LiDar Math for Big B

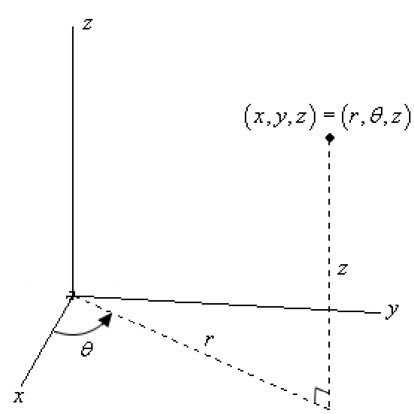
Omkar Guha

March 2025

1 Introduction

This is a short explanation of the math to find the vectors necessary to store the point-cloud data in cylindrical coordinates. Once in cylindrical coordinates, it is a straightforward calculation to convert to other coordinate systems.

I am opting to do the math in cylindrical coordinates because of the symmetries of the problem. The LiDar rotates around an axis; the angle of rotation would be the angular component of the coordinates, θ. The rest of the math can be done on a 2D plane, with standard r and z coordinates.



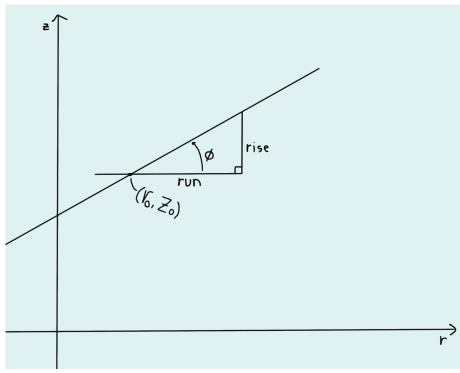
From Paul's Online Notes, Cylindrical Coordinates

2 Lines

We will work with lines on the (r,z) plane. Our goal is to find the equation of a line that forms a particular angle with the r -axis and goes through a particular point. We can use the point-slope formula.

$$z - z_0 = m(r - r_0)$$

Here m is the slope of the line and (r_0,z_0) specifies a point the line passes through. To find the slope given an angle, we can use trigonometry. Let us call the angle a line forms with respect to the r-axis ϕ. Slope is simply rise over run, if we imagine a right angle formed with the angle ϕ then the $tan(\phi)$ will give us opposite over adjacent, rise over run, and thus our slope m.



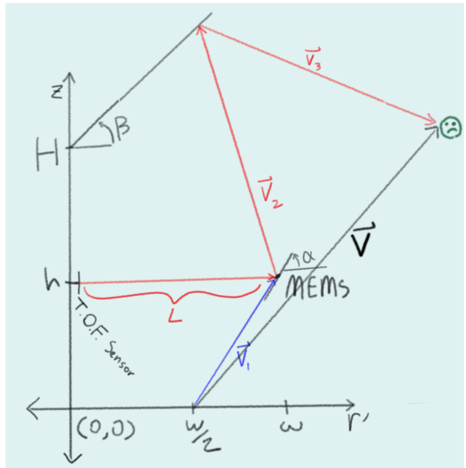
Thus, the equation of a line passing through a point (r_0,z_0) which forms an angle ϕ with the r-axis is

$$z = tan(\phi)(r - r_0) + z_0$$

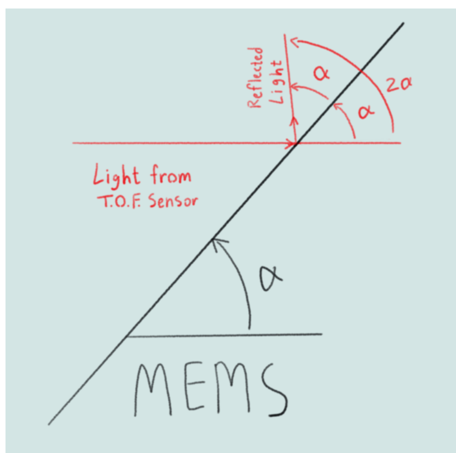
3 Actual Math

In the Desmos demos that Aram and I made, the origin is different from the axis of rotation. Thus I use r'-axis instead of the r-axis. r' and r are related by the translation $r = r' - \frac{w}{2}$. This won't affect the vector mathematics all that much. The origin in (r,z) coordinates is ($\frac{w}{2}$,0) in (r',z) coordinates. This means that the vector $\mathbf{v}_1$ represents the vector from the bottom center of the LiDar pointing to the MEMS mirror. We are interested in storing the vector $\mathbf{v} = \mathbf{v}_1 + \mathbf{v}_2 + \mathbf{v}_3$ as part of the point-cloud data.

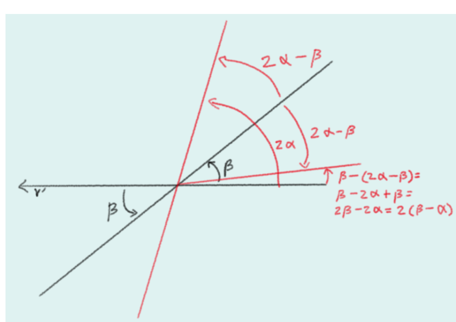
In the image below the T.O.F. sensor emits a ray of light which bounces off the two mirrors and hits the sad face. We want to store the location of where the sad face is hit as a point, in a point-cloud of data.



The light emitted from the T.O.F. Sensor is reflected off of the MEMS mirror at an angle of 2α with respect to the r'-axis.



The light coming off the MEMS mirror then hits the upper mirror. It comes off of the upper mirror at an angle of angle of $2(\beta - \alpha)$.



The T.O.F. sensor returns the length $L_{TOF} = L + |\mathbf{v}_2| + |\mathbf{v}_3|$. We know L, we need to find $|\mathbf{v}_2|$ to find $|\mathbf{v}|_3$. To find $|\mathbf{v}_2|$, we need to find the intersection point between the ray of light that comes off the MEMS mirror and the upper mirror.

We can do this via finding the equation of the line that describes the upper mirror and the ray of light coming off the MEMS mirror. We know that the upper mirror goes through the point $(0,H)$ and is at an angle of β. We know that the ray of light coming off the MEMS mirror goes through the point (w,h) and is at an angle of 2α.

The equation of the line describing the upper mirror is

$$z = tan(\beta)r' + H$$

The equation of the line describing the ray of light coming off the MEMS mirror is

$$z = tan(2\alpha)(r' - w) + h$$

Solving the system we get the point of intersection (r'_p,z_p) is

$$r'_p = \frac{h - H - tan(2\alpha)w}{tan(\beta) - tan(2\alpha)}$$

$$z_p = h - tan(2\alpha)w + tan(2\alpha)\frac{h - H - tan(2\alpha)w}{tan(b) - tan(2\alpha)}$$

$|\mathbf{v}_2|$ is the distance between (w,h) and (r'_p,z_p) given by the distance formula

$$|\mathbf{v}_2| = \sqrt{(w - r'_p)^2 + (h - z_p)^2}$$

Using the relation for L_{TOF} described above

$$|\mathbf{v}_3| = L_{TOF} - (L + \mathbf{v}_2)$$

$\mathbf{v}_1$ is a vector with startpoint $(\frac{w}{2},0)$ and endpoint (w,h).

$$\mathbf{v}_1 = [\frac{w}{2}, h]^T$$

We can find the components of $\mathbf{v}_2$ and $\mathbf{v}_3$ as we know their magnitude and direction.

$$\mathbf{v}_2 = |\mathbf{v}_2|[cos(2\alpha), sin(2\alpha)]^T$$

$$\mathbf{v}_3 = |\mathbf{v}_2|[cos(2(\beta - \alpha)), sin(2(\beta - \alpha))]^T$$

Thus we know how to compute all the necessary vectors to store our point-cloud data.

The final vector in which we are interested is $\mathbf{v} = \mathbf{v}_1 + \mathbf{v}_2 + \mathbf{v}_3$. We know the components of $\mathbf{v}$ are $[r, z]^T$. When storing point-cloud data, we would store it as (r,θ,z).