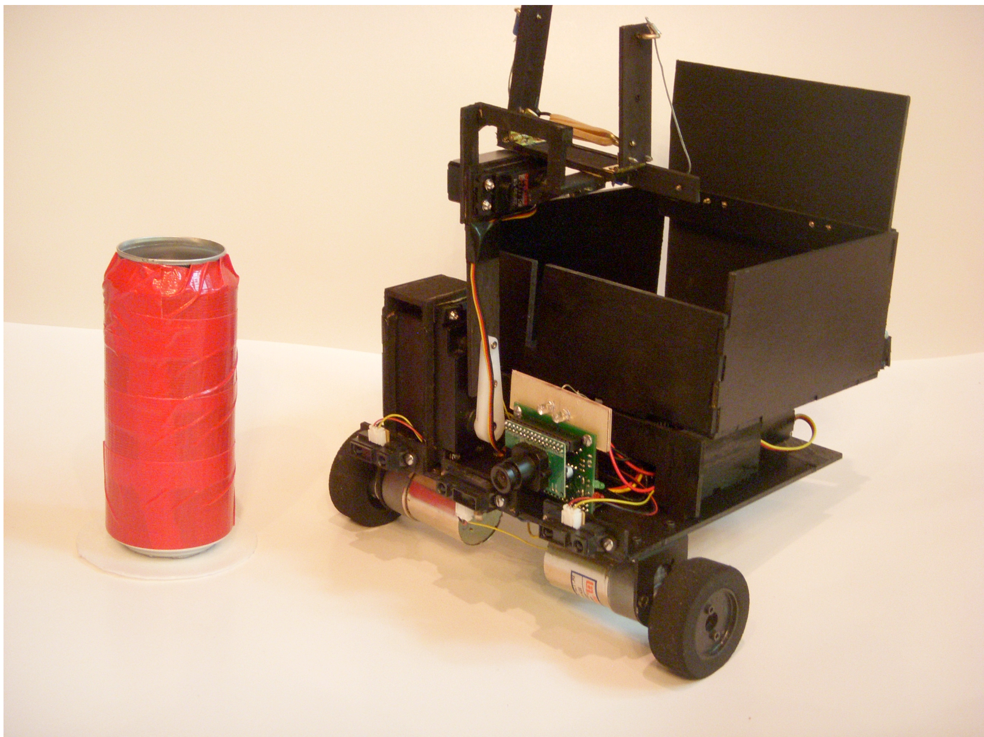
Robo-Dumpster:
The Autonomous Garbage Truck



by
Alan Hamlet

University of Florida
EEL5666
Intelligent Machine Design Lab

Instructors:
Dr. Arroyo
Dr. Schwartz

TA's:
Mike Pridgen
Thomas Vermeer

Table of Contents

Abstract

The goal of this project is to design and build an autonomous mobile robot that performs a relatively simple task. The robot must demonstrate obstacle avoidance behaviors as well as incorporate a special sensor. This paper details the design of Robo-Dumpster, The Autonomous Garbage Truck, an autonomous mobile robot that searches for full garbage cans, empties them into its' on board receptacle, and then dumps the contents of the on board receptacle into a designated garbage area when it is full. How the robot works, the behaviors of the robot as well as the sensors, actuators and mobile platform used are explained.

Introduction

Robo-Dumpster is a proof-of-concept robot that behaves much like a normal garbage truck but is completely autonomous. This could be used in places such as grocery stores, amusement parks or even in neighborhoods to save money that would otherwise have to go toward paying employees to do this tedious and unpleasant job. This robot could work 24 hours a day 7 days a week for only what it costs to maintain it and the batteries used to power it. Any jobs lost would be made up for in the industries building and maintaining these robots.

The Robo-Dumpster seeks out garbage cans that are full using a CMU camera. The robot empties the cans into its' own on board container until it that container is full. Then, using the CMU camera, it searches for the designated garbage collection area where it then dumps the garbage and continues searching for full garbage cans.

Integrated System

The robot is controlled by a Pridgen Vermeer Robotics Xmega128 microcontroller board. The board speed is 32 MHz and has a RISC based 8 bit core as well as 12 PWM signals. The robot's sensors make use of the board's analog to digital converter ports and the CMU camera uses one of the RS-232 serial ports. Five PWM signals are used; two are sent to DC motors and three are sent to servo motors. The DC motors use a Sabertooth regenerative dual channel motor controller from Lynxmotion in order to control their speed and direction. The microcontroller receives data from the sensors and based on that data instructs the actuators to take the appropriate action. The robot also has an LCD screen to provide visual feedback on what it is doing. All of the robot's systems are powered by six 1.2 volt Nickel-metal-hydride rechargeable batteries.

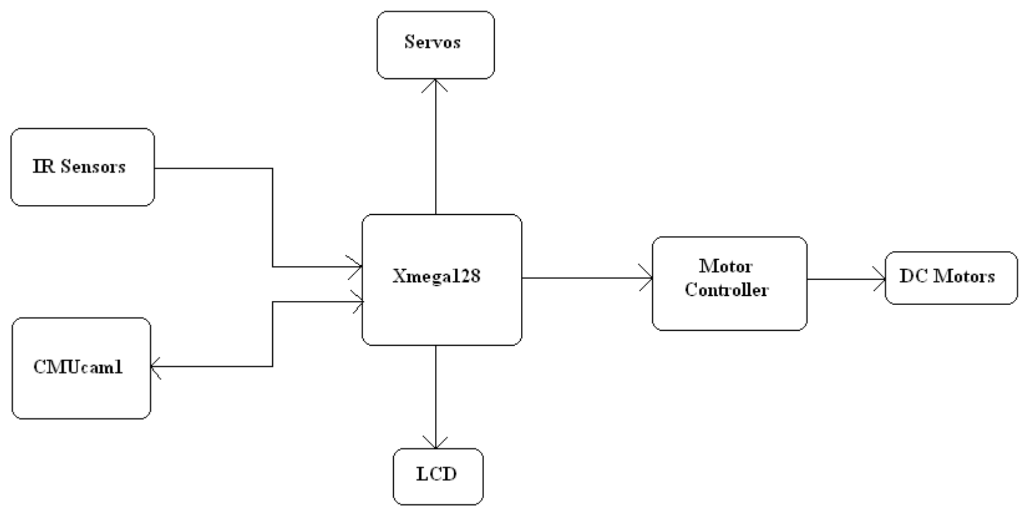


Figure 1 - Integrated system block diagram.

Mobile Platform

The mobile platform needs to allow the robot to move with an acceptable level of speed and agility over the appropriate terrain. The mobile platform of the Robo-Dumpster consists of two front wheels, each driven by a 12 volt DC gear head motor, and a rear caster. The DC motors have 10 kg-cm of stall torque and operate at speeds up to 253 rpm. Since the robot is meant to operate on smooth terrain and will not be moving a large amount of weight, the size of the motors is more than enough to move the robot at an acceptable speed. The wheels are 2.13" diameter rubber to increase traction and prevent slippage. The wheel/motor assemlies are mounted to a 7"x"7 piece of plywood 1/8" thick.

The two DC motors in front allow the robot to operate with differential steering. By having the two wheels drive in opposite directions the robot can perform a nearly zero radius turn. The motors are controlled by a Sabertooth regenerative dual channel motor controller.

Actuation

In addition to the DC motors for propulsion, the Autonomous Garbage Truck will also have three servos for actuators. The arm that grasps the garbage cans and dumps the contents of the can into the robot's bin will consist of two servos. One servo is for gripping the garbage can and one is for raising the arm. The third servo will be used to dump the contents of the robot's bin into the garbage collection area. For raising the arm I am using a HiTec HS-815BB servo due to the high torque demand for the task. The servo has a stall torque of 343 oz.-in. and 140° rotation. For both the gripper on the robot's arm and the dumping mechanism I am using a HiTec-485HB servo with 83 oz.-in of torque and 90° rotation.

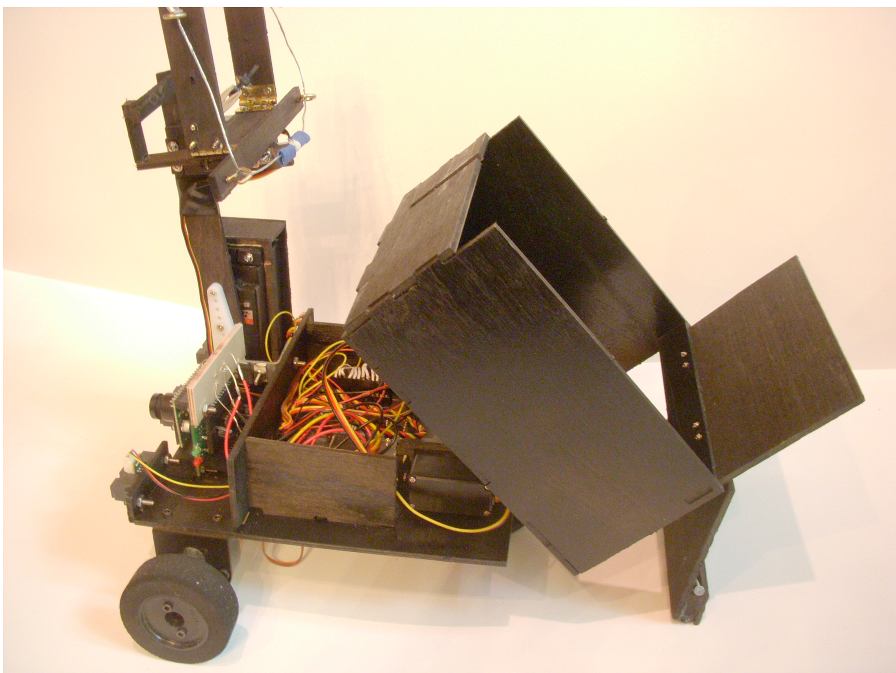


Figure 2 – Dumping mechanism actuated by a HiTec-485HB servo.

Sensors

The Autonomous Garbage Truck uses 2 IR sensors for obstacle avoidance; one on the front left side and one on the front right side. This enables the robot to ‘see’ when it is approaching an object and turn away from it. The robot is also equipped with a short range IR sensor placed on the front center of the robot to detect when the garbage cans are close enough to be picked up.

My special sensor is a CMUcam 1 used to track the garbage cans as well as the green trash collection area. The garbage cans are a bright red for easy tracking. The CMUcam is programmed to search for the centroid of a blob of red while the robot is looking for full garbage cans. The camera detects the garbage cans and the robot will drive toward the can and empty its contents. When the robot needs to empty its containers contents, the CMUcam will be tracking green. Once the camera sees green, the robot will drive up to the garbage collection area, turn around and dump its’ container.

Behaviors

When the Autonomous Garbage Truck is turned on it will go through a quick calibration routine and then begin scanning for garbage cans to be emptied. If no full cans are seen in its' current position the robot will move to a new position while obstacle avoiding and scan the area again. Once a full garbage can is located the robot will drive up to and grasp the can with its arm, raise the arm emptying the contents of the can into the robot's container, and then set the garbage can back down. The Autonomous garbage truck will then scan again repeating this process 2 times until its container is full. Then the robot will scan for the green garbage collection area. When the CMUcam sees the collection area, the robot will approach it until the IR sensors tell the robot it is close enough, turn around, and back up. Then the robot will dump the trash out and continue the search for full trash cans.

The Robo-Dumpster most basic behavior is obstacle avoidance. This behavior is programmed to be done at all times. The more sophisticated behaviors such as finding the garbage can and picking it up are overlaid on the basic behavior of obstacle avoidance. This means that even if Robo-Dumpster sees a full trash can and is driving toward it to pick it up, the robot will still avoid obstacles that get in its way.

Experimental Layout and Results

In order to prevent any outside interference from color sources in the audience I constructed an arena for the Robo-Dumpster to be demonstrated. The two garbage cans were placed at one end of the arena and the green garbage collection area in the corner on the other end. The garbage cans were filled with ping-pong balls to represent garbage.

The Robo-Dumpster successfully and consistently emptied the two garbage into its bin and then dumped the entire load into the garbage collection area. The demonstration successfully demonstrated the effectiveness of an autonomous garbage truck.

Conclusion

The Robo-Dumpster is a proof-of-concept robot that could save many companies and the government a lot of money. A robot that completely autonomously finds and empties full trash cans would be cheaper and more reliable than a human. This would not negatively affect the economy though since jobs in the new industries of building and maintaining these robots would emerge, so everybody wins.

I have learned a lot from designing and building the Robo-Dumpster. The integration of many systems working together to perform even a simple task such as this is quite complex. It takes detailed mechanical and electrical engineering designs together to get the many parts to work in harmony. I have learned many intricacies about robots that I never even thought about before.

After a semester of designing and building, Robo-Dumpster successfully locates trash cans and dumps their contents into its' container. When the dumpster is full, the robot locates the collection area and dumps the load there all while practicing obstacle avoidance.

Appendix

Robo-Dumpster code
Functions:

```
#include <avr/io.h>
#include "usart.h"
#include "global.h"

int ArmPosition;
int CansEmptied;

int MidIR(){
	int sum=0;
	int i=0;
	while(i<4){
		sum=sum+ADCA5();
		i++;
	}
	sum=sum/4;
return(sum);
}

int LeftIR(){
return(ADCA0());
}

int RightIR(){
return(ADCA3());
}

void Straight(){
	ServoD1(-50);
	ServoD0(50);
}

void LTurn(){
	lcdData(0x01);
	lcdString("Avoiding Right");
	lcdGoto(1,0);
	lcdString("LeftIR: ");
	lcdInt(LeftIR());
	int i=49;
	while(i>-50){
		ServoD0(i);
		delay_ms(1);
```

```
		i--;
	}
}

void RTurn(){
	lcdData(0x01);
	lcdString("Avoiding Left");
	lcdGoto(1,0);
	lcdString("RightIR: ");
	lcdInt(RightIR());
	int i=-49;
	while(i<50){
		ServoD1(i);
		delay_ms(1);
		i++;
	}
}

void Backup(){
	lcdData(0x01);
	lcdString("Backing Up");
	int i=50;
	int j=-50;
	while(i>-50){
		ServoD0(i);
		ServoD1(j);
		delay_ms(5);
		i--;
		j++;
	}
	delay_ms(200);
	ServoD0(50);
	delay_ms(500);
}

void Backup1(){
	lcdData(0x01);
	lcdString("Backing Up");
	int i=50;
	int j=-50;
	while(i>-50){
		ServoD0(i);
		ServoD1(j);
		delay_ms(5);
		i--;
		j++;
```

```
        }
        delay_ms(200);
        ServoD0(50);
        delay_ms(1000);
}

void Backup2(){
        lcdData(0x01);
        lcdString("Backing Up");
        int i=0;
        int j=0;
        while(i>-50){
                ServoD0(i);
                ServoD1(j);
                delay_ms(5);
                i--;
                j++;
        }
        delay_ms(500);
        ServoD0(50);
        delay_ms(1000);
}

void ArmUp(){
        if(ArmPosition!=95){
                ArmPosition=-44;
                while(ArmPosition<95){
                        ServoC0(ArmPosition);
                        delay_ms(20);
                        ArmPosition++;
                }
                delay_ms(1000);
        }
}

void ArmDown(){
        if(ArmPosition!=-45){
                ArmPosition=90;
                while(ArmPosition>-45){
                        ServoC0(ArmPosition);
                        delay_ms(15);
                        ArmPosition--;
                }
        }
}
```

```
void Open(){
	ServoC1(100);
}

void Close(){
	ServoC1(-20);
}

void Dump(){
	ServoD0(-40);
	ServoD1(-40);
	delay_ms(4600);
	ServoD1(40);
	delay_ms(1000);
	ServoD1(0);
	ServoD0(0);
	int i=90;
	ServoC0(60);
	while(i>-20){
		ServoC3(i);
		delay_ms(10);
		i--;
	}
	delay_ms(1000);
	while(i<90){
		ServoC3(i);
		delay_ms(10);
		i++;
	}
	CansEmptied=0;
	int p=0;
	while(p<100){
		ServoD1(-50);
		ServoD0(50);
		Avoid();
		p++;
	}
}


void PanRight(){
	ServoD1(0);
	/*int speed=(MX-58);
	if(speed<23){
		ServoD0(15);
		ServoD1(20);
```

```
	}
	else{
		ServoD0(speed);
		ServoD1(speed);
	}*/
}

void PanLeft(){
	ServoD0(0);
	/*int speed=(58-MX);
	if(speed<23){
		ServoD0(-15);
		ServoD1(-23);
	}
	else{
		ServoD0(-speed);
		ServoD1(-speed);
	}*/
}

void Stop(){
	ServoD1(0);
	ServoD0(0);
}

void Straight1(){
	ServoD1(-24);
	ServoD0(20);
}

void Straight2(){
	ServoD1(-40);
	ServoD0(40);
}

void Search(int cnt){
	lcdData(0x01);
	lcdString("Searching");
	if(cnt<70 && MX==0){
		ServoD0(30);
		ServoD1(30);
	}
	if(cnt>=70 && MX==0 && Avoid()==0){
		ServoD1(-50);
		ServoD0(50);
		Avoid();
```

```
	}
}


int Avoid(){
	if(LeftIR()>=2900 && RightIR()<2900){
		RTurn();
		return(1);
	}
	else if(LeftIR()<2900 && RightIR()>=2900){
		LTurn();
		return(1);
	}
	else if(LeftIR()>=2900 && RightIR()>=2900){
		Backup();
		return(1);
	}
	else
		return(0);
}

void EmptyCan(){
	Stop();
	ArmDown();
	Straight1();
	delay_ms(2500);
	Stop();
	Close();
	delay_ms(300);
	ArmUp();
	delay_ms(300);
	//Shake();
	delay_ms(300);
	ArmDown();
	Open();
	delay_ms(300);
	ArmUp();
	delay_ms(300);
	Backup2();
	CansEmptied++;
}
```

Robo-Dumpster main code:

```
int main(){
	xmegaInit();				//setup XMega
	delayInit();					//setup delay functions
	ServoCInit();						//setup PORTC
Servos
	ServoDInit();						//setup PORTD
Servos
	ADCAInit();							//setup
PORTA analong readings
	lcdInit();							//setup LCD
on PORTK			//display "PV Robotics" on top line (Line 0) of LCD
	USARTInit();						//move LCD cursor to
the second line (Line 1) of LCD
	lcdGoto(0, 0);
	lcdString("Hello! I'm ");
	lcdGoto(1,0);
	lcdString("Robo-Dumpster.");
	CansEmptied=0;
	int cnt=0;
	int cnt1=0;
	ServoC3(90);
	delay_ms(1500);
	while(1){
		if(cnt==100)
			cnt=0;
		if(cnt1==100)
			cnt1=0;
		if(CansEmptied<2)
		{
			TrackColor("TC 100 245 0 50 0 50\r");
			if(MX==0 && MY==0){
				Search(cnt);
				cnt++;
			}
			else{
				lcdData(0x01);
				lcdString("Locating cans");
				lcdGoto(1, 0);
				lcdString("X: ");
				lcdInt(MX);
				Stop();
				Open();
				if(0<=MX<=45){
					Straight1();
```

```
                        PanLeft();
                }
                if(MX>=55 && MX<=80){
                        Straight1();
                        PanRight();
                }
                if(MX>45 && MX<55){
                        if(LeftIR()<2900 && RightIR<2900 &&
MidIR()<1900)
                                Straight2();
                        else
                                Straight1();
                        if(MidIR()>=1900)
                                EmptyCan();
                }
        }
}
else if(CansEmptied=2)
{
        TrackColor("TC 0 50 100 150 0 50\r");
        if(MX==0 && MY==0){
                Search(cnt1);
                cnt1++;
        }
        else{
                lcdData(0x01);
                lcdString("Dumping");
                lcdGoto(1, 0);
                lcdString("X: ");
                lcdInt(MX);
                if(0<=MX<=40){
                        Straight1();
                        PanLeft();
                }
                if(MX>=60 && MX<=80){
                        Straight1();
                        PanRight();
                }
                if(MX>40 && MX<60){
                        if(LeftIR()>2100 && RightIR()>2100)
                                Dump();
                        else
                                Straight2();
                }
        }
}/*
```

```
                Open();
                //ArmDown();
                TrackColor("TC 150 245 0 50 0 50\r");
                lcdData(0x01);
                lcdString("MX: ");
                lcdInt(MX);
                lcdGoto(1,0);
                lcdString("MidIR: ");
                lcdInt(MidIR());
                delay_ms(400);*/
        }
}
```