comes from the price of the motors and motor controllers. We estimate this to be upwards of $8000. We contacted the school to pursue more funding and hope it will support us in this. Other than the motors, the rest of the components and materials are relatively cheap and do not pose any economic constraint on us.

4.6 Other Constraints

5.0 Sources Cited:

[1] STMicroelectronics, “STM32F765xx STM32F767xx STM32F768Ax STM32F769xx,” Aug. 2023. Accessed: Jan. 20, 2024. [Online]. Available: https://www.st.com/content/ccc/resource/technical/document/datasheet/group3/c5/37/9c/1d/a6/09/4e/1a/DM00273119/files/DM00273119.pdf/jcr:content/translations/en.DM00273119.pdf

[2] NVIDIA, “DATA SHEET NVIDIA Jetson Nano System-on-Module Maxwell GPU + ARM Cortex-A57 + 4GB LPDDR4 + 16GB eMMC,” *NVIDIA Developer*, 2014. https://developer.download.nvidia.com/assets/embedded/secure/jetson/Nano/docs/JetsonNano\_DataSheet\_DS09366001v1.1.pdf?HZQ-9BeogomVeemquiSOvhM7IlASVgbYrcnerDqbamhgFT5S6OkQQGpJ8msR59j7HNav0uJMUWPGre5Ww15J7JNqRyx1A39U4VKjOtE3ajsUDzBhR9sl7suRkyi5jqqD5-gsoPpMFkgLJSGAWYMUiBRHfKH8gg4qnbs5LaO2P1gqPnN8wqiXcqopTTfS6w== (accessed Jan. 20, 2024).

[3] Kollmorgen Corporation, “Automation and Motion Control Programmable Automation Solutions,” *Kollmorgen Products*, 2021. https://www.kollmorgen.com/sites/default/files/AMC-KM\_CA\_000246\_RevG\_EN-mobile.pdf (accessed Jan. 20, 2024).

[4] Wikipedia contributors, “Angle of view (photography),” Wikipedia, Dec. 29, 2023. <https://en.wikipedia.org/wiki/Angle_of_view_(photography)>

[5] Wikipedia contributors, “HSL and HSV,” Wikipedia, Jan. 10, 2024. <https://en.wikipedia.org/wiki/HSL_and_HSV>

[6] Wikipedia contributors, “K-means clustering,” Nov. 20, 2023. https://en.wikipedia.org/wiki/K-means\_clustering

‌

Appendix 1: Camera Calibration Graph and Equation

A whiteboard with writing on it

Description automatically generated

Appendix 2: Functional Block Diagram

Appendix 3: Stretch Goals Function Block Diagram