The Limpio

Proposed solution

Note: This report is intentionally kept short; the here proposed solution is only worked out to a rough, conceptual level, as we felt that the main goal was to provide a realisable idea more than a detailed design. Simple “back-of-the-envelope” calculations are provided when necessary for design decisions.

The proposed method to clean the awnings is to use a non-contact method, more specifically a high-pressure waterjet that cleans the awnings from above. This offers the advantage to be easy to implement and allows to avoid most of the small-scale obstacles present on the awning, such as transversal beams, … The waterjet is directed by a 3-link, 5 DOF robotic arm. The third link is horizontally suspended above the awning and spans about half the width of the awning. It contains nozzles along its length providing the cleaning waterjets. To move along the platform, this arm is mounted on a wagon base that is self-propelled or actuated by an external locomotive. Figure 1 gives an overview of the mechanism, and later on is referred to a set of visualisations showing the system in action in a virtual environment. The name Limpio comes from Spanish and means ‘Clean’

This approach offers several advantages: it is very flexible and usable in a number of different scenarios. For example, the arm could also be used to clean the platform floor or vertical glass surfaces near the tracks. Moreover, as the mechanism is on rails, it is easy to transport in and out of storage and to other railway stations, where it can perform the same tasks. Mounting the mechanism on a wagon base also allows to easily add the heavy equipment necessary for cleaning, such as pumps and water reservoirs, as wagons have a large amount of available space and can carry a large weight. By designing the system this way, a large modularity can be achieved since extra wagons can be added later on, and even the whole cleaning system can be replaced by another end-effector.

The main disadvantage of this method is that it requires closing tracks adjacent to the awning that has to be cleaned. This can however easily be remedied by scheduling the cleaning procedure at night times, when there is no other traffic on the lines. It is also a pricy solution if we look only at the possibilities to clean the awnings in Leuven station perron A, B and C. But our system is designed to be able to deal with all awnings in Leuven station and can be used on virtually every station if the software is updated. Also the platforms can be cleaned and even walls. The possibilities are nearly limitless.

Design

This section briefly discusses the main design choices. The 5 DOF robotic arm is very large due to the nature of the application, and needs to provide cleaning flexibility while also avoiding collisions with the overhead cables. It has been chosen to not have it custom-made.

Figure 1 provides an overview of the arm, mounted on the wagon. The links are referred to by numbers 1, 2 and 3, the joints by letters *A*, *B* and *C*. Angles between the links are referred to as θ_1 and θ_2. The arm can be swivelled side to side by an angle ψ by the turning platform on which the arm is mounted.

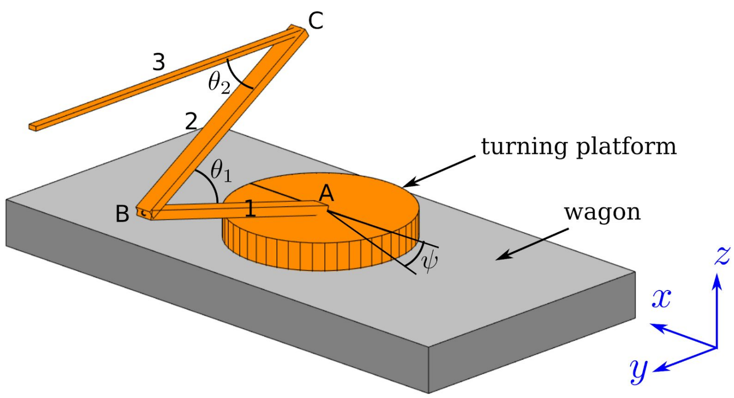

Figure 1: Overview of the robotic cleaning arm on the wagon

The links are made of rectangular steel S235JR profiles to allow space for the different cables and pipes inside while also providing sufficient strength and stiffness. Due to the need for high torques and accuracy, high torque servo motors are chosen to actuate the joints in *A*, *B* and *C*. The chosen motors are ATO180SY-M17015-H, at a price (including encoders) at around 800€ a piece. For safety reasons these motors also contain passive breaks such that any sudden drop in power will result in the arm staying at its current position. This allows to fix the choice of the profiles to NEN EN 10210-2 200x100 profiles for links 1 and 2, and a NEN EN 10210-2 100x50 profile for link 3. The lengths are chosen to provide enough operational space for the cleaning. Links 1 and 2 both have lengths of 2.63 m, while the third link is taken at 4 m, to span at least half of the widest awning available in Leuven station (which has a width of 7.76 m). Joints *A* and *B* both have 1 DOF, while joint *C* has 3 DOF to assure correct cleaning orientation in all circumstances.

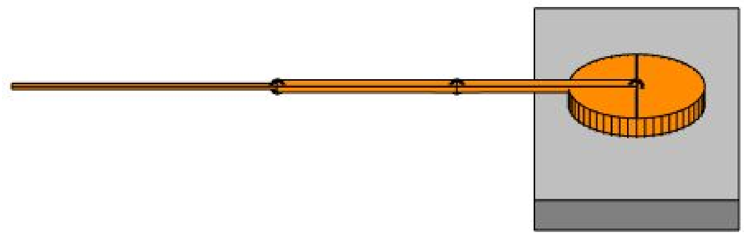

Figure 2: Cleaning arm in fully extended configuration

Strength and stiffness calculations are made for the “worst case” scenario of a fully extended arm as shown in figure 2. This is an unrealistic scenario that will never occur in reality, but is useful to determine upper bounds on necessary stiffness and strength. Details are omitted for the sake of brevity; it is sufficient to note that:

- All links are within strength limits in both bending and shear, which for steel S235JR are about $\bar{\sigma_b} = 218$ MPa and $\bar{\tau_b} = 126$ MPa, as given by [1]. Torsion effects are negligible in this case.
- The deflection of each link due to its finite stiffness is around 0.03 m, and deflections of the two first links are easily be compensated using feedback control on the position of link 3, meaning only the deflection at the end of link 3 is of any significance.
- In nominal operation, the water jets create forces that counteract the torques needed by the arm to support itself, creating a reduction in applied torque and a reduced deflection of link 3.
- Vibrational effects are taken into account using a safety factor of about 1,5 in the calculations, according to [1].

To maintain the arm at the desired position, sensors are placed on link 3 at the positions indicated by figure 7. A possibility would be distance sensors from the ‘OD-value’ range of SICK, which offer enough range at a reasonable cost. Joints *A*, *B* and *C* further contain built-in encoders to allow feedback control on the angles.
For a given position of link 3, the necessary angles θ_1, θ_2 and ψ can easily be computed by solving the loop equations using a nonlinear equation solver using e.g. a Levenberg-Marquardt method and a reasonable begin value*[1]. This is the method employed to create the matlab simulations and visualisations referred to in a later section. These values can then be used as setpoints for the internal controllers.

The wagon on which the arm is mounted is not discussed in much detail; it is sufficient to note that it will be moved by a locomotive at a low constant velocity of around 0.05 m/s during operation. When the arm needs to stay in a *x*-position a bit longer for e.g. obstacle avoidance, the angle ψ will be used to compensate for the moving wagon.

---

[1] Note that the initial values need only be ‘guessed’ in the first step; subsequent steps can use the previous solution as initial guess as the arm will have moved little between computing cycles.

Water Cleaning System

For the cleaning of the surfaces we uses high pressure water jets. What now follows is a rough calculation on the required water for the cleaning. As a rule of thumb we used the water calculations used by Karcher. For every square meter they use 10l water for cleaning, and the water consumption of one cleaner is around 600l/h. We decided to use 12 jets in parallel for the cleaning, to cover the full distance of 4 meter. This would mean that the water usage per hour is 7200l and there is an ability to clean 720 m². We also are able to calculate the speed of the wagon from this, as the part to be cleaned is around 4 m wide, we can move 180 meters in an hour, resulting in a speed of 5 cm/s. The usual cleaning pressure for heavy duty is around 200 bars, so to have slack we propose to use water pumps of 300 bars. As an example, the interpump 66 series has the W3025 pump, with a water supply of 1500L/hr and maximal pressure of 300 bars. By using 5 of them in parallel, the requested amount of water could be delivered. They are in the price range of 800€, giving a total of 4000 € in pumps. Waterjets can be designed self, and if no valves are used, the pumps will compose the main cost of the water system. This would also require a total power of 73,5 kW, but this is easily available with the supply lines of the track. Water itself could be attached as a tank wagon to the system. There are tank wagons with a capacity of 27.000 l, which could possibly give a cleaning of 2700 square meters. If this would prove insufficient, an extra wagon could be attached, again showing the power in the modularity of the system.

Cleaning procedure

The cleaning procedure itself will depend heavily on position control. We won't be using predefined trajectories, as this would require a precise control of the locomotive itself which is nearly impossible. Instead we will isolate the robot arm, and let it operate in its own frame. Since we decouple the speed of the locomotive, this can be held constant, and this also means that the robot arm frame doesn't experience any accelerations. We add a seperate control loop for the third arm, where it's only function is to keep it horizontal, so we don't have to worry about it in the future. This way we can make an FSM for the whole motion, and from this broader FSM we subdivide it in smaller FSM's to do each task.

The idea is to work with four phases. The first is just the Rest state, from where the robot starts and returns to at the end of the job or for transportation. The second one is the deployment state, where the arm will be stretched out till it reaches its beginning position.Third state is the actual cleaning state, where a constant distance is held from the surface and possible obstacles are avoided. When moving between to awnings, one can go again to a deployment state, but if a whole platform is completed, we go to a resetting phase where the arm is retracted and the whole system goes to rest phase. This can happen when we switch platforms, which can easily be done by driving till hitting a track change and going back afterwards. Of Course, if the job is done, we also go back to the resting phase. This results in the following FSM. As this is just the main task division, no transition requirements are added.

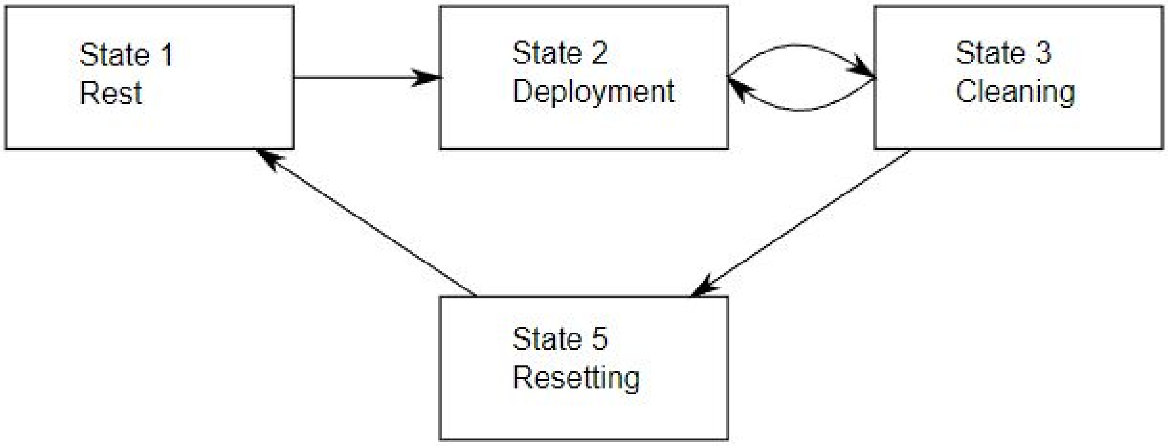

Figure 3. FSM of the whole cycle

In the text a lot of reference frames are used. The different frames are visualised that are used in the text. With a robot frame, trajectory frame and end-effector frame respectively.

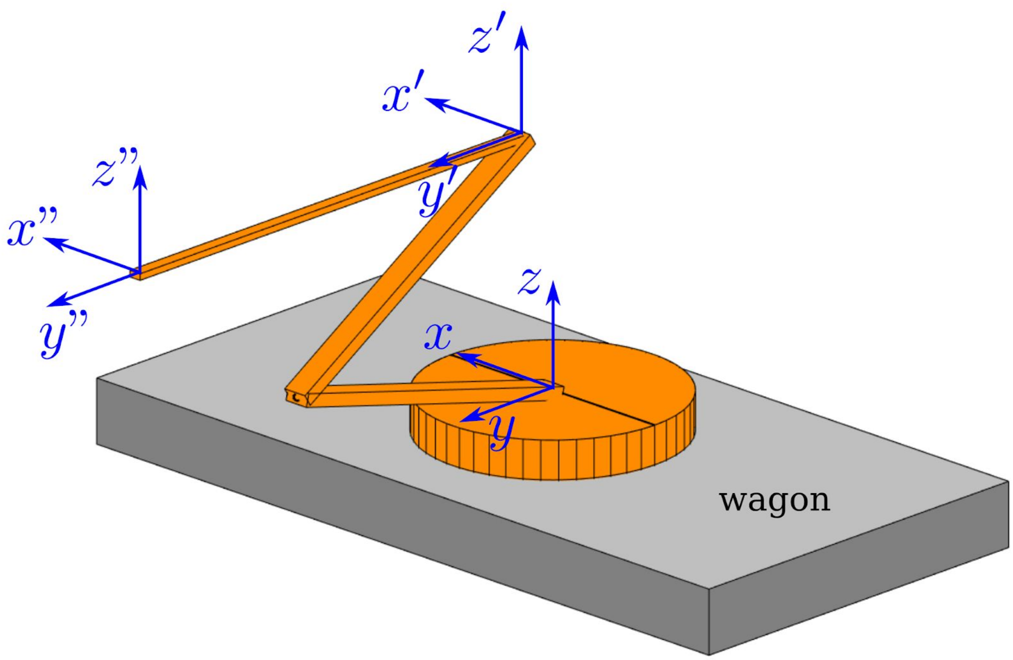

Figure 4. Reference frames used in the control system

1. Starting position
   The Starting position is when the robot is at rest. With a camera, the first awning can be located, and also the distance. When to robot reaches the distance before the awning where deployment has to start a transition happens to state 2. This distance is chosen in such way that it is long enough to complete the full state 2.
2. Deployment
   In this second state, the arm is deployed. This happens simply by turning the robot platform 90° or -90° depending on which side cleaning will happen. Next the second arm of the robot will be actuated and brought into the angle that is necessary to get to the setpoint defined in stage 1 denoted as angle 1 and 2. Next on, the first arm of the robot is brought into place. This way we can guarantee that the robot arm never comes to safe to the track lines, ensuring safety. This results in the following task FSM, where explicit monitoring is added. The values come from encoders on the joints. Note that at termination this subtask is ready, but moving onto cleaning happens when the edge of the awning is reached

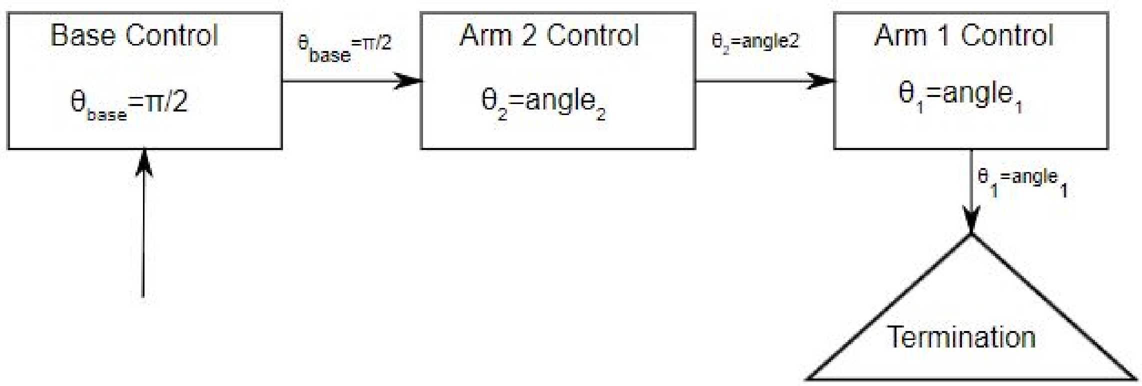

Figure 5. Deployment FSM

3. Cleaning
   In this state we start-off with turning on the pressure cleaners. This will not be explicitly mentioned, since it will just be on the whole time. Because it will be used often, figure 8. Shows the sensors used in the control scheme. The +x sensors are in the purple box, the -x sensors the green, and thy y-sensor on top. The little red squares facing down are the z-sensors. This is actually a bottom view without the jets.The main part is the following of the awning and keeping a set distance. We choose as distances from objects to be 20 cm, and the distance for cleaning to be 40 cm. These are just chosen values, they can be optimized later. The following FSM is derived to avoid collisions. In the nominal state, we just keep the distance of 40 cm while keeping x value on zero. This x is in the robot-arm reference frame, which means that the base is kept at 90°. We also have a control for avoiding collisions in the x direction and to avoid collision with

pillars. Here is the resulting FSM which will be elaborated below.

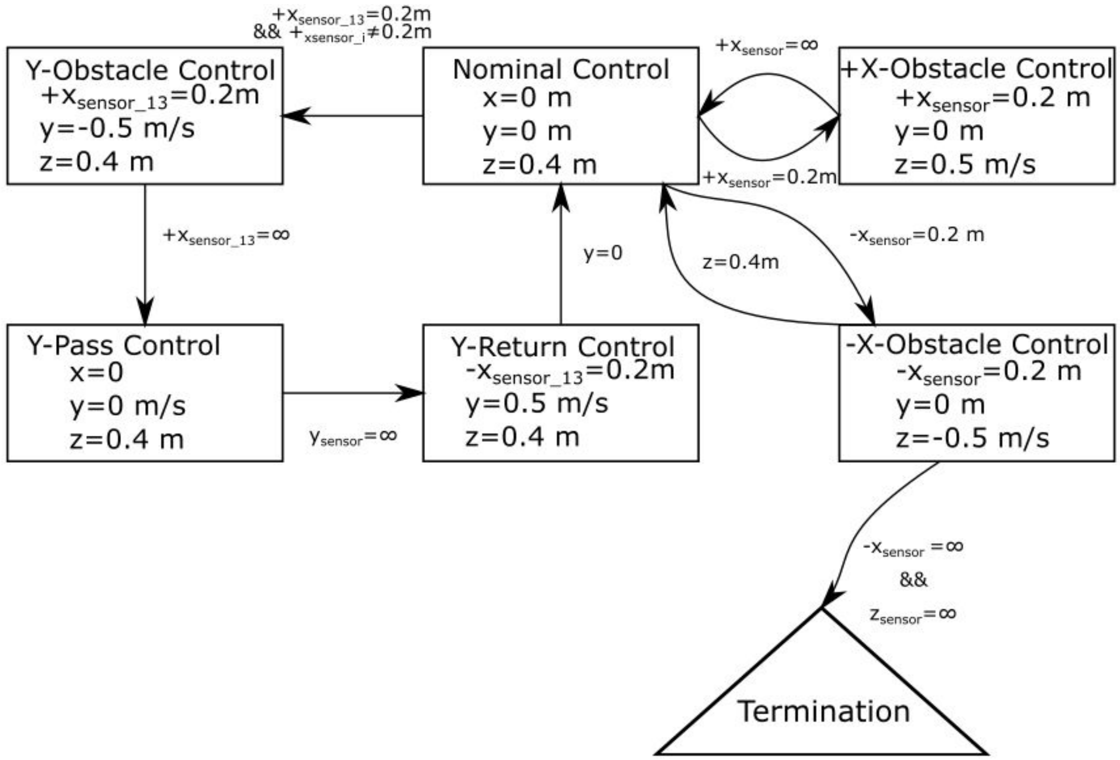

Figure 6. Cleaning FSM

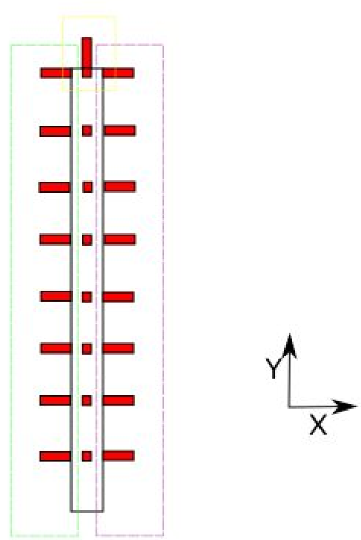

Figure 7. Sensor placement on third arm

Y-obstacle control is triggered when the last distance-sensor in the x direction is triggered, but the other ones not, indicating there is a pillar ahead. By keeping the distance that the x-sensor measures 20 cm, but letting the y value change the collision is avoided. Since we have no idea till when we have to execute this, a speed is used with as transition that the x-sensor saturates and no more obstacle is in front of him. After this, we make a transition phase. The reason we don't go back to nominal control, is that after passing the object, the y-sensor on the tip of the arm will saturate, and in nominal control it is always saturated, so we couldn't transition to the Y-return. The Y-return is just a mirroring of the Y-obstacle control, where we now keep the distance at the back at 20 cm. Next on, if there is a obstacle ahead which can be passed above, +x obstacle control will start, keeping the distance at 20 cm and increasing height, until the x-sensor in the front saturates, and we go back to nominal control. When we approach a downward edge, nominal control will still be working, pushing the arm down since it reads out that the distance is to big. But when the x-sensors at the back are triggered that the distance is 20 cm, the arm will follow that edge down, until the condition is satisfied that the z-sensors at the bottom read out 40 cm, and we go back to nominal control. Termination occurs when during the minus x control the end of the awning is detected. This is done by simultaneously have that the z-sensor saturates and the minus x sensor saturates. At this point the cleaning jets are turned off.

Here are two movies from the cleaning in action. We made a matlab simulation with a predefined environment and coded virtual sensors executing the FSM we had in mind.

> https://www.youtube.com/watch?v=xSd9h6LTvYs&fbclid=IwAR2PfrA1-wOiHvzH8Se2rYVBw-KK89y1EolyEUCJj-3c5aXhXs9Q3K4Sc34
>
> https://www.youtube.com/watch?v=KNJSaF9vOgM&fbclid=IwAR2ARVBCgfLjbKJevnYX7dYjA18UH41GaHw2hRhzQ2p_37Kp1O-i9wYipyo

4. Resetting
   The resetting phase is actually just a mirroring of the deployment phase so no much explanation is needed. For the sake of clarity here is the final task fsm

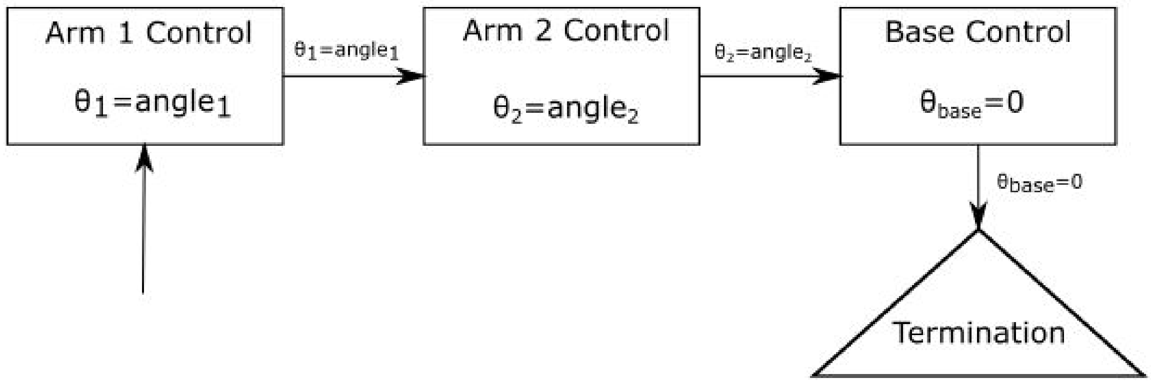

Figure 8. Resetting FSM

Cost estimation
With the hydraulic motors being around 4000 € and the motors being 1200€ the main cost will be the laser sensors. The laser sensors don't have to be very precise, and you could get a low-spec distance laser for around 100€ a piece. That being sad, in the current design 40 laser sensors are used, which adds up to 4000 €. Also every motor will need a seperate controller and a controller costs around 500€ a piece depending on platform. If C++ is used as operating language, the prices will be lower. Also a tankwagon can be bought second hand for around 1000€. All this adds up to 11700€ just in parts that have to be bought, but total costs will be a lot more so 25000€ in parts is more realistic, if we don't count for the parts that NMBS-SNCB will have to deliver itself, which is the motor/locomotive, the power-electronics,the cleaning jets,... but those are things that NMBS-SNCB can more readily give an answer to.

References

[1] Wittel, H., Muhs, D., Jannasch, D., and Vossiek, J. (2013). *Roloff/Matek Machineonderdelen, Tabellenboek,* Den Haag: Sdu Uitgevers.