TU/e 4GA10 Design of a modern trebuchet

Report Design of a modern trebuchet

4GA10

Group Number:	20
Group members:	Dennis Lambertus Theodorus Boots
	Joris Sebastiaan van den Brekel
	Dylan Marinus Duijmelink
	Sebastiaan Joächim van der Grift
	Tim Heiszwolf
	Bas Hubertus Cornelis Sprengers
	Quinten Swaan
	Thijs Verfuurden
Tutor:	Jesper Hegeman
Project coordinators:	M. van Gorp
	H.C. de Lange
	G.P.M. van Hattum
	M.H.M. Willems

Contents

1 Overview of symbols

Symbol	Subscript	Quantity	Unit
F	F_d (drag), F_b (buckeling), F_c (collision)	Force	Newton (N) or kilogram meter per second squared (kg*m*s^{-2})
E	E_g (potential gravitational), E_r (rotational)	Energy	Joule (J) or kilogram meter squared per second squared (kg*m^2*s^{-2})
L	L_p (projectile arm), L_c (counter-weight arm)	Length	Meter (m)
D		Distance	Meter (m)
y	y (instantaneous), y_o (starting)	Height	Meter (m)
ρ	ρ_a (arms), ρ_g (air/gas)	Density	Kilogram per meter (kg*m^{-1}) or kilogram per cubic meter (kg*m^{-3})
m	m_p (projectile), m_c (counter-weight)	Mass	Kilogram (kg)
I		Inertia	Kilogram meter squared (kg*m^2)
Θ	Θ_s (starting), Θ_e (ending)	Angle to the horizontal	Degrees (°) or radians (rad)
V	V_0 (launch), V (absolute), V_x (x-direction), V_y (y-direction)	Velocity	Meter per second (m*s^{-1})
p	p_x (x-direction), p_y (y-direction)	Position	Meter (m)

2 Introduction

A trebuchet is a medieval war-machine which uses a falling counterweight to provide kinetic energy to a projectile. A trebuchet conforming to a list of constraints will have to be designed and constructed. The goal is to have this trebuchet throw the projectile as great a distance as possible..
After the first design step, in which the requirements, preferences and constraints, so called RPC's, for the design are set, the next step is to determine the functions of the design and after that to come up with a solution for each function. These solutions together form the concept for the preliminary design.

After the preliminary design is finished the design can be further elaborated upon. This is the phase of the design in which the preferences are taken into as much consideration as possible in order to generate a final draft. For the final detailing, models are made in NX motion and Matlab. This way the model is optimized as much as possible to ensure the trebuchet throws the projectile as great a distance as possible possible.

The trebuchet and the projectile, will have to be put in Siemens NX10 as a design. This design is sent to a company by the name of Shapeways to create the model using 3D-printing. After the model has been printed, it can be analysed for any mistakes which occurred during printing. Consequently, the design will be tested and optimized for the final demonstration.

The final step of the design process is to evaluate the production process and the choices that were made during the process. Also, the design must be verified to conform to the RPC's. This is the report of the production process of group 20.

3 RPCs, functions and solutions
At the start of the design process, some requirements, preferences and constraints (RPC's) will have to be set. After that functions for the trebuchet must be set up using those RPC's. The solutions of those functions, the functions and the RPC'S for the trebuchet are in this chapter.

3.1 RPCs
Two main requirements have been set for the trebuchet.:
- The trebuchet needs to be able to fire at least 4 meters or further.
- The frame is to be stable. The trebuchet should not fall over, nor should it collapse.

Also, a few preferences were set:
- A simple design is preferable over a complicated design. This preference takes the value of mathematical approximations into consideration.
- The trebuchet should, in its essence, use its materials as efficiently as possible.

The following constraints were given in for the design of the trebuchet to be in accordance with:
- The construction must be built on a steel base plate (100x100 millimeters).
- The construction must be mounted to the steel base plate with up to 4 M5 bolts
- The projectile's starting position must be above the base plate
- The projectile needs to be designed and printed by the group.
- An actuation mechanism must be provided so the structure can be put in motion with one hand.
- The prototype must fit into a rectangular beam measuring 300x100x50 millimeters of which ten percent (150 milliliters) can be solid material.
- Adding other materials is not permitted.

3.2 Functions and solutions
One of the main steps in the designing process is the creation of the list of functions and their solutions. In the case of this trebuchet, the list of functions and solutions was not that long, but interesting nonetheless.

To distinguish the functions , the trebuchet was divided into four segments. The first segment, consisting of the counterweight, is responsible for the generation of the potential energy required to launch the projectile . The second segment of the trebuchet was defined as the arm. This segment has multiple functions, ranging from holding the projectile, to converting the potential energy into kinetic energy. The arm is also responsible for throwing the projectile at a specifically defined angle. The third segment is the projectile itself. Its sole function is to cover a distance as great as possible using the kinetic energy transferred to it by the arm. The fourth and final segment is the frame of the trebuchet. Its function is to keep all the moving parts in place and hold them in the right position. It is essentially the backbone of the trebuchet.

Every function has a solution. Therefore, four solutions have been defined to go with the functions.. The first solution is for the counterweight. The counterweight is dropped from a relatively great height to generate potential energy. The second solution is for the arm, which holds a basket at one end containing the projectile. To convert the potential energy of the counterweight into kinetic energy, the counterweight is attached to the arm and dropped. This motion rotates the arm, transferring some of the potential energy into kinetic energy for the projectile. The arm will be stopped at a certain stage in its circular motion to give the projectile the optimal angle of release. Thirdly, the projectile has the solution to be shaped like a ball. This shape is able to move through the air relatively easily and is a safe choice compared to other options. Finally, the frame, which will be a construction in the shape of a tower, supported by relatively small joists. This way not a lot of material is used in it, yet it doesn't lose strength because of that. This should make the frame strong enough to not have it fall apart when the trebuchet is fired.

4 General design

After the RPC's were set up there was searched for a general design. This chapter goes into detail on how the overall concept of a floating arm trebuchet works. The floating arm trebuchet is the general design on which the final design was based.

4.1 Basic design

Figure 4.1.1: The basic design of a floating arm trebuchet.

As the functions state, the function of the counterweight is to generate kinetic energy by means of the conversion of potential energy. To give this conversion maximum efficiency, all horizontal movement of the counterweight will have to be eliminated. The floating arm trebuchet is a general design in which this horizontal movement is constrained, making it an ideal baseline.

The basic design of the floating arm trebuchet is seen in figure 4.1.1. In this design the arm can roll horizontally across rails.The tower through which the counterweight falls restricts the counterweight to a strictly vertical motion. This way, the maximum efficiency rate of the conversion of potential energy into kinetic energy is reached.

In the starting position, the counterweight is held at the top of a tall drop shaft and is attached at the end of the arm. The projectile rests at the other end of the arm. There is a pivot point in the middle of the arm, where wheels are attached to it. The wheels smoothen out the horizontal slide of the arm on the rails.

4.2 Trigger mechanism

A triggering mechanism will have to be created to keep the trebuchet in firing position until a designated firing moment. The most accessible way to accomplish this is to put a pin through the frame on which the counterweight rests. When the trebuchet needs to be put into motion, the only action required is to pull the pin out of the frame, causing trebuchet to fire.

4.3 Operation

The counterweight is held up by a pin. The trebuchet is at rest, it can not move until the pin is removed. When that happens, the counterweight ceases to have support, causing it to fall. This falling motion causes the end of the arm with the projectile on it to move upwards. When the counterweight reaches its lowest point, the projectile will be released from the arm. The sling arm will now be sliding from side to side over the rails until it stops as a result of of the friction it encounters.

5 Physics of a floating arm trebuchet
To go from the general design to the specific design some calculations needed to be done. In this chapter the physics of the general design is described and applied.

5.1 Assumptions and relevant factors
Before useful physics can be done on a few assumptions need to be made. If those assumptions are not made than the physics becomes more complex while they don't increase the accuracy of the physics significantly.

First no friction is assumed. Whilst ignoring kinetic contact friction is not very impactful ignoring air friction is. This is because the projectile has a small mass and a relatively large surface which means that air friction is going to significantly slow down the projectile which will make it fall short of the expected distance. Secondly, the projectile and counterweight are assumed to be point masses, and the arms are assumed infinitely thin. Also a uniform density of the arms was assumed to make calculating the center of mass of inertia easter. And lastly it was assumed that once the arms are stopped (and thus no normal force is present on the projectile) the projectile will continue its path according to Newton's laws.

What are the relevant factors? First of all the length of the arm is relevant. The arm is split into two parts. The arm from the pivot point to the projectile (the projectile arm with length L_p in meter) and the arm from the pivot point to the counter-weight (the counter-weight arm with length L_c in meter). It is also important that we know the density of the arm (ρ_a in kilogram per meter) and of course also the mass of the projectile (m_p in kilogram) and the mass of the counter-weight (m_c in kilogram) are important. There needed to be a way to define the starting and ending position. For this the angle of the arm to the horizontal was used (the starting angle Θ_s in radians and the ending angle Θ_e in radians) and the starting height of the trebuchet (y_0 in meter). For an overview of the variables and symbols see chapter '1 Overview symbols'.

5.2 Calculating the launch velocity of the projectile

There are several ways of calculating the launch velocity of the projectile but for the trebuchet, energy calculations were used. Since friction is nullified, there are only conservative energies which makes it easier in a way that only the starting and ending position need to be considered.

First the released potential energy needs to be calculated. This can be done by using the following equation,

$$E_g = m * g * h \quad \text{(equation 5.2.1)}$$

where E_g is the potential gravitational energy in Joule, m the mass in kilogram, g the gravitational acceleration in meters per second squared and h the height to the reference frame in meter [1]. This needs to be done for each part (the counter-weight, the counter-weight arm, the projectile arm and the projectile) for the starting and ending position. Then the difference in the potential gravitational energy between the starting and ending position is taken. This resulted in the following equation for the released potential energy

$$E_g = g * (sin(\Theta_e) - sin(\Theta_s)) * (m_c * L_c + \tfrac{1}{2}\rho_a * L_c{}^2 - \tfrac{1}{2}\rho_a * L_p{}^2 - m_p * L_p) \quad \text{(equation 5.2.2).}$$

Before the angular speed can be calculated from the moment of inertia is needed. It is a measure of how hard it is to induce rotation to an object. The moment of inertia is

$$I = \sum_{i=o}^{N} r_i{}^2 * m_i \quad \text{(equation 5.2.3)}$$

where I is the moment of inertia in kilogram meter squared, m_i is the mass of a point mass on the object and r_i is the distance of that point mass to the axis of rotation [1]. When this equation is applied to the trebuchet the inertia becomes

$$I = m_c * L_c{}^2 + \tfrac{1}{3}\rho_a * L_c{}^3 + \tfrac{1}{3}\rho_a * L_p{}^3 + m_p * L_p{}^2 \quad \text{(equation 5.2.4).}$$

Once the inertia and potential gravitational converted into rotational energy is know it is not hard to calculate the angular velocity and launch velocity. This equation can be used

$$E_r = \tfrac{1}{2}I * \omega^2 \quad \text{(equation 5.2.5)}$$

where E_r is the rotational energy in Joule, I is the moment of inertia in kilogram meter squared and ω is the angular velocity in radian per second [1]. If the angular velocity is multiplied by the distance of a point of that object than the tangential velocity is obtained. So

$$V_o = L_p * \omega \quad \text{(equation 5.2.6)}$$

where V_o is the launch velocity of the projectile in meter per second and ω is the angular velocity in radians per second.

5.3 Calculating the distance of the projectile

The distance that the trebuchet will throw can be calculated using two methods. First the analytical method will be elaborated. The y-coordinate as a function of time has the value of

$$p_y(t) = y_0 + sin(\frac{\pi}{2} - \Theta_e) * V_o * t - \frac{1}{2} g * t^2 \text{ (equation 5.3.1)}$$

where t is the time since the launch of the projectile. The x-coordinate (which is also equal to the distance thrown) is

$$p_x(t) = cos(\frac{\pi}{2} - \Theta_e) * V_o * t - cos(\Theta_e) * (L_p + L_c) \text{ (equation 5.3.2)}$$

where t is the time since the launch of the projectile again [1]. Now a system with two equations and three unknown variables (t, $p_y(t)$ and $p_x(t)$) is obtained. But luckily one of the unknown variables can be determined via the desired condition. $p_y(t)$ is zero because that is the moment the projectile hits the ground. From equation 5.3.1 t can now be determined

$$t = \frac{\sqrt{2*g*y_o + (V_0 * sin(\frac{\pi}{2} - \Theta_e))^2} + V_0 * sin(\frac{\pi}{2} - \Theta_e)}{g} \text{ (equation 5.3.3).}$$

This t then can be entered in equation 5.3.2 and the distance that the trebuchet will throw is obtained.

The second method is the numerical method. This method is based on differential equation and can be solved by making tiny steps in time (called ticks) and calculating the changes during that delta time (dt in seconds). With this method air friction can can also be taken into account. The friction force is

$$F_d = \frac{1}{2} \rho_g * C_d * A * V^2 \text{ (equation 5.3.4)}$$

where ρ_g is the density of the air in kilograms per cubic meter, C_d is a dimensionless number based on the shape on the projectile, A is the cross sectional area of the projectile in square meter and V is the (absolute) speed of the projectile in meter per second. For both the x and y direction the acceleration can be calculated

$$a_{x_i} = -\frac{V_x * F_d}{V * m_p} \text{ (equation 5.3.5)}$$

$$a_{y_i} = -g - \frac{V_y * F_d}{V * m_p} \text{ (equation 5.3.6).}$$

These differential equations can then be solved numerically with certain initial conditions and stopping conditions.

5.4 Matlab bruteforce optimisation

An analytical solution was tried (see appendix chapter ‘10.3 Attempt of analytical optimisation’) but yielded little results and since a Matlab model needed to be made anyway a brute forcing algorithm was written in Matlab to optimize the trebuchet. This algorithm calculated the launch velocity of the projectile and how far it would go in a analytical way or in a numerical way (with air friction). It tried all combinations of L_c, L_p and Θ_e and determined for each L_p what the optimal configuration was for throwing as far as possible and exported those configurations to a csv file (for more documentation see appendix chapter ‘10.2 Documentation Matlab bruteforce’). This approach did yield results.

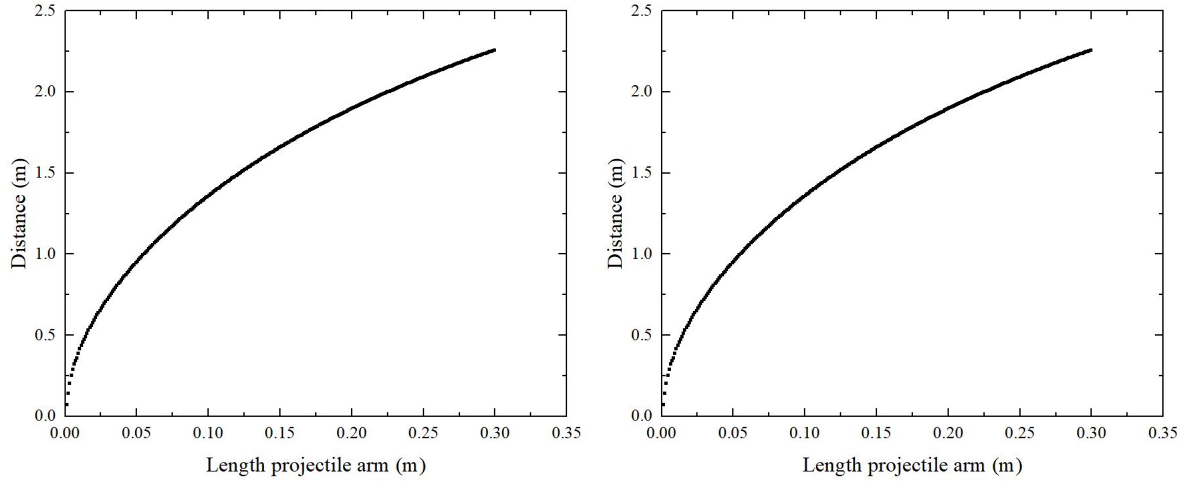


Figure 5.4.1: The throwing distance (m) in the optimal configuration as a function of the length of the projectile arm (m) without air friction (left) and with air friction (right).

In figure 5.4.1 it can be seen that the friction has nearly no effect on the distance the trebuchet throws. This is because the velocity and the cross sectional area of the projectile are relatively low/small while the mass is relatively high which makes it so that the air friction has little impact.

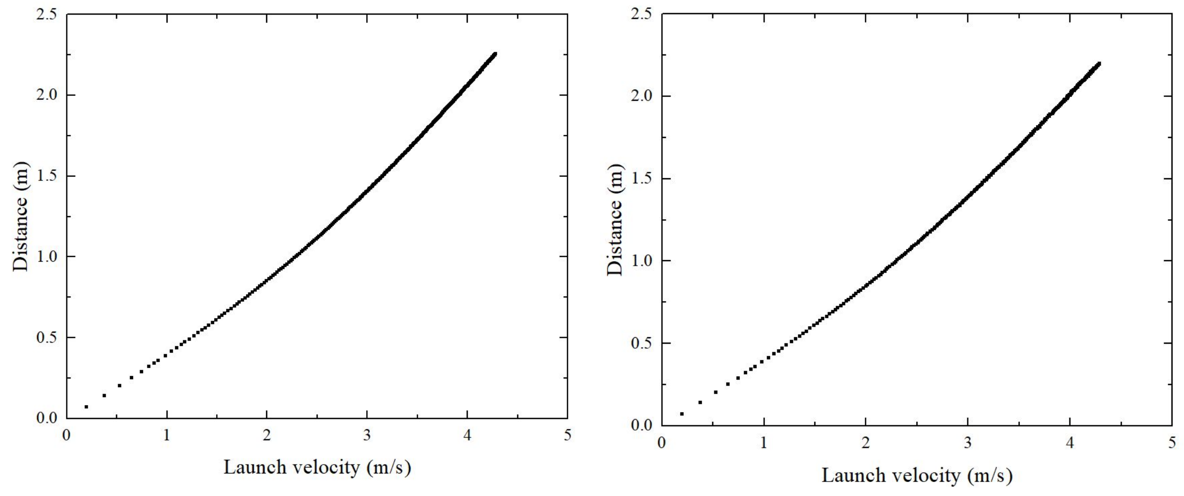


Figure 5.4.2: The throwing distance (m) in the optimal configuration as a function of the launch velocity (m/s) without air friction (left) and with air friction (right).

In figure 5.4.2 it can be seen that the throwing distance scales (nearly) linearly with the launch velocity and that air friction has little impact.

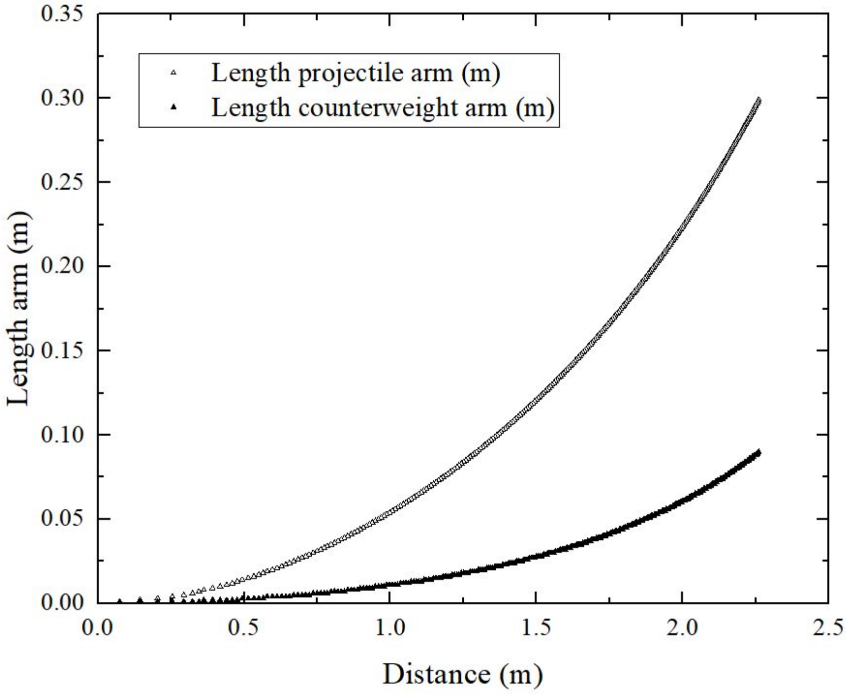


Figure 5.4.3: The length of both arms (m) in the optimal configuration as a function of the throwing distance (m) without air friction.

As can be seen in in figure 5.4.3 the speed at which the length (of both arms) increase as function of the distance is positive. From this one can conclude that the longer the arms the further the distance it can throw but this process has diminishing returns. To find the exact dimensions of the trebuchet the spreadsheet file on which these graphs are based can be viewed and the desired optimal trebuchet can be picked.

5.5 Strength of the trebuchet

It is also crucial that the trebuchet doesn't collapse when firing. The most intense point during the launch is when the arm collides with the part that is stopping it. To make calculating if it will break easier some assumptions have been made. First of all it is assumed that all the energy in the counter-weight gets absorbed by the stopping blok also during the collisions a constant force stopping the arm is assumed.

Next the buckling force needs to be calculated because if the force in the trebuchet is higher than the buckling force it will break. To do that we need the second moment of area of the frame which will be calculated. Because the hight and the width of the beam are the same $I_{xx}=I_{yy}$ the formula for the second moment of area will be

$$I_{xx} = I_{yy} = I_{beam} - I_{gap} = h^4 beam/12 - h^4 gap/12$$ (equation 5.5.1).

Using this the buckling force F_b will be determined with the formula

$$F_b = \pi^2 * E * I/L$$ (equation 5.5.2) [1].

Next the expected collision force is going to be calculated and compared against the maximum collision force. Since the counter-weight and projectile are not relevant the inertia becomes

$$I = \frac{1}{3}\rho * (L_p{}^3 + L_c{}^3)$$ (equation 5.5.3)

Since the angular speed is known the energy can be calculated according to equation 5.2.5. If a reasonable amount of bending (B in meter) is also estimated that the force on the stopping block becomes

$$F_c = \frac{\rho*(L_p^3+L_c^3)*\omega^2}{3*B} \quad \text{(equation 5.5.4)}$$

If this force is smaller than the maximum collision force than the stopping block and arm will not break.

6 Specific design and dimensions

The physics of the trebuchet gave results. Those results were used to make an specific design for this trebuchet. In this chapter the decisions made in the design process for each loose part are described.

6.1 Frame and triggering mechanism

The frame (figure 6.1.1) has several functions. First of all it keeps the trebuchet steady when the trebuchet launches the projectile. The frame contains four main beams, which will hold the trebuchet together. The main frame contains four beams, because this way the counterweight can fall vertically in between the beams. The dimensions of the beams are 10 x 10 x 260 mm. To save materials in the main frame the walls of the beams are hollow and they are 2 mm width. The counterweight won't drop to the bottom of the frame, this is caused by the length of the launching arm. The optimal dimensions of the arm are determined with the Matlab script. The dimensions of the arm will be further discussed in chapter 6.3 When the trebuchet fired the projectile, the counterweight fell 10.8 cm. To prevent the arm and the rest of the trebuchet from collapsing, there needs to be a block in the frame to stop the counterweight from falling further than the 10.8 cm. the stopping blocks are placed on the side of the main frame and the dimensions are 10 x 10 x 10 mm.

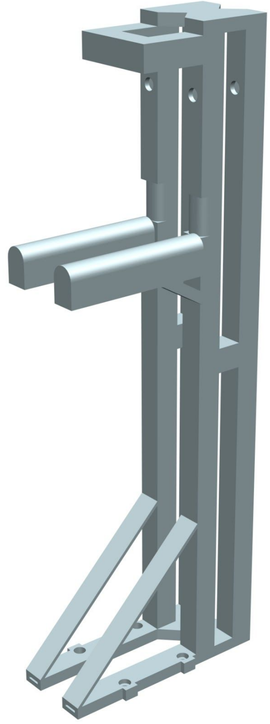


Figure 6.1.1: The frame.

The frame also contains two smaller diagonal beams, these beams connect the bottom of the frame to the beams of the frame. The function of the bottom is connecting the main frame to the steel base plate. According to the buckling strength, which is calculated with equation 5.5.3, the frame will not break after firing the projectile. So no extra reinforcements are added to the frame.
It's important to stop the arm in the optimal angle, this way the projectile will travel the biggest distance. This trebuchet a optimal launching angle of arm of 63 degrees (more about this in chapter 6.3). There has to be a stop block, so the arm will stop at this angle. The stop block is placed on top of the beams of the frame. The dimensions of the stop block are 26.5 x 35 x 10 mm. To save more material, the stop block has the shape of an U. A notch is als placed in the beams of the frame. The notch is round and has a 5 mm radius, which is the same as the radius of the wheels of the arm (see chapter 6.2). The function of the notch is to let the arm as vertical as possible. The arm needs to be as vertical as possible or else it won't meet up with the constraint: the projectile has to start above the baseplate (see chapter 3.1). The notch is placed directly above the rails.

Another constraint is to add a trigger mechanism to the trebuchet, so the trebuchet can put into motion with one hand. To meet up with this constraint there is a pin which will go through holes in the beams of the frame. The counterweight will rest on the rods of the pin. To set the trebuchet into motion, the pin needs to be pulled out of the trebuchet. This can be done with one hand.

6.2 Rails and wheels

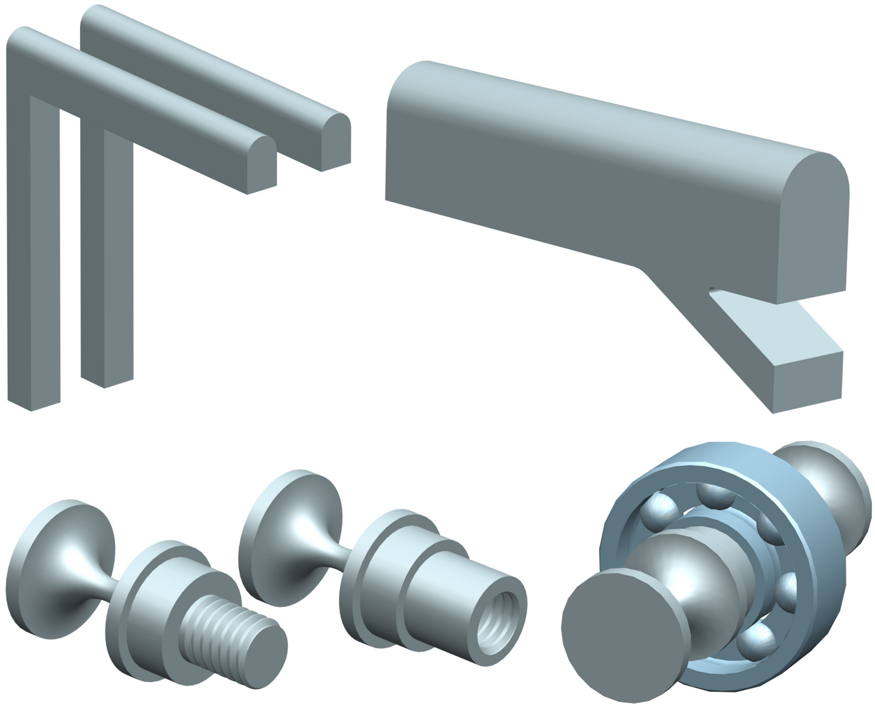


Figure 6.2.1: Various parts of the trebuchet. Two designs for the rails (top) and the wheels (bottom)

The function of the rails is to support the arm during the firing of the trebuchet. Two designs were made for the rails (see figures). In figure 6.2.1 the rails have a 90 degrees angle and it supports on the table. The design in figure 6.2.2 is attached directly to the main frame of the trebuchet. With the help of the Matlab script, the dimensions of the arm were determined. The arm is pretty long compared with the height of the main frame. One of the constraint is: the projectile has to start above the baseplate (see chapter 3.1 RPC's). To achieve this constraint, the rails need to be attached quite high to the main frame. For the final design it is easier to use the second design than the first design. A big advantage is that the second design use less material than the first design. To save even more material, the rails are hollow.

It's very important that the arm starts on the rails before the trebuchet fires the projectile. When it doesn't start on the rails there is a possibility that the arm starts bouncing on the rails and even falls of the rails. To let the arm roll across the rails, the top part of the rails can't be flat, because this can cause the arm to roll of the rails. The rails need to have a round top (see figure 6.2.1). The width of the beams of the trebuchet are 10 millimeter, so the radius of the round part is 5 mm. The wheels need to be attached to the arm. The wheels are shown in figure 6.2.1**.** The notch of the wheels has a radius of 5 mm so it fits perfectly on top of the rails. The diameter of the whole wheel is 12 millimeters. The two wheels can be screwed together in to the arm. A ball bearing fits between the two wheels . The wheels and ball bearing in the arm are shown in figure 6.2.1. The final dimensions of the rails are: 60 millimeter long, 10 millimeter wide and 21.35 millimeter high. The rails are hollow, so the sides of the rails are 2 millimeter wide. The round part of the rails has a width of 1 millimeter.

6.3 Arm and launching angle

Before the specific design of the arms can be determined a approximate design is needed so it can be entered in the Matlab code to determine the dimensions of arms. A hollow beam with a thickness of two millimeters was assumed. Since, as shown in chapter '5.4 Matlab bruteforce optimisation', it is always better to have longer arms a upper limit was needed. This limit was the height of the box in which the arm needed to be printed since it was preferred to print the arm in one piece. This means the total length of the arm couldn't be more than thirty centimeters long. This constraint and the results of the Matlab code resulted in a optimal dimensions of the trebuchet. The projectile arm needed to be 19.8 centimeters long, the counter-weight arm needed to be 5.2 centimeters long and the ending angle needed to be 63 degrees. This trebuchet could fire up to 1.89 meter if no friction of any kind is assumed and 1.85 meter if only air friction is taken into account.

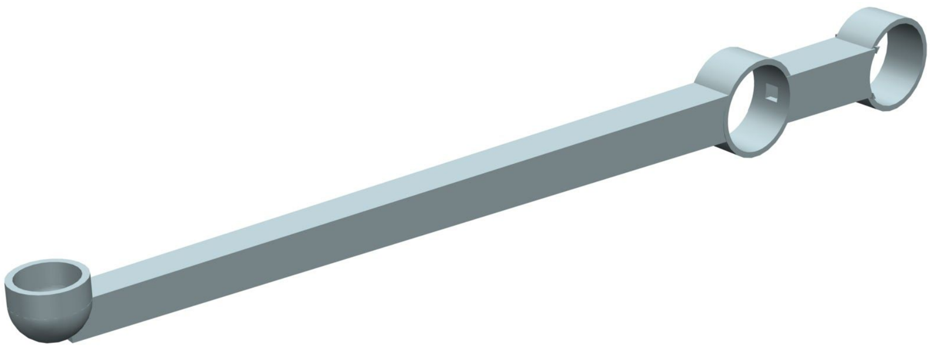


Figure 6.3.1: The design of the arm.

Although the dimensions of the arm were determined the shape of the arm still had to be designed. the counterweight arm had to be attached to the launching arm. also the launching arm have to be able to move around the counterweight with as least friction al possible. This means there has to be a gap at the end of the launching arm in which the counterweight fits. also the wheels which have to roll over the rail have to be attached to the launching arm so between the counterweight arm and the projectile arm there is also a hole needed. At the other end of the arm there has to be a bucket for the projectile. Because the least possible is desired between the wheels and the launching arm and between the counterweight arm and the launching arm in the holes in the arm a bearing will be placed to reduce the amount of friction. This means this description leads to an arm with a hole at one end, a bucket at the other end and another gap in between them. This would use very much material so this design has to be improved to save material. To do this the arm is mad non-uniform. At the end of the bucket has an crossarea of 64mm and at the other end it has a cross area of 175.5 millimeter.

6.4 Counterweight

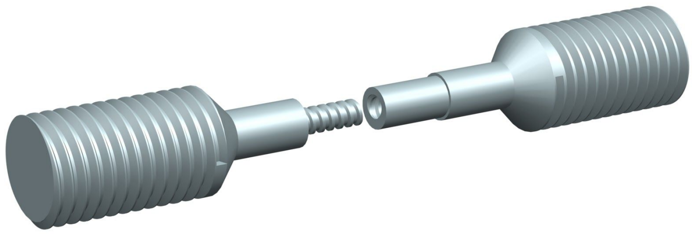


Figure 6.4.1: The attachment point for the M20 nuts.

The counterweight has to hold the four M20 nuts, fit in the frame, be attached to the arm and it has to rotate independently from the arm. Two bearings are used to make sure the arm and the counterweight arm can rotate freely from each other. The counterweight arm should go in the inner diameter of the bearing and the arm should go around the outer diameter of the bearing. Two seperate counterweight arm pieces are made with threads that fit into each other to fit the counterweight arm in the inner diameter of the bearing. The bearings are also used to add more weight to the arm.

To fit the counterweight arm in the frame the diameter of the counterweight arm is, not where the bearings are, nine mm so there is one mm space between the frame and the arm. Now the counterweight arm can move through the frame without much friction. The whole counterweight arm is made solid because of the weight attached to the arm. A hollow arm could more easily break. At both ends a thread is made to screw the M20 nuts on. The final design is shown in figure 6.4.1

6.5 Projectile

With the assignment to design a trebuchet, came an assignment to design the projectile it would launch. First the boulder or ball, the most commonly used projectile was analyzed. This sort of projectile doesn't have many problems but it isn't the optimal form either. When the projectile is round it has no front- or back side, this allows it to be fired without an accurate release angle. This is the most commonly occurring problem with many other shapes of projectiles. For example, if the projectile were made in the shape of a bullet, it would be more aerodynamic, but for it to be effective, it has to fly straight with the point forward. To achieve this with a trebuchet is very near impossible. Furthermore, if the throwing angle were not optimal in such a case, this could result in a wobble in the flightpath. As a direct result of this, the projectile would actually lose speed.

There was another projectile considerable. This projectile had the shape of a tear. The reason that this is a viable option is that because of its shape it should be able to straighten itself when in the air. So even if it is not thrown with the perfect angle it would stop wobbling and thus travel a greater distance. The only problem with this design is that for the projectile to stabilize itself, there needs to be a large enough airflow around it, so it needs to go fast enough.

In, conclusion the decision has been made to keep the projectile simple. The projectile shall be simply shaped like a ball. The reasoning behind this is that most other objects would wobble in the air, making them less efficient. The next step is determine the optimal dimensions of the ball. The dimensions, such as the optimal diameter of the ball, were determined with the help of the Matlab script. The optimal diameter for the projectile is 10.087 millimeter. Also the mass of the projectile is determined with the Matlab script. The script ran two times. One time with the mass of a solid ball made out of the printing material nylon. The second time the script was used with the extra mass of a M6 nut, which is placed inside the projectile. According to the results of the script the the projectile without a M6 will reach a further distance than a projectile with a M6 Nut inside. The final mass of the projectile is 0.499 gram.

7 Results
In the results the results of different methods will be summarized. First of all the results of the Matlab script, NX-motion and the reality will be described. After that the results will be compared to each other.

7.1 Matlab

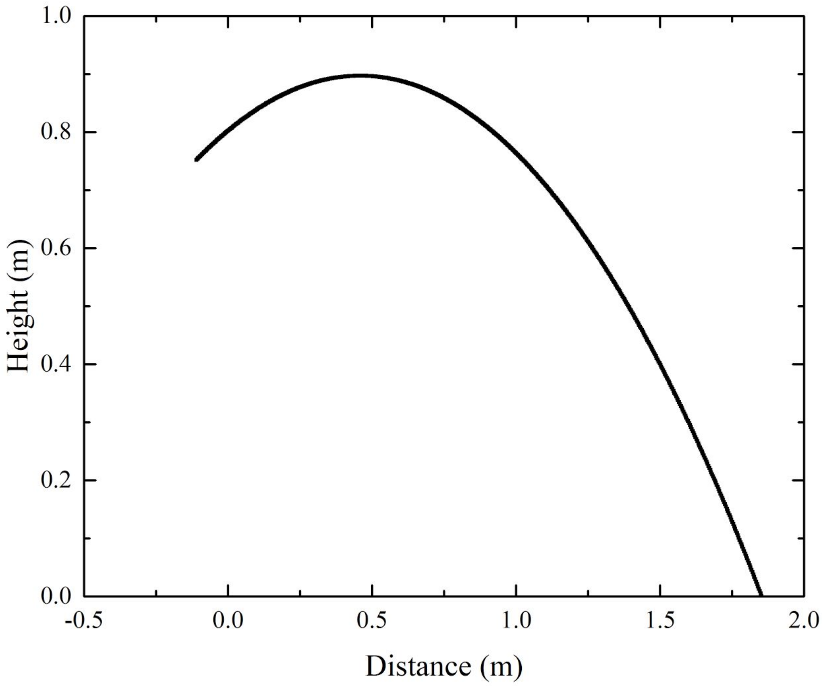


Figure 7.1.1: The motion of the projectile with air friction.

The Matlab code as described in chapter '5.4 Matlab bruteforce optimisation' and appendix chapter '10.2 Documentation Matlab bruteforce optimisation' could be slightly adjusted such that it could calculate the results for just one design of the trebuchet. The design as described in chapter '6 Specific design and dimensions' could fire up to 1.89 meter if no friction of any kind is assumed and 1.85 meter if only air friction is taken into account. As calculated and seen in figure 7.1.1 air friction has nearly no impact on the projectile.

7.2 NX-motion
To do the calculation in NX motion a feature was used where an object can be selected and NX motion will give the displacement of that object. First the vertical distance was calculated against the time. The ground is 75 centimeter below the projectile so the time the projectile was negative 0.75 meter was taken. This time was looked up in the graph for the horizontal distance. This gave a total throwing distance of 2.29 meter.

For NX motion 3D simulation and animation contact between links was used. This was to let the swinging arm stop when it hit the stopping blocks. Also revolving and sliding joints were used for the wheels that is rotate and slide along the rails. The projectile was also in 3D contact with the throwing basket. This resulted in an movement in the z axis and was probably the projectile wobbling in the basket. The program automatically determines the weight of all the parts. But in the file there weren't ball bearings so it was needed to apply an additional force to the sling arm. This was done by calculating the gravitational force on the bearings and applying this force to the origin of the part where the ball bearings are attached.

7.3 Reality

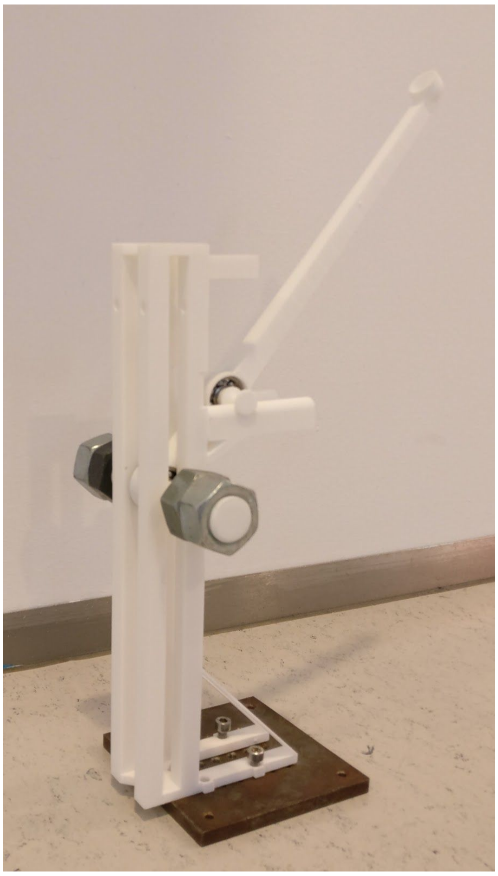


Figure 7.3.1: Picture of the trebuchet after being fired.

In reality the trebuchet fires on average 2.9 meter with a standard deviation of 0.2 meter. however,as can been seen in figure 7.3.1, it also turned out that the stopping block for the counterweight was placed to high which increases the launching angle (but decreases the launch velocity). Besides this flaw the trebuchet is stable and doesn't break.

7.4 Discrepancies

The large discrepancy between the Matlab script, NX-motion an reality might be caused by several things. First of all the assumptions of the Matlab script such as the uniform density of the arms but especially the fact that the floating arm trebuchet was promoted as a normal trebuchet might cause a difference. Also, the counterweight might be heavier than just the uniform arm and the mass of the nuts because several bearings have been added to the design which would have also caused a higher launch velocity. Besides this the most significant factor might have been that the height of the trebuchet wasn't added to the starting height of the projectile. For example when the extra height is accounted for and a bit of extra velocity is added the measured 2.9 meter is reached.

NX-motion was off because the density of the arms was set significantly higher than it actually was.

8 Conclusion and evaluation

In this chapter the design process is evaluated. The results are compared to the RPC's and the different choices are evaluated.

The final product can shoot a projectile with the use of a counterweight within the given constraints, so the final product is a success. The design is also fairly simply made. The whole mechanism involves only the counterweight arm, the arm and the wheels on the rails. It's very easy to set up too. The final design also uses material very efficiently with as much hollow material as possible but also with solid material on the weak spots. However, some spots were not very well made like the rails. The rails stick a bit out from the frame and it does not line in perfectly with the inlets. The trebuchet is very stable and rigid and it doesn't break after firing. Also, the trebuchet does not fire four meters or further. So not all the RPC's are met.

What the could have been better is the the connection between the different parts of the trebuchet. These connections could have been made more accurately. Also, the design is made so that the counterweight does not drop from the top to all the way down. This way, a lot of potential energy is wasted. This could have been done better.

In conclusion, the final product works very well but could have done better with a few changes in the design.

9 Bibliography

[1] H.D.Young, R.A.Freedman. (2016). *University physics with modern physics (14th edition)*. University of california

10 Appendix

10.1 Overview formulas

$E_g = m * g * h$ (equation 5.2.1)

$E_g = g * (sin(\Theta_e) - sin(\Theta_s)) * (m_c * L_c + \frac{1}{2}\rho_a * L_c{}^2 - \frac{1}{2}\rho_a * L_p{}^2 - m_p * L_p)$ (equation 5.2.2)

$I = \sum_{i=o}^{N} r_i{}^2 * m_i$ (equation 5.2.3)

$I = m_c * L_c{}^2 + \frac{1}{3}\rho_a * L_c{}^3 + \frac{1}{3}\rho_a * L_p{}^3 + m_p * L_p{}^2$ (equation 5.2.4)

$E_r = \frac{1}{2}I * \omega^2$ (equation 5.2.5)

$V_o = L_p * \omega$ (equation 5.2.6)

$p_y(t) = y_0 + sin(\frac{\pi}{2} - \Theta_e) * V_o * t - \frac{1}{2}g * t^2$ (equation 5.3.1)

$p_x(t) = cos(\frac{\pi}{2} - \Theta_e) * V_o * t - cos(\Theta_e) * (L_p + L_c)$ (equation 5.3.2)

$t = \frac{\sqrt{2*g*y_o+(V_0*sin(\frac{\pi}{2}-\Theta_e))^2}+V_0*sin(\frac{\pi}{2}-\Theta_e)}{g}$ (equation 5.3.3)

$F_d = \frac{1}{2}\rho_g * C_d * A * V^2$ (equation 5.3.4)

$a_{x_i} = -\frac{V_x*F_d}{V*m_p}$ (equation 5.3.5)

$a_{y_i} = -g - \frac{V_y*F_d}{V*m_p}$ (equation 5.3.6)

$I_{xx} = I_{yy} = I_{beam} - I_{gap} = h^4 beam/12 - h^4 gap/12$ (equation 5.5.1)

$F_b = \pi^2 * E * I/L$ (equation 5.5.2)

$I = \frac{1}{3}\rho * (L_p{}^3 + L_c{}^3)$ (equation 5.5.3)

$F_c = \frac{\rho*(L_p{}^3+L_c{}^3)*\omega^2}{3*B}$ (equation 5.5.4)

10.2 Documentation Matlab bruteforce optimisation

10.2.1 How to use Matlab script

To run the Matlab script the following steps can be carried out:

1. Download the Matlab script from Github: https://github.com/TimHeiszwolf/DMT_Optimisation.
2. Open the ‘Controller.m’ and type (in the command window) ‘Controller()’. The script will now start.
3. The script will prompt if a manual data entry is wanted. If yes then the desired ranges and settings can be entered if not then it will proceed with default settings.
4. The script will ask if a numerical solution for calculating the distance of each possible trebuchet should be used. If numerical it will ask for some more input data and calculate the distances stepwise (and if wanted with friction). If not numerical it will calculate the distances analytically with kinematics.
5. Once done calculating the distances the script will ask if the raw output data (all the possible trebuchet configurations and their solutions) needs to be exported. If so a filename will be asked and the script will write the output file to the Matlab folder.
6. The script will also ask if the analyzed data needs to be exported. If so it will again ask for a filename and export it to the Matlab folder. After this the script is done.

10.2.2 How does the Matlab script function?

This explanation of the Matlab script leaves out some of the bloat such as the ‘Testing.m’ and ‘Play_audio.m’.

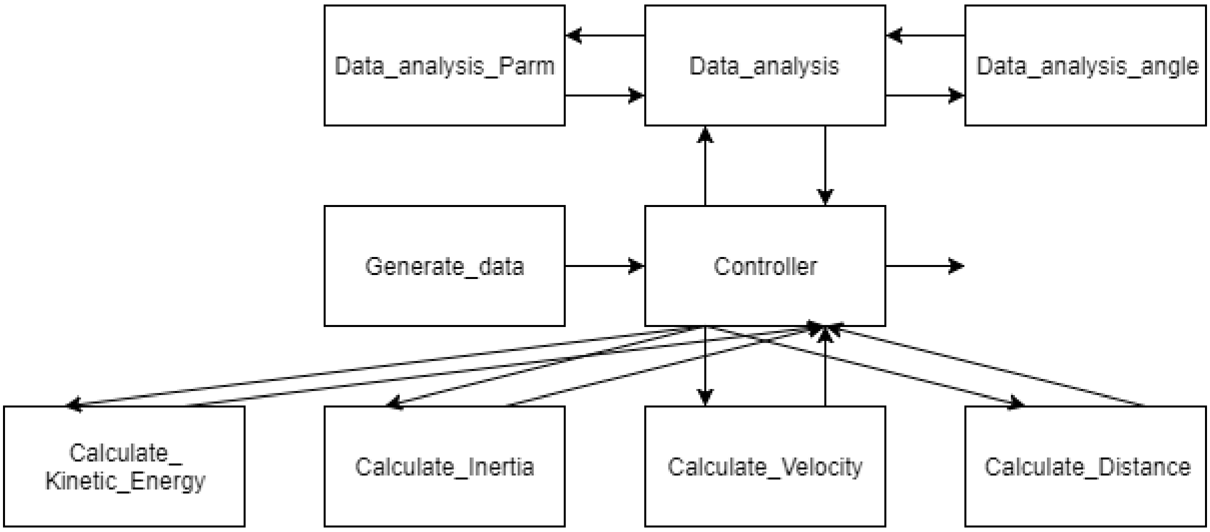


Figure 10.3.2.1: Flowchart of the Matlab script

1. When the program is started ‘*Controller*’ is going to ask ‘*Generate_data*’ for data. This is going to be a list of different combinations of variables for the brute forcing. The user is asked for for input but can also choose to run with preset variables.
2. ‘*Controller*’ then takes one item from the list of data.
3. ‘*Controller*’ gives the item of the data to ‘*Calculate_Kinetic_Energy*’ and ‘*Calculate_Inertia*’.
4. ‘*Calculate_Kinetic_Energy*’ and ‘*Calculate_Inertia*’ calculate gravitational energy and inertia and return these to ‘*Controller*’.
5. ‘*Controller*’ gives the item of the data and the results of ‘Calculate_Kinetic_Energy’ and ‘*Calculate_Inertia*’ to ‘*Calculate_Velocity*’.
6. ‘*Calculate_Velocity*’ calculates the velocity and returns this to ‘*Controller*’.
7. ‘*Controller*’ gives the item of the data and the result of ‘*Calculate_Velocity*’ to ‘*Calculate_Distance*’ or '*Calculate_Distance_numerical*'.
8. ‘*Calculate_Distance*’ or '*Calculate_Distance_numerical*' calculates the distance travelled and returns this to ‘*Controller*’.
9. ‘*Controller*’ stores the result of ‘*Calculate_Velocity*’ and ‘*Calculate_Distance*’ in the item of the data.
10. ‘*Controller*’ overwrites the old item of the data in the data list with the item of the data from step 9
11. ‘*Controller*’ repeats steps 2 till 10 for each item in the list of data.
12. ‘*Controller*’ asks the user if he wants to export the raw data. If so, it exports the data.
13. ‘*Controller*’ gives ‘*Data_analysis*’ the data.
14. ‘*Data_analysis*’ gives the data to ‘*Data_analysis_angle*’.
15. ‘*Data_analysis_angle*’ determines the optimal angle for each configuration of the arms and only returns these optimal data points to ‘*Data_analysis*’.
16. ‘*Data_analysis*’ gives the results from ‘*Data_analysis_angle*’ to ‘*Data_analysis_Parm*’.
17. ‘*Data_analysis_Parm*’ determines the optimal length of the projectile arm for each counterweight arm and only returns these optimal data points to ‘*Data_analysis*’.
18. ‘*Controller*’ asks the user if he wants to export the analysed data. If so, it exports the data.

10.3 Attempt of analytical optimisation

The attempts of analytical optimisation were done using wolfram mathematica. Wolfram mathematica is a piece of software that can do a lot of complex mathematics automatically such as substituting, differentiation, solving (systems) of equations and much more.

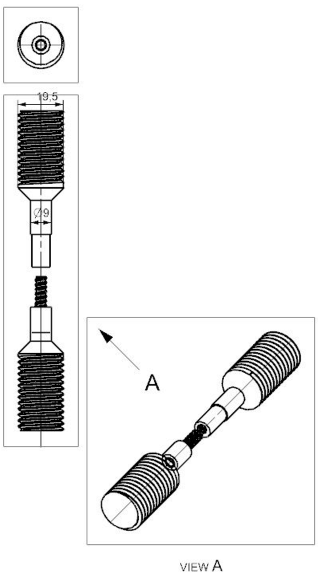
First the optimal angle was determined. This was done by making an analytical solution for the distance thrown and setting the derivative of that solution equal to zero and solving for the angle. This didn't work because in the starting height of the projectile motion also depended on the angle. So the assumption was made that the height on which the trebuchet is standing is way taller than the height of the trebuchet and this equation was obtained

$$\Theta \;=\; \frac{\pi}{2} - arccos\left(\frac{\sqrt{V_0^{\,2}+2*g*y_0}}{\sqrt{2*V_o^{\,2}+2*g*y_0}}\right) \text{ (equation 10.4.1).}$$

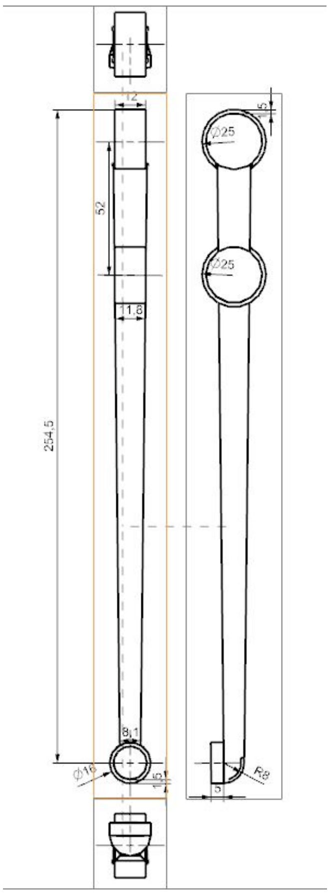
Next an attempt was made to optimize the dimensions of the trebuchet but this failed since, as later discovered, there is no optimum length of arms that is finite (see chapter '5.4 Matlab bruteforce optimisation').

10.4 NX-Draft

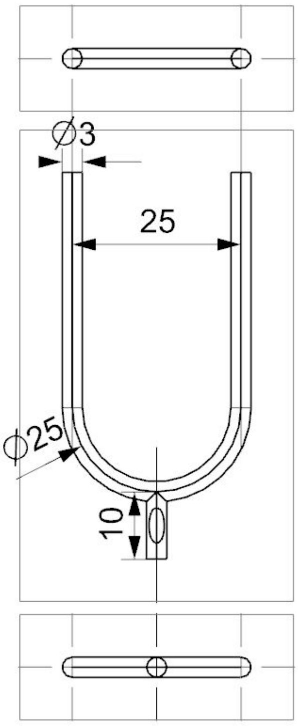
NX Draft is a part of Siemens NX-10, this program allows you to make sketches of a design model or parts of it. With sketches of an object from different side views and the measurements of this part it is possible to remake the model.
Below here are the drafts of all the part of the trebuchet:



Counterwieghtarm

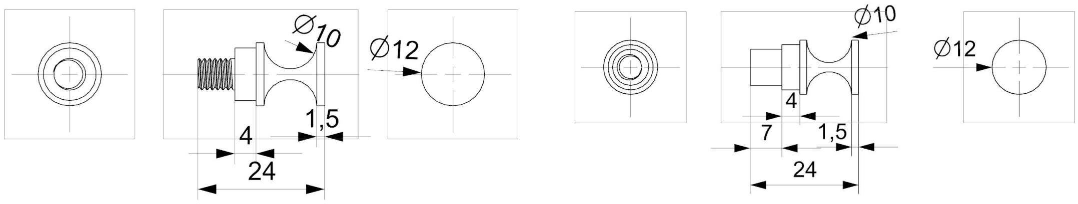


Arm

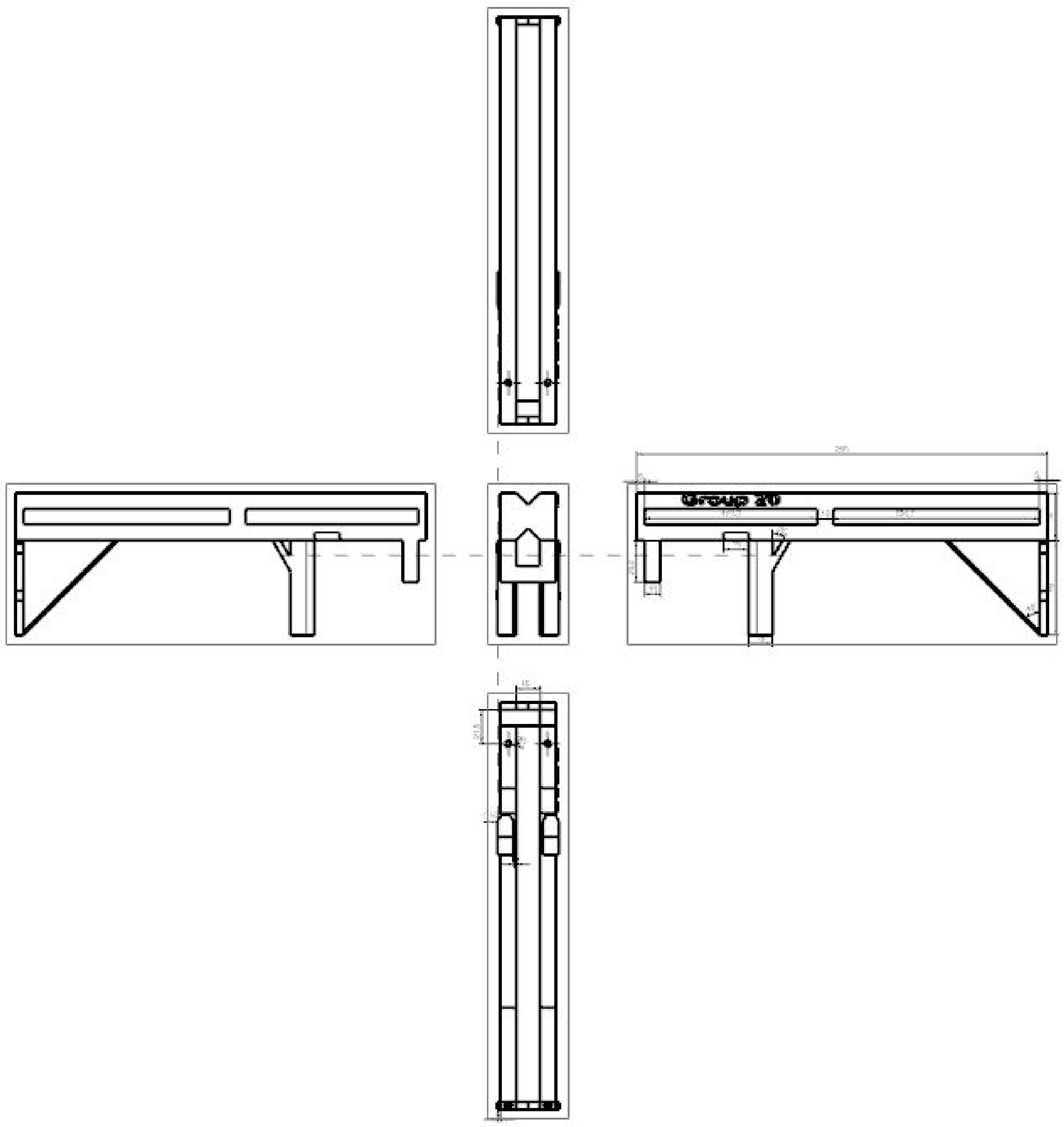


Pin

Wheel



Frame



10.5 Exploded view