Analysis

Trajectory Analysis

There are various factors that weigh in when designing a missile turret. This section will cover factors such as the distance to the target, whether it is within range of missilelauncher, since it would be very inefficient to have a missileturret, that shoots after targets it wont be able to hit. This section will also cover how we are going to calculate the missile trajectory, this includes what angle the missilelauncher should be in, to be able to hit the target. The section will introduce numerous formulas, some of which might serve the same purpose, but have some small changes, this is because we don't know which formula would be the best to use in our project thus it will be wise of us to cover multiple formulas. We have chosen to neglect factors such as wind resistance, friction and other aerodynamics. The variables in this section are as follows:

- v is speed denoted in m/s
- θ is the angle of the turret adjacent to the ground
- g is the g-force, which is "1", since we will assume that we are at sealevel
- f is the number of meters the missile launcher is off the floor
- t is time until impact
- d is the distance the projectile can or will travel.

Jeg valgte med vilje at bruge floor istedet for ground, for at give læseren et indtryk af at f står for floor. Da g allerede var optaget.

Distance

When calculating how far the projectile will be able to travel on a flat surface this formula is used:

$d = \frac{(v^2 * \sin(2*\theta))}{g}$

But what if we were too put our missile-launcher on a tower? Then we would need another formula, since the tower would then be able to shoot in a downward angle, causing a negative angle. To do this we would have to use this formula:

$d = ((v * \cos(\theta))/g) * (v * \sin(\theta) + \sqrt{\frac{(v*\sin(\theta)^2 + 2*g*f)}{g}}$

Time

Since our goal is to hit a moving target, we then need to know how long it will take for the missile to travel the distance from the missile launcher to the

target. Since the target is moving, the target might not be in the same place from which it was seen, but it has moved within the time of the missile was fired. So the traveltime of the projectile is needed in order to predict where the target will be. To calculate when the projectile will hit the ground once it has been launched, we would have to use this formula:

$traveltime = \frac{(v*\sin(\theta)+\sqrt{v*\sin(\theta)^2+2*g*f)}}{g}$

Though this formula is not complete since it shows when the projectile will hit the floor from when it is launched, but we need to know when it will the target from a certain distance. We still don't know whether it will be used or whether it will be modified to meet our requirements.

Using Coordinates

If we know the height of the target that we need to shoot, which should be possible when using the Xbox 360 Kinect, we can then implement this formula: $y = y0 + x * \tan(\theta) - \frac{g*x^2}{2*(v*\cos(\theta))^2}$

This formula will figure out the height of the projectile, when it is at distance x in angle θ. Then by using a loop, we can then calculate at what angle the turret has to be in in order to hit with the missile launcher. But this form of calculation might take too long. Also there is no way to know how long it will take for the NXT brick to calculate the angle, which means we do not have a precise time constant to use, when taking into account the time of the projectile's trajectory plus the time it takes to calculate the correct angle. But nonetheless the calculations should be right. This is why it is important for us to try and have a computer calculate everything, then send the data to NXT brick, telling it what position it should move into before launching the missile. If it hits, that means our calculations are correct and we can try and have the NXT brick do the calculations, since this course is about embedded systems, if it misses that means that either the computer is too slow to do the calculations or the calculations are wrong. Another thing about the formula is that it is only working in 2 dimensions, meaning that the turret has to be face to face with the target and then there has to be a separate formula that calculates the speed of the moving target and then moves the turret in the correct position according to that speed, the travel time of the projectile and the time it takes to do those calculations.

Getting the coordinates

Using coordinates we get from the Xbox 360 Kinect, we can get 6 coordinates, the first 3 would be $C0 = (x0, y0, z0)$. And the next three coordinates we would receive within a fixed time schedule, for the sake of the example at the end, let's just say that the timelapse between the coordinates will be 1 second. So the next three coordinates, $C1 = (x1, y1, z1)$. And by using these two sets of coordinates we can create a third set, $C2 = (x2, y2, z2)$. Now lets say $(C0 = 15, 6, 0)$ and $(C1 = 30, 6, 5)$. We can then assume that it one second later be in $C2 = (x2 = x1 + (x1 - x0), y2 = y1 + (y1 - y0), z3 = z1 + (z1 - z0))$, which would be $C2 = (45, 6, 5)$. This should be a fast way of calculating where the target is going to be. And would take out the guesswork from the formula mentioned above.

Denne section skal nok flyttes op, og så skal de 2 sektioner skrives om

Within range

Another way to determine whether a projectile is within range is to use the angle formula:

$\theta = \frac{1}{2} * \arcsin(\frac{g*d}{v^2})$

We are not actually going to use everything from this formula, we are actually only going to use the fraction within the arcsine. Because we know that if it can find an angle for the distance, which would be the distance from the Kinect to the target, then it is within range, but if we cannot, then it is not within range. Now the domain of the arcsine is $-1to1$ so if we just compute

$\frac{g*d}{v^2}$

and check whether it is within the domain of $-1to1$. If it is, then it is within range, if is is not then it is not within range. However this is not entirely accurate, since the Kinect will measure the distance from the turret to the target.

Working formula

So none of the formulas described above is one that meets our requirements. It need to able to take two or three coordinates. and by using those coordinates it should be able to calculate the angle of turret.

Has to be done later, since we don't have a set of working formulas

Conclusion to section

To summarize the formulas. We are going to get the coordinates using the method described above in subsection "Getting the coordinates". We are then going to use those coordinates to figure out the angle the turret has to be in by using the formula described in the subsection "Using Coordinates". The time formula is going to be used to calculate the time it takes for the projectile to reach the target, so that the NXT brick can take that into account when predicting where the moving target is going to be. The distance formula will be used to determine whether the target is within range of the missile launcher.

We need to figure out the speed at which the missile launcher can fire. Jeg kigger derudover stadig efter en formel som vil virke i tre dimensioner