Machine Design

25-2-2015 2IO70

The purpose of this document is to explain the design of our machine, how we decided on this design and why we decided on this design. To do this we will take a look at our requirements and priorities. Afterwards we will look at the design and the decisions leading to that design

Group 16

Rolf

Wigger

Stefan

Version 1.1

Table of Contents

High level Specification....3
System Level requirements....4
- USE-cases....4
  - Sort unsorted disks....4
  - Abort the process....4
  - Starting the machine....5
  - Shutting down the machine....5
  - Stop the machine....6
  - Booting of the machine....6
- User Constraints....7
- Safety Properties....7

Design Decisions....8
- The Feeder....8
- The Transportation and Scanning....9
- The sorting mechanism....10

High level Specification

The specification as given in the Technical Guide

The goal of this project is to build a simple sorting machine that is able to separate small objects, plastic discs that may be either black or white, into two sets: the black discs and the white discs. (…) The only real requirement, is that the machine must contain at least one conveyor belt.

Our specification

We have to make a so called sorting machine. This machine should be able to separate, by colour, small black and white plastic discs. The only real requirement in achieving this is that we need to use at least one conveyor belt.

System Level requirements

The system level requirements consist out of 3 parts. These 3 parts are the USE-cases, the safety properties and the user constraints.

USE-cases

There are 6 USE-cases which are described below.

Sort unsorted disks

Primary Actor: Machine operator (student or teacher at Tu/e)

Scope: A sorting machine

Brief:
The machine sorts the unsorted disks provided into two separate containers based on colour.

Postconditions:
- There are no unsorted disks left
- All sorted disks are in a container based on their colour

Preconditions:
The machine is not already running.

Trigger:
The user provides unsorted disks and presses the "START" button.

Basic Flow:
1. The user provides a number of unsorted disks
2. An unsorted disk id moved to the colour detector
3. The machine decides to which of the two containers the disk needs to be moved
4. The machine moves the disk to the designated container
5. The machine repeats step 2 through 4 until all disks have been sorted
6. The machine pauses

Abort the process

Primary Actor: Machine operator (student or teacher at Tu/e)

Scope: A sorting machine

Brief:
The machine should immediately stop doing anything.

Post-condition:
The machine stopped running.

Precondition:
The machine is sorting discs.

Trigger:
The use wants to immediately stop the machine.

Flow:

1. The machine stops transporting the discs. And doesn't put any more discs on the transporting mechanism.
2. The user is required to remove all discs that are neither in the container unit nor sorted.
3. Then the machine has stopped running.

Starting the machine

Primary Actor: Machine operator (student or teacher at Tu/e)

Scope: A sorting machine

Brief:
The machine operator starts the machine, machine parts go to their initials state and the machine starts sorting.

Post conditions:
The machine starts the sorting process.

Preconditions:
The machine is in its initial state.

Trigger:
The user performs an action on the machine.

Basic flow:

1. Machine puts devices in their initials state.
2. Machine starts sorting

Shutting down the machine

Primary Actor: Machine operator (student or teacher at Tu/e)

Scope: A sorting machine

Brief:
User unplugs the power supply and disconnects the processor from the PC and the machine.

Post conditions:
The PC can be used for other things and the processor and machine can be stored separately.

Preconditions:
Everything is in its initial state or the machine has stopped.

Trigger:
N/a

Basic flow:

1. Unplug the power supply of the machine.
2. Unplug the power supply of the processor.
3. Disconnect the processor from the machine.
4. Disconnect the PC from the processor.

Stop the machine

<u>Primary Actor:</u> Machine operator (student or teacher at Tu/e)

<u>Scope:</u> A sorting machine

<u>Brief:</u>
The machine is waiting for the current process to end before it is send into an inactive state .

<u>Post condition:</u>
The machine is sent into an inactive state with no process interrupted.

<u>Precondition:</u>
The machine is running.

<u>Trigger:</u>
The STOP button is pressed.

<u>Basic flow:</u>
1. The machine finishes sorting* the disk currently in the machine
2. The machine enters an inactive state and will not take any more disks form the storage* unless the START button is pressed.

Booting of the machine

<u>Primary Actor:</u> Machine operator (student or teacher at Tu/e)

<u>Scope:</u> A sorting machine

<u>Brief:</u>
The machine will prepared to start the program. And do the required actions.
<u>Post-condition:</u>
The machine is ready to get instructions of the user.

<u>Preconditions:</u>
The machine is off.

<u>Trigger:</u>
N/a

<u>Flow:</u>
1. Connect the PP2-board to the pc.
2. Plug the pp2-board in to the power socket.
3. Start the debugger
4. Connect the pp2-board using the debugger.
5. Load the program into the debugger.
6. Run the program.

User Constraints

- Before the start buttons is pressed the user is required to place all discs to be sorted in the container unit
- While the machine is running the user is not allowed to move the machine or touch anything except the buttons.
- When the abort button is pressed or the machine has been shut down, the user is required to remove all discs that are neither in the container unit nor sorted.

Safety Properties

- After pressing an emergency button, within 50 ms there should be no moving part in the machine
- If all disks are sorted the machine should stop within 5 seconds.
- After the startup of the machine, the assembly program should not stop until the machine is shut down.

Design Decisions

Before we can talk about the design decisions we made, we have to specify what our priorities are. These are our priorities:

1. Reliability of the machine (as reliable as possible)
2. The speed of the sorting (as fast as possible)
3. The robustness of the machine (it should not break very easily)
4. User accessibility (It should not be hard for the user to do the actions required)
5. Lowest amount of space (as little floor space used as possible)
6. Difficulty of building the machine (as low as possible)
7. The amount of parts of the machine (as few as possible)

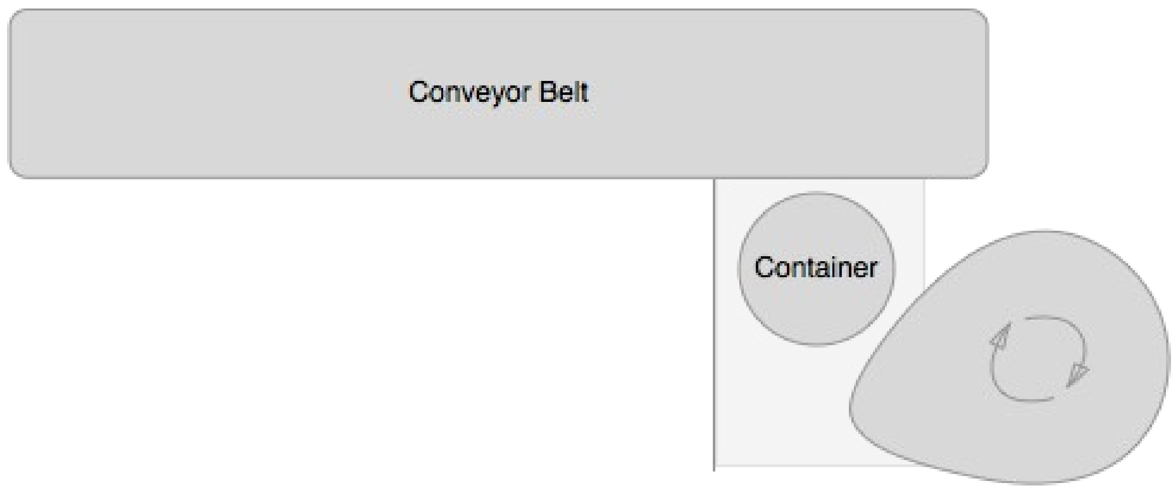
The way we approached the design of the machine is by separating the machine into multiple parts. Those parts exist out of: the feeder, the transportation mechanism, and the sorter.

The Feeder

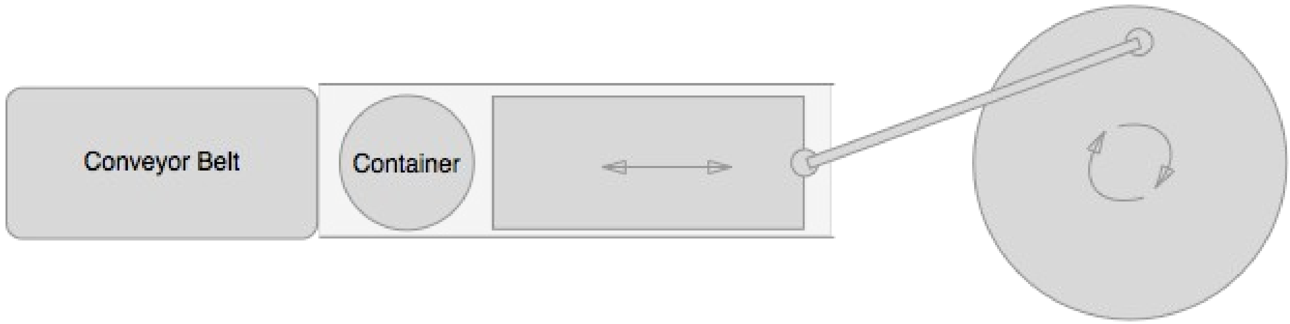
The feeder has as objective that it needs to somehow get the disks from the container onto the conveyor belt. This is needed for the use case "Sort unsorted disks".

For the design of this feeder we had two competing designs. Both use the two hollow tubes stacked as a container. We chose to do this because they are completely reliable in containing the disks and because a new disk simply falls out if the bottom one is removed, they are very fast. Because the container is made off two big parts and some small parts to make them stack, the container is also very robust. It's quite easy to put the disks into the big hole at the top, so user accessibility was very high. In short, the first solution that came to mind scored extremely high on all priorities and we looked no further.

The first design for the feeder consist of 3 important parts. First you have the container. The container drops a disk, which is then pushed onto the conveyor belt using a cam. A wall to the left of the container makes sure the disk is pushed up and not to the left.



Our second design also consisted of a block that pushes the disk. To make this block move a lever attached to a wheel is used. Rotating the wheel makes the block move back and forth, pushing disks onto the conveyor belt.



Both designs correctly implemented the use cases. To test which one would be better we build both and tested them. They scored the same on almost all top priorities. They where both completely reliable for instance. There was also no difference in speed, both would push a disk onto the conveyor belt with every turn of their wheels. Both did not hinder the user, so the good user accessibility of the container was unchanged. When we came to the last three priorities there where some differences making us choose the first design: It was easier to build, used less parts and was a lot more compact.

The Transportation and Scanning

When considering the transportation method we had a 3 main ideas. The first one was that we used a short conveyor belt. The second idea was about a long conveyor belt. And the last idea used a turning wheel and 2 conveyor belts. All these ideas included a conveyor belt because that was required.

The thought behind the short conveyor belt was that in the feeding mechanism would push the discs hard enough so that we could put the sensors on that part and to have a small but conveyor belt to transport the discs. The conveyor was short because nothing needed to happen on it. Thus it would only be there because it was a requirement. To us it seemed a bit useless to not do anything on the conveyors belts. So that was when the second arose.

The second idea had a long conveyor belt to put the sensors on. And also a part of the separating mechanism. The conveyor belt would limit how fast the machine can run but all the actions would happen on the conveyor belt so that time wouldn't be wasted. It also isn't that hard to create a long conveyor belt so we kept the idea in mind.

Finally the idea that there would be some sort of wheel with separate compartments for discs in the centre which would rotate and put discs on to two different conveyor belts. Each conveyor belt led to a storage unit of the sorted discs. The problem with this idea was that it would be hard to prevent the discs from spinning out of the compartments when they shouldn't while still being able to let the discs go out when they had to. Because we couldn't get it to work the idea was dropped and we went back to the idea about a long conveyor belt.

The idea about the long conveyor belt we were capable of realizing it. But during the build of the conveyor belt we noticed that it would not be tight enough around the gears. Thus we tried to remove a small part of the belt. But this still didn't have to effect we hoped for. So we added a third gear in the middle which tightened the belt to an acceptable state.

The conveyor belt was still far from perfect because it would tilt at certain points and the discs could fall off. So to prevent it we build 2 walls around the belt. On the first part they are low because the low walls were more robust than the high walls and for the user it is easier to access the discs on the conveyor belt. The high walls have been secured using 4 pillars because that made it robust enough to make sure they didn't break. The walls had to be high because we needed to put a set of sensors on it.

Those sensor had to be above the conveyor belt. They also needed to be at an angle to work properly. That was required else the sensor wouldn't be able to check if the disc was black or white.

The other set of sensors didn't need to be place at an angle thus they were simply put on each side of the conveyor belt. This set of sensor would then be capable to scan if there was a disc on that spot of the conveyor belt. This sensor is need to time at which moment the other set of sensor had to check the colour of the disc. And it is also used to check if there are any more discs left to scan.

The sorting mechanism

For the mechanism that does the actual sorting we chose between a number of different designs. These designs are listed and explained below.

The first, and most simple design was to use just one conveyor belt that would move left or right based on the colour of the disks. This design is listed under the use of the conveyor belt above, this is why I will not describe it again.

The second design is a slight improvement on the first one where we would use a second, shorter, conveyor belt to do the sorting. This design would place the two conveyor belts in a T-shape with the colour check done on the first one, after which the second conveyor belt moves left or right. We considered this design an improvement on the first one because the second conveyor belt could be made much shorter. This means that the design can sort faster than the single conveyor belt one.

The second conveyor belt was faster than the first design with only one belt, however we soon realized that we could do this even faster. By removing the second belt and replacing it with a seesaw that could be angled to face one of the two sorted containers, we could increase the speed even more. Since the disk would essentially be sorted the moment it reached the end of the conveyor belt. This would be a great design, was it not for the fact that the seesaw required a lot of height. In fact, the entire machine looked like it was placed on stilts, requiring us to use lots of parts and having a lot of wasted space underneath. This design could do it faster at the cost of requiring more space than any of the others.

While the use of a seesaw sped up the sorting process, it also took a lot more space, so we went back to the drawing board and discarded this idea. Instead coming up with a wedge that would be slide onto the conveyor belt from the side whenever a disk of a certain colour is detected. This would then allow the conveyor belt to push the disk against the wedge making a roughly 45° angle thus pushing the disk of the side of the belt and into the collection box. The second colour could just continue while the wedge was pulled back and off the end of the belt. This meas that the design cuts off part of the machine at the end and allowing us to make the machine lower than before.

We liked the idea of letting the conveyor belt doing the sorting by placing a wedge in the way, but after some thinking we realized that it could be done both faster and more compact. The trick was to change the direction in the wedge moves from horizontal to vertical. Doing so moves the entire mechanism, aside from the wedge itself, in an upright position pusing it very close to the machine. Aside from savind space, this also allowed the wedge to move much less, since it only has to move just over 1cm above the conveyor belt rather than move all the way over it to the side. This final design does not sacrifice any reliability from its predicessors while being the fastest. It also takes by far the lowest amount of floor space, charactarized by the fact that this final design including this sorting mechanism is our only design that fits on only one of the two provided floor plates. For these reasons we believe this design for the sorting mechanism to be the best.