Final Report

Jeremy Fowers

Sentry Wizard

TAs: Mike Pridgen
Tim Martin
Ryan Stevens
Thomas Vermeer
Devin Hughes

Instructors: Dr. A. Antonio Arroyo
Dr. Eric M. Schwartz

Contents

Executive Summary

Sentry Wizard is an autonomous, FPGA powered motion tracking robot that is designed to patrol an area that is supposed to be empty and engage unauthorized targets by identifying movement with a response time under one third of one second. The robot combines four independent subsystems to accomplish this task—navigation, vision, turret and wireless control—all of which are mounted on a mobile platform. All of these subsystems are interfaced through an Arduino microcontroller.

The robot accomplishes navigation by driving with two wheels, each of which is mounted on a servo that is subsequently mounted on the main platform. The wheels are actuated by an Arduino microcontroller that sets them to drive forwards when the attached sonar detects the way is clear and instructs the robot to spin in place when the way forward is detected to be blocked. This system allows the robot to navigate through areas with varying obstacles without ever coming within 1.5 feet of them.

The vision system uses a Prioria Merlin 2, which includes a Cyclone III C120 and 4 gb of RAM and an Omnivision 5 MP camera to accomplish high reaction speed motion tracking. The camera captures two images within milliseconds of each other and sends them into the FPGA, through a video pipeline and to a NIOS II processor, which then analyzes the two images side by side to determine if any motion has taken place. Subtracting each individual pixel magnitude in the first image by the magnitude in the same position in the second image results in a difference map of the two images. Drawing a bounding box around all points in the difference map which exceed a certain threshold, determined by lighting conditions and set wirelessly during run-time, identifies a moving target. Repeating this process several times and calculating the centroid of each resulting bounding box can establish a motion path. The FPGA finishes its operation by predicting the next point in the motion path and outputting it over RS232 to the Arduinio, which is responsible for aiming and firing the turret. The FPGA can identify a moving target every 100 milliseconds.

When targeting data is provided by the FPGA, an instruction is sent to aim the turret in the predicted path of the target and fire three rounds. The turret is aimed using a 180 degree panning servo and is fired by actuating a relay which can supply power to the turrets firing mechanism. The turret can fire 4 rounds per second continuously by holding the relay closed.

The robot is controlled by its operator over a Bluetooth network. The software on the robot is capable of accepting a variety of commands that can operate the turret manually, engage various behaviors, and toggle the safety for the gun. The robot can also send diagnostic data back to the operator.

Robot functionality is demonstrated by a patrol behavior. Upon receiving a start command over the Bluetooth network, the robot drives through its environment until it encounters and obstacle. After it avoids the obstacle the robot uses the motion tracking system to attempt to identify any moving objects for 10 seconds. If it finds a moving object it tracks it, predicts its motion, aims at it, and fires the turret. After the 10 second period is over it continues to drive and repeats the above steps.

Overall this project was very educational to design and build and met all of its initial objectives.

Abstract

The Sentry Wizard is a motion sensing patrol robot with enhanced vision processing abilities. It will roam and area and rely on a high frame rate CMOS camera and an FPGA to identify moving objects, track them, and predict their future path. From there the robot will aim a target marking turret at the target and fire at it while continuing to track.

The robot will has a short cylindrical body and two wheels for easy turning, a rotating turret with automatic firing for target marking, an Arduino to interface with peripherals, and a vision tracking system handled by a Prioria Merlin 2 with Cyclone III FPGA.

Introduction

The major purpose of designing this robot was to gain experience with embedded FPGA applications, particularly the Merlin 2 platform provided by Prioria Robotics. FPGAs can provide high performance and low power consumption in several critical embedded tasks where an embedded microprocessor simply would not come close to the same performance/power ratio and a desktop computer is not practical.

FPGAs are especially adept at high performance video processing tasks due to their ability to handle large amounts of data in parallel. A common robotics task is object recognition and interaction, so an FPGA-accelerated vision system seems to be the ideal testing ground for the Merlin 2 platform. The robot will have a huge advantage over similar projects because it is capable of far higher computational loads, which eliminate many of the limiting factors of vision-based robots.

One example of a common embedded vision system is the CMUcam. Due to the fact that it uses a small, embedded microprocessor, it is limited to color coded motion tracking that relies on specific lighting conditions and target profiles. The CMUcam is also only able to provide data at a limited frame rate, requiring robots to take multi-second pauses in operation to perform vision analysis.

The FPGA system in place on this robot is capable of high performance video processing, allowing it to detect motion with a refresh rate of up to 10 Hz and track moving targets such as walking humans. The FPGA is interfaced to an Arduino Duemilanove microcontroller via RS232. The Arduino communicates wirelessly with the operator over Bluetooth, reads sonar data, and actuates the robot through full rotation wheel servos, a 180-degree turret servo, and a relay, which activates the turret.

Sentry Wizard demonstrates its capabilities by patrolling an area, avoiding obstacles and scanning for moving objects. If it detects a moving object it tracks it with its vision processor, anticipates its future location, aims there with a turret, and fires at the target to mark it.

Integrated System

The system is comprised of four major subsystems: patrol system, motion tracking system, target marking system, and communications system.

The patrol system includes the Arduino Duemilanove microcontroller and all of the motors and sensors required to navigate the robot around its environment. The components are mounted on a Lowe's 4" PVC flange that provides a central cylinder for storing circuitry and an outer ring for mounting sensors and servos. The robot moves using two speed controlled, full rotation servos attached to wheels. Obstacles in the environment are detected using a Maxbotix LV-EZ1 Ultrasonic Range Finder (sonar).

The motion tracking system is made up of a Prioria Merlin 2 and an Omnivision 5 MP camera. The Merlin 2 system contains a Cyclone III C120 FPGA and 4 gb of RAM along with power regulators and other support circuitry. The FPGA communicates its results back to the Arduino through a RS232 interface which is attached to a RS232 shifter and plugged into a software serial port on the Arduino..

The target marking system uses a modified electric SoftAir device mounted on a 180 degree rotation servo. The SoftAir is modified to include a relay in place of the regular finger-operated trigger system, allowing the Arduino to activate it by holding a pin high.

The communication system uses a BlueSMiRF serial Bluetooth module to communicate with a remote laptop. The laptop can send simple commands to the robot over this network and receive back diagnostic data. The Bluetooth module is connected to the Arduino on a hardware UART.

Mobile Platform

Scope

The design intention with the mobile platform was to keep it as simple as possible because it serves mainly to carry the motion tracking system. The mobile platform had to be capable of moving in a straight line, rotating in place, and supporting the Arduino, Merlin 2, turret system, and battery.

Platform Structure

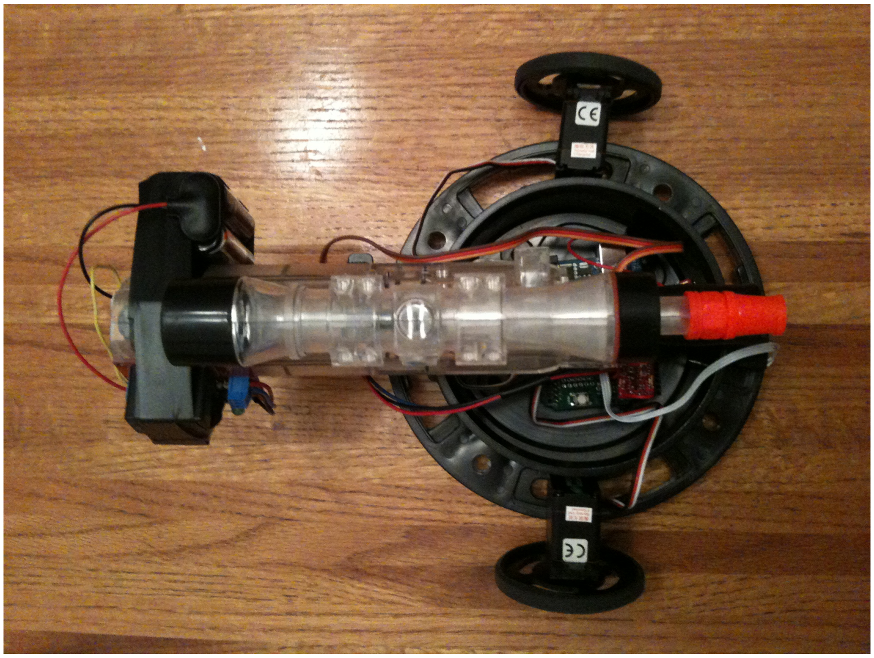

Figure 1. Top view of the platform design.

The platform is built around a 4" Flange. The two wheel servos, the turret servo, and the sonar sensor are all mounted on the outer ring. The inner cylinder contains the Arduino, Merlin 2, camera, and an 11.1 V, 1100 mAH LiPo battery. The turret, which consists on a 4 AA battery pack and a relay, is mounted on the turret servo. A caster ball on the under side of the flange below the turret servo provides balance and support.

Lessons Learned

The use of a premade part for the platform rather than a purpose built had both advantages and drawbacks. On the one hand, it was easy to come by and didn't require a lengthy design and iteration process, but on the other hand it turned out to be cramped for space in the end and provided limitations in part placement. The lesson learned here is to completely map out the mechanical design, including board placement, before beginning any work on programming or assembly.

Actuation

Scope

The robot's actuation has two goals: first, to allow the robot to autonomously patrol and area and secondly to aim and fire a mounted turret. The patrol actuation was meant to be as simple as possible to allow more time for other goals, while the turret was meant to snap onto targets and fire quickly.

Structure

Two wheels, each of which is mounted on a SpringRC SM-S4303R Continuous Rotation Servo, provide the robot's patrol actuation. The motors allow the robot to move slowly throughout a flat environment such as carpet or tile. This system, while very simple, accomplishes the original design goals of the robot.

The turret uses a Large Servo from Sparkfun Electronics to pan over the 180 degrees in front of the robot. This servo is kept in the 90 degree (straight forward position) and can rotate to and can rotate to any other position in under one second, aiming the turret in the requested direction. The SoftAir device uses a small motor to actuate a firing mechanism. The terminals of this motor are connected to a battery pack through a relay. Actuating the relay causes the SoftAir to fire 4 times per second.

Lessons Learned

The actuators in this project all draw a lot of current, which the linear regulator on the Arduino is not capable of handling all at once. Steps had to be taken in software to never start or stop two items at the same time, slow down the turret servo, and only fire when stationary.

Sensors

Obstacle Avoidance

Theory of Operation

Sonar devices bounce ultrasonic frequency sound off of nearby objects and then measure the time it takes for the sound to return to a sensor. Since the robot is designed to stay away from all obstacles directly in its driving path, a single, wide angle sonar device is perfect for picking up possible obstacles around the robot.

Hardware

The Obstacle Avoidance system consists of a Maxbotix LV-EZ1 Ultrasonic Range Finder (sonar) wired to an Arduino Duemilanove. The sonar is connected to power, ground, and an analog to digital pin on the Arduino. The Arduino continuously reads from the sonar's ADC pin and initiates the avoidance algorithm whenever it detects an object within the sonar's range (represented by a value under 400 on the ADC).

Algorithm

Upon detecting a potential obstacle, the Arduino sets the wheels to spin in opposite directions and continues to read from the sonar. When the sonar no longer detects an obstacle (represented by a value over 400 on the ADC) the robot continues with its previous behavior.

Experimental Results

The robot performs admirably using a single sonar sensor for basic obstacle avoidance in real world environments. It only stumbles when approaching an obstacle at very wide angles (over 90 degrees to the sonar device), a problem which can only be mitigated by adding additional sensors on the edges of the device. Given that the demonstration for the robot will involve a minimum of movement in a controlled environment, this was deemed outside the scope of the project and never addressed.

Target Marking

Theory of Operation

In order to demonstrate its motion tracking abilities the robot will mark its target by impacting it with properly aimed small plastic balls. This behavior will verify that the target tracking algorithm is capable of not only identifying the target, but accurately predicting its motion.

Hardware

The robot will mark moving targets with a modified SoftAir brand device. The target marker is equipped with a digital relay controlled motor capable of the sustained launching of 4 small plastic balls per second, which will reduce the need for perfect accuracy by creating a stream of plastic balls in the air. The device has its own battery pack made up of 4 AAs. The device is mounted on a 180 degree servo which allows panning left and right for aiming. The unit specifically lacks the ability to tilt in order to avoid possible accidents—in its current configuration the robot can only launch the plastic balls in a straight line a few inches above the ground.

Software

The relay is actuated by the Arduino unit with a single digital input. Bringing the input to VCC causes the relay to close and provide power to the motor in the device. The servo is actuated by the Arduino using a PWM signal. A safety for the device is provided in software.

Experimental Results

The device is able to use its servo mount to quickly aim, which will be essential to quick target marking. Plastic balls can be launched at will by powering the motor and come out at a rate of 4 per second for as long as the motor is active.

Bluetooth Wireless

Theory of Operation

Bluetooth modules provide easy to set up wireless networking between embedded devices, personal computers, and pocket devices such as phones and media players. The range of operation for most modules is about 30 feet, which is sufficient for the conditions Sentry Wizard will be demonstrated in.

Hardware

The robot platform uses a Blue SMiRF Gold Bluetooth Modem connected to the Arduino on its serial communication bus. The modem is a serial communication module that creates a wireless UART between the Arduino and a remote computer.

Software

Currently, the embedded wireless software consists of a simple set of instructions, which allow for basic robot control. The instructions are transmitted using ZTerm, a simple terminal emulator on the Mac. Additionally, the robot is able to send status messages back to the terminal.

The current list of instructions is:

- "a": toggle target marker safety
- "s": stop automated driving behavior
- "d": start automated driving behavior
- "f": activate target marker
- "w": turn target marker left 10 degrees
- "e": center the target marker
- "r": turn target marker right 10 degrees
- "D": toggle detection behavior
- "T" + [0-9]: set the motion tracking threshold to 10 times the integer
- "F" + [0-9]: set the number of frames used for motion tracking to the integer
- "B": toggle patrol behavior

Experimental Results

The Bluetooth network connects within 60 seconds of starting up the laptop and robot, which is well within the requirements for the system. The network maintains its integrity up to 30 feet, which is also sufficient.

Motion Tracking System

Theory of Operation

In an environment with only one moving target, a stationary camera can take two photos in quick succession whose only difference will be the shape of the moving target. Subtracting the two images against a sufficient threshold will reveal the center of the moving object, which can be tracked across several frames. A series of moving centroids reveals the motion path of the target.

The steps of the algorithm are as follows:

- While the robot is still, take two photos of the environment in rapid succession.
- Convert the images to grayscale and store them in memory on the Merlin 2 at an appropriate resolution.
- Using a predefined tolerance, find the difference between the two images by subtracting the pixel values and deleting pixels (making them black) if the value falls below the tolerance threshold.
- Find a bounding box around the remaining pixels, which only contain the moving object.
- Find the center of the bounding box.
- Repeat the above steps to ensure that an accurate series of bounding boxes is being created around a real moving target.
- Trace the midpoints of each bounding box to form an approximate path which the target is following.
- Relay the path information back to the target painter

This algorithm is best executed on an FPGA due to the hardware's ability to quickly take images into memory and analyze them in parallel with very little latency.

Hardware

The motion tracking "special sensor" is composed of a Prioria Merlin 2 platform with an Altera Cyclone III C120 FPGA and an OmniVision OV5633 5 megapixel CMOS camera. The camera is connected to the Merlin 2 with a specialized parallel cable and the Merlin 2 is connected to the Arduino with over an RS232 connection.

Software

The FPGA contains a NIOS II soft processor, a video capture pipeline, an RS232 controller, and a video analysis pipeline. The details of this software are protected by a nondisclosure agreement between Dr. Stitt's UF laboratory, Prioria Robotics, and myself.

Prototyping Results

The algorithm described above was rapidly prototyped in Python code and executed on a MacBook Pro equipped with an iSight camera. The results, shown in the first oral presentation, prove that the algorithm works even with shifting shadows present between frames. The Python script was able to track a moving object across 5 frames with a degree of precision sufficient for the target tracking platform.

Experimental Results

The motion tracking system is able to acquire a moving target in 100 ms by processing 20 video frames per second (10 pairs, any of which can show motion). If only one frame pair is taken the device can be used to identify an object which suddenly appears or follow a slow moving object with a 10 Hz refresh rate. If the number of frame pairs is set above 3 per sample the system takes the last recorded position and adds the average distance traveled to predict motion. This mode is better suited to fast moving targets such as a remote controlled car. The 3 frame pair setting results in a 3 Hz refresh rate, which is the lowest rate at which the system is still usable.

Lessons Learned

The Bluetooth module was an invaluable tool for acquiring large amounts of real time diagnostic data and I would recommend its inclusion in all future IMDL projects. A hard lesson learned was that all FPGA systems can be incredibly tricky to deal with, and it is possible to spend a lot of time without making much progress. I would advise caution to anyone attempting to use an complex FPGA system in a real-world semester long project due to the difficult debugging process.

Behavior

The robot is equipped with three autonomous behaviors and manual turret controls. The first two modes, obstacle avoidance and motion detection, show off the two major capabilities of the system separately. The third mode has the robot patrol an area by autonomously driving and scanning for targets. All controls and behaviors are set wirelessly using a Bluetooth network with the operators computer.

Manual Controls

When the robot is stationary it is possible to zero the turret, adjust its aim left and right by increments of 10 degrees, toggle the turret's software safety, and actuate the relay to fire the turret. The turret only fires if the safety is disabled, an action which must take place before every firing event.

Obstacle Avoidance/Driving

The robot checks the analog value on the sonar as part of its main loop. If any object is detected on the sonar the robot spins in place until it is no longer facing a barrier. If the sonar does not detect anything, the robot drives forward until the sonar interrupts it.

Motion Detection

The robot sits still and scans using the camera, attempting to detect motion. If a target is identified, the Merlin provides a firing solution and the turret aims at it automatically, following it for as long as it can. If the turret safety is toggled off the turret will fire 3 times at the target in quick succession while continuing to track it. The safety is toggled back on automatically after this three round burst.

Patrol

The patrol mode is main demonstration mode for the integrated system. The robot drives forward until it encounters an obstacle, then rotates to avoid it until it is clear to drive forward again. If it can drive forward for one second, which implies that it has avoided that particular obstacle, it initiates motion tracking mode for 10 seconds. If a target is identified, the Merlin provides a firing solution and the turret aims at it automatically, following it for as long as it can. If the turret safety is toggled off the turret will fire 3 times at the target in quick succession while continuing to track it. The safety is toggled back on automatically after this three round burst. Regardless of whether a target is engaged the robot will continue to drive after the 10 second scanning period. This behavior repeats until it is wirelessly disabled.

Lessons Learned

An integrated demonstration mode is far more impressive than the separate functions of the robot performed on their own. The key to wowing the audience is to show them everything the robot can do in one coordinated effort that shows them how it could be applied in a real world situation.

Conclusion

The robot accomplished its original goal of carrying the Merlin 2 system and performing video analysis tasks. Many challenges were encountered along the way, including power management software to compensate for inadequate regulation hardware, last minute platform design changes, and lengthy learning, programming, and debugging process with the FPGA.

Future work on this robot could include a completely redesigned power management system with a regulator that can handle the changing actuation of multiple servos and relays without negatively impacting microprocessor operation. The motion tracking algorithm can be improved to support frame rates up to 90 per second to allow for more accurate motion prediction and higher resolution. The microprocessor would be updated to a more advanced model which with two hardware UARTs in order to avoid software serial related instability. Lastly, the platform would be updated with custom hardware to be better built for mounting the specific hardware in this project.

Building the robot was a very rewarding experience and after several revisions of the software, electrical hardware, and platform it delivered its intended demonstration on Media Day and met its original goals.