The design of the robot „Rattlesnake“
Julian Hammerl and Lisamarie Schuster
Vienna Institute of Technology (TGM) – iBot (13-0233)

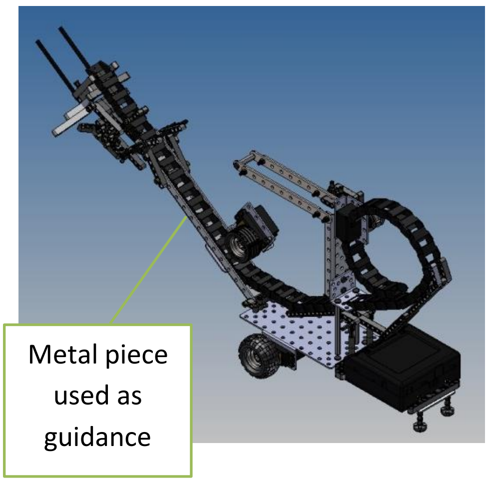
The design of the robot „Rattlesnake“

1 Introduction

When the team iBot unpacked all the parts of the Botball set during the Botball workshop, we were astounded of how many different pieces there were. We took a closer look at them, one by one, and tried to figure out what they were for. There were some parts we couldn't find any usage for, but also weren't quite interested in. However, one of those strange parts drew our attention, because it made such a funny noise when you rolled it up: The IGUS chain.

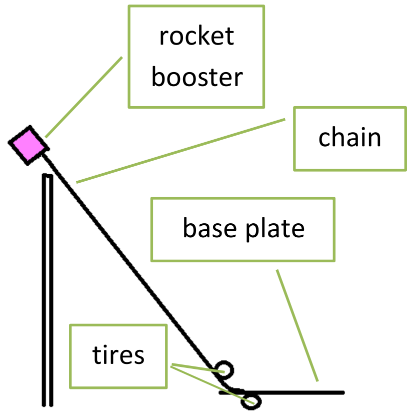
We are a very playful group, so everytime one of us was underemployed, he would take the chain and play with it. Some members of other groups already looked at us in a strange way because of the noise we produced with the chain.

During the first day, the game table was presented and this was how we discovered that the rockets in the center are 80cm high. Everybody shook their heads in disbelief at this enormous height. However in the meantime, one member of the team iBot was playing with the chain and when he heard the number of centimeters, took a look at what was lying in his hands. After a quick measurement, it was clear that the chain was even longer than the rockets when rolled out. It also occurred to us that the chain fitted almost perfectly in a metal piece, which could be used as guidance. From then on, it was only a matter of time until we found a proper way using the chain to transport the rocket boosters up to the top of the rockets and created our robot called "Rattlesnake", which you can see to the right (modelled in Autodesk™ Inventor® by one of our team members).



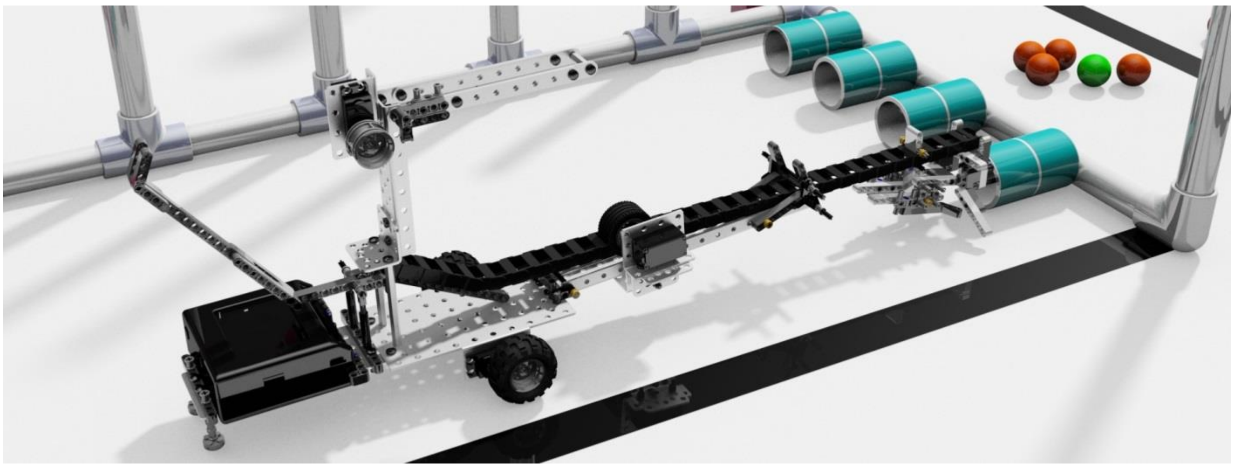
2 The basic idea

Basically, we discovered that the chain can be rolled up in one direction, but is unmovable in the other direction and acts – almost – like a solid bar. Due to the fact that the starting height of the robot at the beginning is only 38cm, we had to take advantage of this feature. We built a section at one end of the chain which gets pushed in the large hole in the rocket boosters. By the usage of a rope roll, we created the ability to change the angle in which the chain gets rolled up.

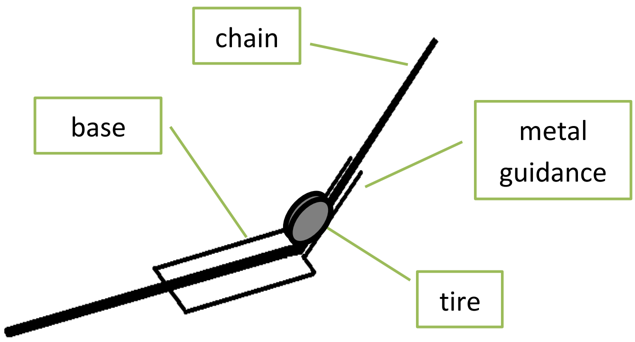


The robot is intended to lift the rocket boosters onto the rockets in the center of the game table. In order to do this, first of all the guidance gets tilted in a horizontal position and the chain gets pushed forward so that the axles move in the hole in the rocket booster. Then the angle of the guidance gets changed again in order to lift the rocket booster in a proper position. This depends mainly on the weight of the rocket booster, so whether it is a single, double or triple booster. This is followed by a 90° turn and alignment on the game table depending on which rocket the booster is supposed to get lifted onto. The chain gets pushed upwards, even a little over the rockets (as you can see on the schematic drawing to the right), so that the booster gets caught at the rocket when the chain is moving backwards again. The end of the chain, also called lift part, mostly stays in the booster about half the way down, while the chain is constantly moving backwards and then the lift part abruptly pops out, which causes the chain to struggle. Sometimes it even falls over completely, but we found a solution for this case (you will read more about complications we had to face in section “4.”). The chain rolls up completely and the robot moves on to the next booster.

3 Mechanical design in detail

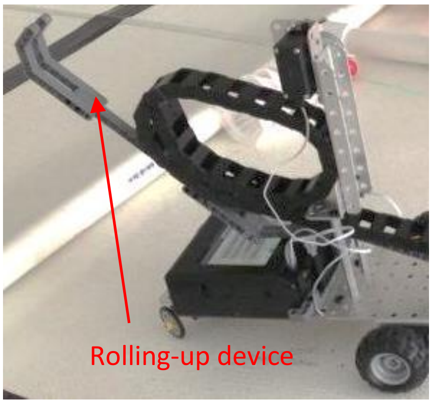


3.1 The rolling up mechanism



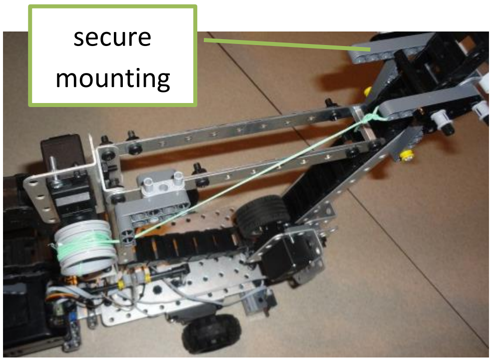
A big topic while developing the idea properly was how we could roll the chain up and out again. The solution was quite simple: We placed a motor near the hinge of the guidance and screwed a tire onto it. The friction between the rubber from the tire and the plastic from the chain was actually high enough to move the chain upwards and downwards, respectively forwards and backwards (depending on which angle the guidance is in)! The next thing we had to come up with a solution for was the fact that the chain of course didn’t ROLL up itself, but acted like a solid bar after a downward movement, as you can see on the schematic drawing above.

This is the reason why we built an almost half-round device made of Lego parts to give it the impulse to bend and in consequence roll up.

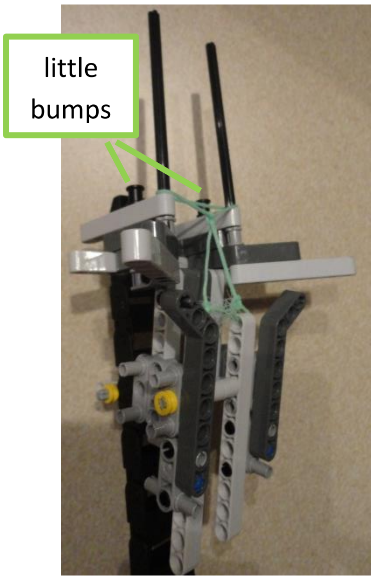


3.2 The mechanism for changing the angle

The guidance got secured to the base plate by creating some kind of a hinge using a long axle, as you can see on the photo to the right. A secure mounting on top of the guidance was built to prevent the chain from bending over and to hold it in the guidance. As you can see to the left, we knotted a rope to the secure mounting and led it to a vertical metal device, where another motor was mounted. By letting the motor turn, we can change the angle in which the chain is pushed forwards or backwards – the guidance can be tilted from a bit less than 90° to around 0° (you can see the latter on the photograph below the caption of section "3.")



3.3 The lift part



The picture to the left shows the part that is mounted to the one end of the chain: the lift part. The two black Lego axles form the most important function. By tilting the guidance in a horizontal position and pushing the chain forward, the axles get placed in the hole in the rocket boosters. For holding the booster reasonably in place, the two little black bumps on top of the whole device were mounted. The space between them and the axles is almost of the same span as the thickness of one rocket booster.

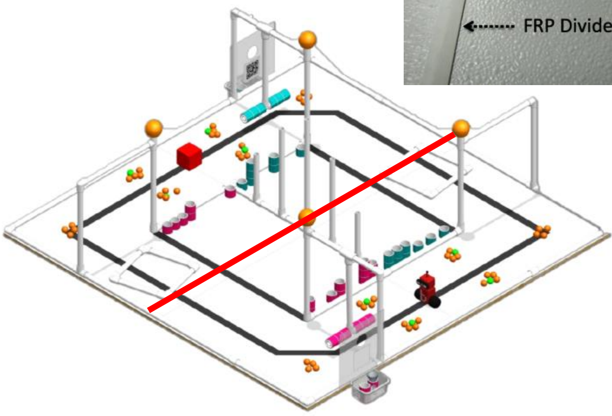
The rest of the device is mostly built in purpose of simplifying the alignment at the rocket directly. Without it, the chain often drifted to the right or left of the rocket, which caused the booster to not get caught on the rocket – it sometimes even fell off the lift part completely and landed on the other side of the game table. The long angular Lego pieces are for rough alignment. When they touch the rocket, the chain gets aligned to it automatically just by the weight of the booster pulling it down.

The lower part of the mounting imitates the round shape of the rocket so that the chain glides down the rocket smoothly and doesn't break out to any side.

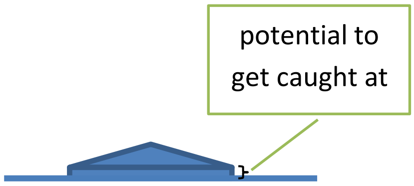
During testing, we also discovered that the whole mechanism worked way more precisely when the axles were mounted a little bit closer to each other. At first, we built the axles in a way that they would each be on one side of the rocket when dropping the booster. However, this caused it to pop backwards more extremely, so we mounted them closer together. The difference still was not enough, which is why we used a rope to pull them together just a tiny little bit more. Offsetting it one hole would have been much too much, so we had to do it this probably uncommon way.

3.4 Sensors used

For alignment, we concentrated on the ridge on the game table. It is located rectangular to the rockets and is caused by an FRP divider, which is placed at the seams between the large white FRP pieces. We marked it bright red in the picture to the right. At first, we found it a big disadvantage because we had problems stabilising the robot on it (this topic is described in detail in section “4.”). However, after we had come up with a solution for that, we started seeing the potential for alignment in the divider.



The touch sensors seemed the most suitable in this case and so we developed a device that is both able to move upwards and downwards a little while constantly sending a signal and also does not get caught at the small ridge that is marked at the schematic drawing to the right. This device was first mounted only at the very back section of the robot until we figured out that this system would only work when we would install a second sensor somewhere else, which we did, in between the two tires (you can see both sensors in the photographs below). The idea behind this is that the robot only stands in the very center of the table if BOTH sensors give a signal (so when both of them are on the divider). If only one of them sends a signal, the robot has to align another time.



Approximate cross section of the FRP divider

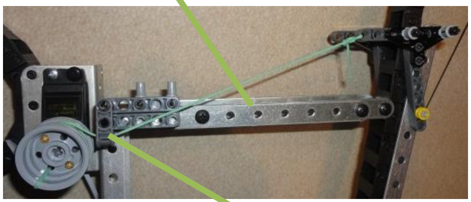
sensor at back section with stilts on each side

sensor between tires

In addition to this, we also mounted an IR sensor at the back for knowing when the robot drives over one of the black lines on the game table and en ET sensor at the very front for distance measurement.

We used another IR sensor for a probably very extraordinary purpose, but you will read more about this in the following section.

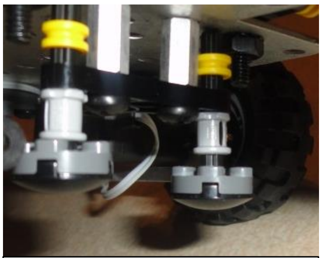
4 Problems faced



As mentioned above, the chain sometimes falls over backwards when it lets go of a heavy rocket booster on the rocket. At first we had many problems with this circumstance because the lift part got caught in the rolled up chain or the rope and therefore the chain could not get pushed further downwards.

This is why we built a horizontal metal guidance, on which the lift part or chain is supposed to fall when it pops backwards. The guidance is very straight and sleek so that there are no parts or small pieces where the chain could get caught at.

Another problem we had to face during testing was that the rope didn’t roll up properly anymore when the chain had fallen over and thereby also pushed back the guidance a bit. Then the rope lost its strain and got rolled up not at the rope roll itself, but besides it at the motor axis, for example. To avoid this, we mounted a small guidance for the rope, as you can see above.

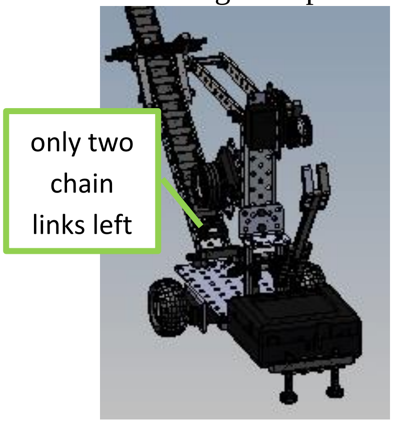


stilts at the front of the robot

As you may have noticed, heavy rocket boosters have caused many complications with our robot being comparatively light-weighted. When the lift part let go of the boosters at the rocket, not only the chain falls over, but also our whole robot tilted. Hereby, the tires formed the center of rotation, so we had to do something about it. For displacing the gravity center further to the back section of the robot, we changed the position of the tires and mounted them one centimetre closer to the front. In addition to this, we built a stilt at the very front of or robot, which should make the “new” center of rotation (you can see them in the photo near the next paragraph). Our whole robot is now in a tilted position.

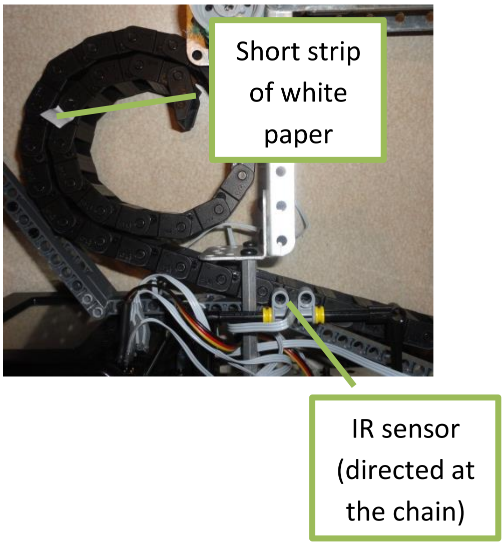
One topic is explained in section “3.d” already: the divider-problem. Now, we use it for alignment on the table but as mentioned before, it has caused us many sorrows until that point. We had worked with the original DemoBot installation using the metal ball for stabilisation in the backsection of the robot. The metal ball, however, always slipped to one side of the divider because its cross section is triangular and not rectangular at the top (this is also shown in the schematic drawing in section “3.d”). So we frantically looked for a component that would both have the ability to not get caught at the small ridge at the divider but could also stabilise the robot. We figured out that the only solution was building two separate stilts which would each be on one side of the divider. Finding a proper ending for the stilts was also a challenge but we managed this as well, using partly sphere-shaped Lego pieces, as you can see to the right. We used this technique at the front of the robot as well as in the back section on both sides of the sensors (see also photograph in section “3.d”).

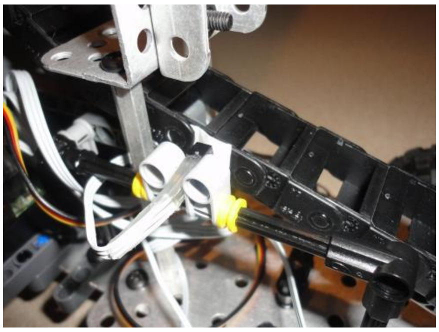
Eventually, there is one quite extraordinary solution we came up with and that is worth describing. The problem was that we never knew exactly when the motor for the rolling up



mechanism (the one with the tire on it) should stop turning and telling him how long he should turn just by giving him seconds was way too imprecise. It always stopped either too early so that the booster was a few centimetres off the rocket or too late so that the robot tilted and completely fell over because the gravity center moved over the edge of the robot or the motor led the complete chain into the guidance so that it could not get hold of it for moving it downwards again. The robot is right on the edge of having these two complications in the picture to the left. You can see how few chain links there are left behind the tire and if the lift part had a heavy rocket booster loaded, the robot already would have tilted and probably fallen over completely.

This is why we simply put a short strip of white paper between the chain links where we wanted the motor to stop turning. We mounted an IR sensor at the base plate at the suitable height so that it would always and only “see” the black chain links passing when the motor turned. But then at one point, the white paper stuck between the chain links would pass and the sensor sends a signal to the KIPR Link, telling the motor for the rolling up mechanism to stop turning.





This is where the sensor sends a signal and the KIPR Link tells the motor at the front section of the robot to stop turning.

5 Acknowledgements

First of all, we are extremely grateful for proALPHA® giving us this big, once-in-a-lifetime opportunity by sponsoring our team.

Special thanks to the whole PRIA team and especially to Mr. Dipl.-Ing. (FH) Mag. Gottfried Koppensteiner for all the support that is constantly offered us.

The modelled Rattlesnake robot will also be a big part of our presentation at the conference and we are planning on animating its movements on the PC to make our idea clear to everybody, not only by using pictures but also an animated video.

However, modelling the robot in the first place would not have been possible without the work that has been already done for us. The KIPR team modelled all the parts contained in one Botball set and made them accessible to all Botball teams via the KIPR Homebase platform, which is why we want to give a big Thank You to KIPR in general.

In addition to this, we want to thank Autodesk for giving its software Inventor to the TGM for free. Without it, we would not have been able to even open the Botball set content files provided by the KIPR team.