Using a Lidar for Robot Navigation in a Room

Mike Anderson

Chief Scientist

The PTR Group, Inc.

http://ThePTRGroup.com

mike@theptrgroup.com

Who is The PTR Group?

- The PTR Group was founded in 2000
- We are involved in multiple areas of work:
  - Robotics (NASA space arm)
  - Flight software (over 35 satellites on orbit)
  - Offensive and defensive cyber operations
    - I'll leave this to your imagination ☺
  - Embedded software ports to RTOS/Linux/bare metal
  - IoT systems architecture and deployment

Speaker/Author Details

- **Website:**
  - **http://www.theptrgroup.com**
- **Email:**
  - **mailto:mike@theptrgroup.com**
- **Linked-in:**
  - **https://www.linkedin.com/in/mikeandersonptr**
- **Twitter:**
  - **@hungjar**

Almost 40 years in the embedded and real-time industry for both commercial and Government customers.

What We'll Talk About…

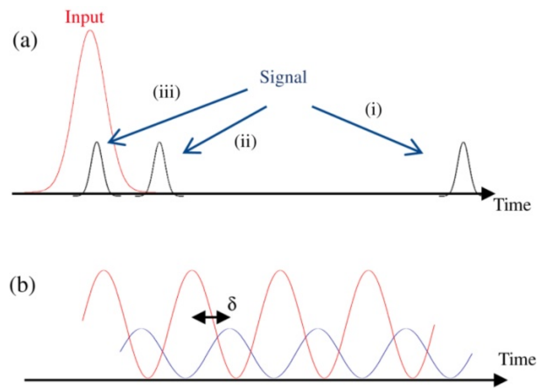
- What is a Lidar?
- What options exist?
- Why put it on a robot?
- How did it work?
- Summary

What is a Lidar?

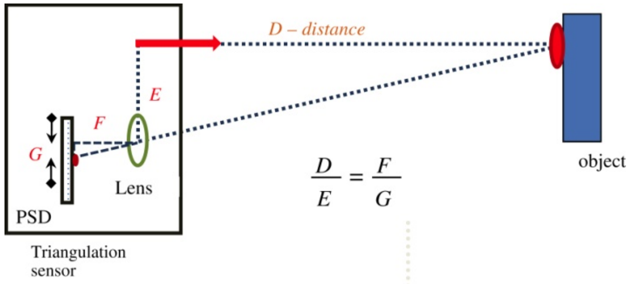
- A lidar (a.k.a, LIDAR, LiDAR, LADAR) is an abbreviation for Light Detection and Ranging
  - Essentially, a light-based version of radar
    - We'll use the lower case so it doesn't look like we're shouting ☺
- Lidars most frequently use time of flight measurement for a laser pulse to be reflected off of a target to determine its range
- The frequency of the laser light is a concern
  - 600-1000nm is most common for non-scientific applications
- Unfortunately, these frequencies can be focused and absorbed by the human eye
  - Therefore, the power needs to be limited to Class 1
- High-performance lidars tend to use 1550nm frequencies
  - Not easily absorbed by the eye so you can use higher powers

How does Lidar Work?

- Most lidars use a time-of-flight calculation between when the laser pulse is sent and when it reflects off an object
- Then, triangulation is used based on the detector's distance from the laser source to the laser detector to determine the distance
    - That's why many lidars have two "eyes"



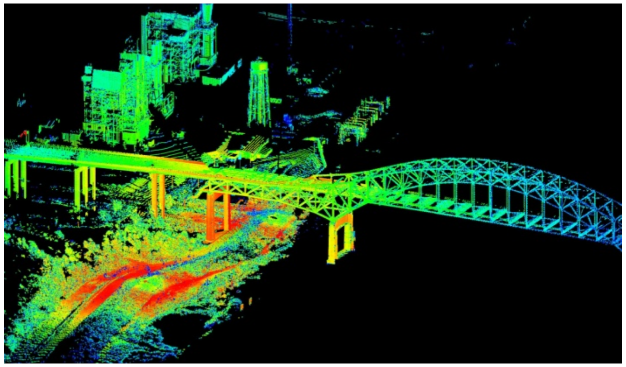
Source: hackaday.com



Source: hackaday.com

Where are Lidars Being Used?

- Lidars were originally designed for surveying applications
  - Used to make high-resolution, 3D maps from airborne platforms like satellites, aircraft and drones
- Also used in self-driving cars and for 3D modeling



Source: Geo-Plus.com

Source: Geo-Plus.com

Single Point Lidars

- Depending on the application, lidars can be either a single point-and-shoot distance measurement or be built to run in scanning mode
- The point-and-shoot variety are often found in laser-based measuring tapes
- Single point lidars can now be had for ~ $150
  - Like the Garmin lidar from various vendors
- It's possible to mount a single-point lidar on a spinning platform to make it a scanning lidar
  - I tried this, but wire-management and balancing the load on the platform proved to be difficult

Source: walmart.com

Source: sparkfun.com

Scanning Lidars

- Scanning lidars typically spin and measure distance in a full 360 degree circle
  - Scanning lidars are considerably more expensive than point-and-shoot types
- Scanning lidars have a spinning frequency as well as a pulse duration
  - This determines the number of distance measurements you can produce per second
- Your interface and controller must be able to keep up with the data rate
  - I2C, PWM, SPI and serial are all common interfaces
- Inexpensive scanning lidars start in the $400 range and go up in price quickly

Source: robotshop.com

Source: sparkfun.com

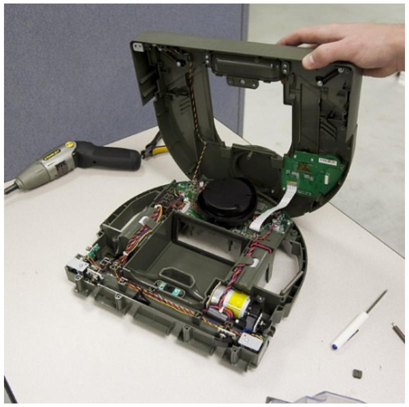
Commercial Drone Lidars

- There are a number of manufacturers of commercial drones equipped with scanning lidar systems
- However, the weight of the lidar and the amount of power that it uses limit the drone’s time aloft
- Still, they are neat if you can afford the $120K price tag ☺

Source: altigator.com

Cheap Scanning Lidars

- The Neato cleaning robot is known to have a scanning lidar built in
- There are teardowns on sites like Sparkfun for how to extract it
  - And, some instructions for how to interface to it
- Just the lidar can be purchased from eBay for about $120
  - But, a new cleaning robot unit will be $400

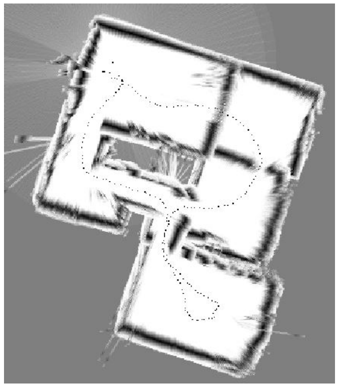


Robots and SLAM

- Simultaneous localization and mapping (SLAM) is an approach for creating a map of an unknown environment while keeping track of an agent's location within the environment
  - Defined by this simple equation ☺ :

$$P(x_t|o_{1:t},m_t) = \sum_{m_{t-1}} P(o_t|x_t,m_t) \sum_{x_{t-1}} P(x_t|x_{t-1}) P(x_{t-1}|m_t,o_{1:t-1})/Z$$

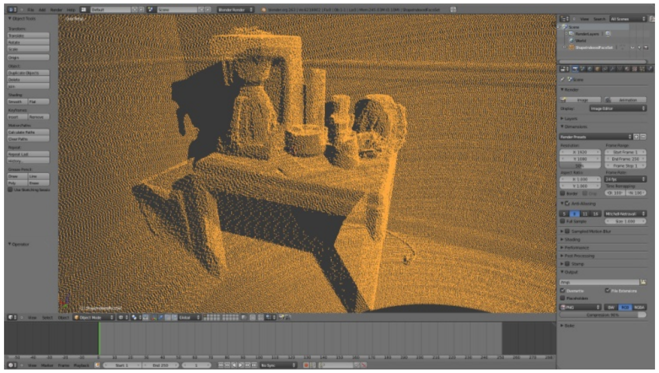
- SLAM can be performed using different measurement modalities
  - Ultrasonic, infrared or various forms of lidar
- FWIW, Google has just released their SLAM algorithms for their self driving cars
  - But, they're really not optimized for hobby use
- And, for a proper SLAM implementation, you will also need an inertial measurement unit (IMU) sensor package
  - X/Y/Z accelerometer, X/Y/Z gyroscope, X/Y/Z magnetometer at a minimum



Source: linked-in.com

Creating a Point Cloud

- The field of view can be thought of as a collection of individual measurements yielding points in the 2D or 3D field of view
  - This is referred to as a *point cloud*
- These can be imported into an application like Blender for rendering or graphed using Processing, python, etc.
  - Alternatively, you can plot them in Matlab, etc.



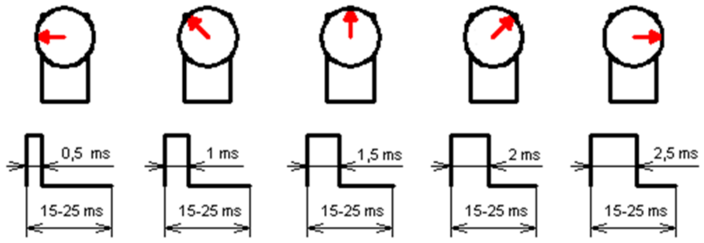
Source: blenderartists.org

Creating a Robot

- Creating a robot involves a lot of time and attention to detail
- Topics to address include:
  - What's supposed to do?
  - Mobility (wheels, tracks, hovercraft, flying drone, etc.)
  - Motor controllers (power draw, sustained current vs. instantaneous current and stall current of the motors)
  - Motor types (brushed or brushless)
  - Powering the motors (battery chemistry such as NiMH, LiPo, LiFe, etc.)
  - What controller will you use (uC, Linux, dedicated controller, etc.)
  - Powering the controller (maintaining a consistent power source)
  - Sensor packages (ultrasonic, lidar, camera, altitude, GPS, etc.)
  - Wireless link type (distance and interference are the major issues)
  - Software for the motor controllers and software for the main controller
  - Even more…

Linux and Motor Controllers

- Many of the inexpensive motor controllers use RC-hobby servo type PWM signals
  - Usually running at about 50 Hz with pulse widths in the 0.5-2.5 ms range

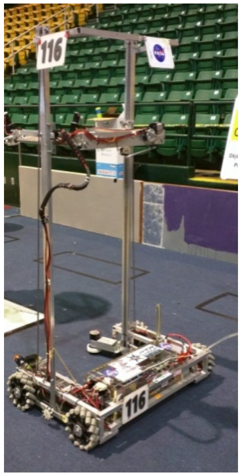


Source: stackexchange.com

- The variation of the pulse width makes the device move forward or backwards or controls the speed
- While servos themselves are fairly tolerant, many motor controllers are less so
  - A little jitter in the signal can cause a motor to self destruct
- So, while the real-time patch for Linux does reasonably well for most things, the jitter can be an issue
  - The use of uCs like the Sitara PRU or Arduinos can be used to address the jitter problem

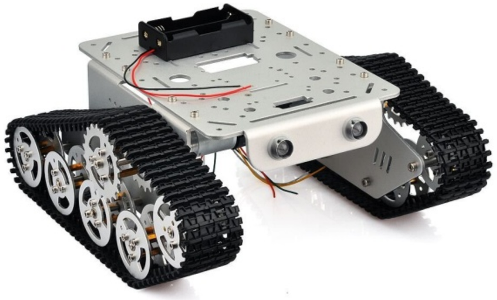
What do I want my Robot to do?

- In my case, I wanted a way to experiment with the new lidar models that are coming onto the market
  - I'm a FIRST Robotics mentor, so range finding is often part of the challenge every year
- I also wanted to start playing with SLAM approaches to help make the robots more autonomous
  - My student drivers are getting overwhelmed with the amount of data that's available
- I'm also considering adding voice enablement to create a "companion" bot for seniors
  - Something they can naturally interact with but can also remind them to eat meals and take their medications on time
    - But, it also must navigate through the house autonomously for health and status checking
- However, simply mapping a playing field or simply the room so you can do obstacle avoidance is a good place to start



Choosing a Robot Platform

- I’ve built many large (100lbs+) robots and a few smaller platforms with wheels
    - They’re too hard to transport and TSA goes nuts when they X-Ray my luggage ☺
        - The batteries are also a problem
- So, I wanted something different
    - Small and tracked sounded neat
- Dual 9V motors w/ 4.5A stall current
    - Unfortunately, not much mounting space, as I discovered later
- And, the stall current puts the motors beyond the typical L298 H-bridge motor controllers
    - They typically max out at 2A



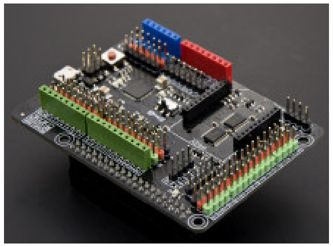
Source: amazon.com

Plans for the Controls

- I've used both the Raspberry Pi and the Beaglebone for robots in the past
    - I wanted both Bluetooth and Wi-Fi
        - But, I was concerned for the compute load that the lidar would put on the controller
            - So, I went with the Raspberry Pi 3
- I also wanted to avoid the PWM jitter issue, so I used an Arduino and a high-power motor controller
    - Max current 30A, sustained 14A and max voltage 16v
        - Some soldering required
- The Arduino was available as a Raspberry Pi "hat" that plugged into the Pi and provided a functional "Leonardo" equivalent board

Source: amazon.com

Source: sparkfun.com



Source: dfrobot.com

Plans for the Controls (2)

- Powered from a 5A 9.6v RC car battery
  - This can't be used to power the Pi directly
    - I used an RC aircraft universal battery eliminator circuit (UBEC) for 5V
- And, I found this nifty USB Micro B breakout board for getting power to the Pi
  - Available from Sparkfun
- Assembling and wiring the robot took several days

Source: amazon.com

Source: amazon.com

Source: sparkfun.com

Choosing the Lidar

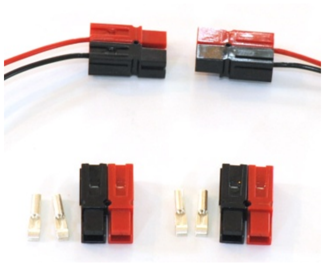
- I have several of the Garmin lidar units
  - Great for attaching to a servo for obstacle avoidance
    - You can scan back and forth looking for objects
- But, I wanted to try a 360˚ scanning lidar
  - I've seen the Uber self-driving cars in Pittsburgh and I thought a scanning lidar would be cool
- So, I opted for the RPLIDAR A2 unit from Slamtec
  - 4K samples/s @ 10Hz, 600 RPM

Source: robotshop.com

Source: slamtec.com

Wiring it up

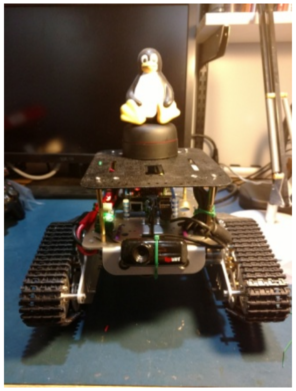
- The motor controller is built as an Arduino shield
  - External power for the motors via the shield
    - The motors get wired directly to the shield
- I can talk to the Arduino and the RPLIDAR via RS-232 UART connections
- I need 5V@2A for the Raspberry PI 3
  - 9.6V will work for the motor controller
- Use the UBEC for 5V to power the Pi and the Arduino hat and power the motor controllers directly
- I'll just disconnect my battery using Anderson power poles as a power switch

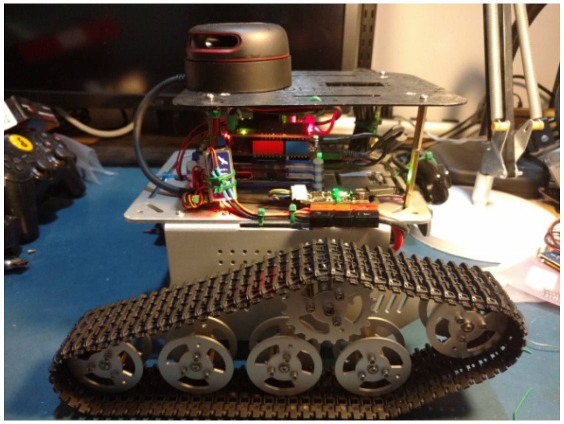


Source: cumulus-soaring,com

The Finished Robot

- After several days… I finally have a mechanically finished robot!
- The lidar is mounted on the back of the bot to provide room to mount other sensors like a 9DOF IMU that are below the lidar's scan height
- I also mounted a USB camera on the front of the bot so I could see where I'm going
  - Tux is optional… ☺





Software Anyone?

- A software example for the Arduino and the motor controller is provided by Sparkfun
  - Some tweaks needed to be able to control it from the Pi via the serial port
- I needed to write software to control the RPLIDAR and take its samples and then forward them to something that could use the data for plotting
  - SLAMTEC offers an SDK for C/C++ in Linux
- However, the path of least resistance for SLAM code appeared to be the Robot Operating System (ROS)
  - http://www.ros.org/
  - They have instructions for installation on Raspberry Pi 3 using Ubuntu MATE (16.04.2 LTS) and Raspian Jessie
- The most complete instructions for ROS (and a prebuilt set of packages) seemed to be for Ubuntu MATE

Installing ROS

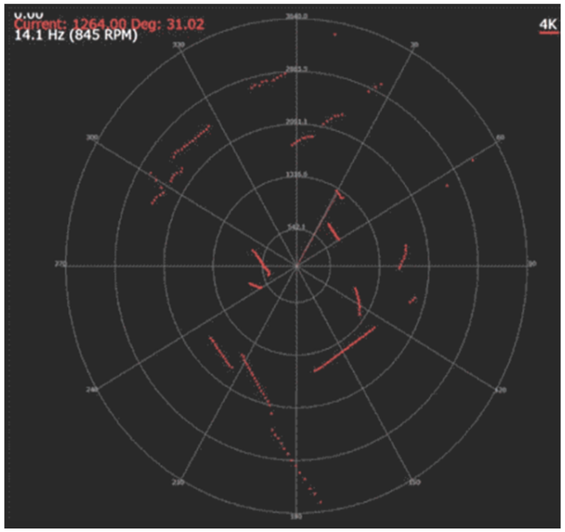
- Just search for “install ROS raspberry pi” and you’ll find several sets of instructions depending on what distro you’re running on the Pi
  - I found references to the RPLIDAR on http://hackaday.io to try and get it working for an older version of ROS
- ROS is a pretty complete system for robot control
  - But, it’s big and takes a while to figure out what’s needed
- ROS supports a pub/sub approach for data collection and monitoring
- Fortunately, there’s a wiki for the RPLIDAR on the ROS site
  - http://wiki.ros.org/rplidar

Software Pieces

- Ubuntu MATE 16.04 with modifications for:
  - Running the Pi in AP mode so I can connect to it
    - Set fixed IP address for the AP and “pi-robot” SSID
  - Installed tightvncserver and openssh for remote access
    - Set up Avahi to announce the “pi-robot” via mDNS so we can find it when we try to connect
  - Support for the camera
    - I’m just using cheese at the moment, but hope to hook in ROS
- The motor controller is controlled via the Arduino serial port
  - So, I wrote software for driving the robot using an Xbox controller in Linux and it relays to the robot using TCP
- ROS is currently being used for the RPLIDAR only
  - I’m still in the learning process for everything that ROS can do
- All of this will be in a github repository (user ID: taichichuan) as soon as I get everything debugged

Scan from the Lidar

- Using ROS code, I have the lidar scanning
- The output is pretty crude because I don't have all of the SLAM software running yet
  - I also need to integrate ROS-compatible IMU for the SLAM functions



Current: 1264.00 Deg: 31.02
14.1 Hz (845 RPM)

Current Status

- The wireless communication software and motor control are working
- The lidar is also scanning and producing lots of distance measurement information
- No integrated IMU yet
- Unfortunately, after I got everything built, I discovered that one of the gearboxes was totally borked ☹
  - I'm waiting on a replacement which is why I don't have the robot here
- If I can't get a replacement from the Kookye folks, I've already ordered an alternative platform
- As soon as I have everything put back together and working, I'll post it up on github
  - If you're interested, send me an email and I'll let you know when it's posted

Source: amazon.com

Summary

- This first attempt at using a lidar has been an interesting exercise
  - Lots of money and time invested and only a few crude pictures to show for it so far ☹
- Linux can definitely handle the process with aplomb
  - I’d like to transition to the Beaglebone and use the PRUs for motor control instead of an Arduino
    - Maybe switch to the Linux drone code base
- The mechanicals can be daunting
  - Power, wiring, interconnects, etc. plus the motors and motor control are a challenge
    - Building a mid-sized robot was new to me
- I’m looking forward to getting the unit back online and continuing with the software process
  - ROS and Linux drone code both look attractive moving forward