Autonomous Parcel Delivery at UT

1st Lars Klunder
University of Twente, EEMCS
Enschede, The Netherlands
l.a.klunder@student.utwente.nl

2nd Martijn Sikking
University of Twente, EEMCS
Enschede, The Netherlands
m.d.sikking@student.utwente.nl

3rd Mark Huizinga
University of Twente, IEBIS
Enschede, The Netherlands
m.j.huizinga@student.utwente.nl

***Abstract*—This study evaluates the feasibility and efficiency of implementing autonomous last-mile delivery systems on the University of Twente campus as a sustainable alternative to traditional delivery methods. The experiments identified an optimal fleet configuration consisting of 3 Mobile Parcel Lockers, 8 Street Robots, 1 Quadcopter with Hoist, and 3 Quadcopters without Hoist. This configuration is determined by looking at various KPIs with a people, planet and profit perspective. This setup achieved a delivery success rate of 92%. The simulation also demonstrated that the time window duration could be reduced by 19% without compromising service reliability, leading to improved efficiency and enhanced delivery performance. The results provide actionable insights for Campus & Facility Management to implement a scalable, efficient, and environmentally friendly parcel delivery solution on campus. The study recommends adopting this configuration and emphasizes the importance of continuous stress testing and iterative updates to the simulation model to adapt to growing parcel demands and environmental factors.**

I. Introduction

The University of Twente is exploring Autonomous Parcel Delivery as a last-mile delivery alternative to traditional third-party delivery services. These parties often use combustion or electric vans that cause disruptions in the environment. This study aims to simulate and evaluate the viability of a autonomous parcel delivery system using discrete event simulation (DES).

Building on a pre-established base model, we conduct research using a traditional benchmark delivery model and a decoupling model that involves the autonomous vehicles. To create a working decoupling model a parcel-scheduling algorithm is build, delivery time windows are analyzed and adhered. We look at fleet optimization and assess these with KPIs that are deemed important. By performing experiments and an iterative algorithm we seek to find the best configuration of the vehicles used that perform well within set time windows.

II. Research Design

A discrete event simulation will be build using Siemens' Plant Simulation software. We start with the base model that is provided. The model will be developed to integrate key components like dispatching and scheduling logic, fleet configuration and experiment optimization.

A. Project objectives

This research aims to provide recommendations to the Campus & Facility Management (CFM) on a feasible, efficient and sustainable design and implementation of autonomous last-mile logistic options. The study looks at key performance indicators (KPIs) to evaluate and compare outcomes of the model using different configurations of the vehicles.

B. Scope

We focus exclusively on last-mile parcel delivery at the UT campus. We consider the autonomous vehicles only as a replacement for the deliveries on the campus. We only have simulate two delivery scenarios.

1) Benchmark Scenario: This represents the traditional way of delivery where a van drives around the campus in a one-by-one fashion.
2) Decoupling Scenario: A normal delivery van drops all parcels at a central point (Decoupling Point) and last-mile delivery is done by autonomous vehicles.

III. Model description

A. System description

Campus layout

The simulation operates in a virtual map of the UT Campus. Each delivery possibility, like a house or dorm, is mapped on the specific location and has a delivery marker. Not every destination has a marker to keep a clear model. The vehicles stop at the closest marker to the destination and moves from marker to marker. The system uses "Campus Frame" to represent the entire campus. On the campus frame there is a "Decoupling Point" that acts as the central hub. There are flowcharts located in the Appendix C that describe the system from a discrete event simulation point of view. The system is also described in the conceptual model.

Delivery Vehicles

For the last-mile delivery there are different options for what vehicle can be used. These vehicles are, a quad copter with and without hoist, a streetrobot, and, a mobile parcel locker. These vehicles are autonomous

and are powered by a battery. All these vehicles have their own specification and (dis-)advantages when used for last-mile delivery. The quad copters can be used to deliver just one parcel at a time and have a relative high speed. The drones do make a lot of noise which can be experienced as annoying by people in the area. The quad copter with hoist does not need anyone to accept the parcel, which is unique. The streetrobot has the capacity of delivering 1 parcel at a time and delivers the parcels relatively slow. However, they do not cause any annoyance, at most some traffic hindrance. The mobile parcel locker behaves like a streetrobot, except it has a larger capacity of 24 parcels.

Parcels

Parcels are created and a time window and destination are allocated to each parcel. The parcels are allocated to a vehicle and brought to their destination. The way the parcels are sorted determines the route or order by which they are delivered. In the benchmark situation all parcels are delivered by the delivery van. In the decoupling scenario the parcels are brought to the decoupling point and from there the parcels are delivered with the last-mile autonomous vehicles.

Unloadtimes destination

Provided with the project was a list of delivery times for the benchmark situation. This data is used to fit a probability distribution to use as input for the delivery times of the different vehicles. Using the Data Analysis tool in Excel a summary statistics is made from the data and based on this a few distributions are more likely than others. A Gamma distribution is possible, so we use estimators for the alpha and the beta values with the mean μ and the sample variance. Based on the number of datapoints n and the maximum and minimum values from the empirical data, we use the following binsize:

$$\alpha = \frac{\mu^2}{S^2}, \beta = \frac{S^2}{\mu} BinSize = \frac{\sqrt{n}}{MaxNr - MinNr}$$

By using the data analysis and the histogram function we calculate the frequency of datapoints per binsize and divide this by the number of datapoints to find the fraction. We use this data to make a histogram and put the theoretical gamma distribution as a fluent line. The distributions seem to be similar or have the same shape. However, more tests can be done to make sure the empirical data can be fitted to the gamma distribution. This is done by making a QQ-plot of the empirical data and the gamma distribution. By making this plot we see the data on a diagonal line of about 45 degrees. This indicates the distribution is fitted properly. Another test that can be done is the Chi-squared goodness-of-fit test. The squared sum of the difference between the empirical distribution and the theoretical distribution. Based on this test statistic and the others we can conclude the gamma distribution with the alpha en beta.

$$\chi^2 = \sum_{i=1}^{n} \frac{(O_i - E_i)^2}{E_i}$$

- χ^2: The Chi-Squared statistic
- O_i: Observed frequency in bin i
- E_i: Expected frequency in bin i
- n: Number of bins or categories

B. Performance Metrics

The decoupling situation needs to be assessed, and in order to do experiments and compare the model with different configurations we designed KPIs. These KPIs are from a People, Planet and Profit perspective to provide a well-rounded analysis of the system, regarding the campus environment and make conclusions and recommendations for CFM.

1) People

On-Time Delivery Rate

Important is the time in system for a parcel. This indicates how long it takes per parcel to be delivered. Taking the average reflects how efficient the system works and whether the parcels are delivered in a timely manner.

On time delivery rate:

$$ParcelsDelivered = \frac{ParcelsDelivered}{ParcelsDelivered + ParcelsNotDelivered} \quad (1)$$

This is the ratio that shows whether the system is operating proficiently. If the ratio is very low it indicates a lot of parcels are not delivered and something needs to be adjusted with regards to the configuration of the fleet. This KPI needs to be as high as possible, when parcels are not delivered they are moved to be delivered the next day.

Parcels not delivered

This KPI can be retrieved directly from the model and this KPI is important since every parcel that is not delivered has costs connected to it, meaning it is important to keep this KPI low.

Noise pollution

The use of vehicles for last-mile delivery makes noise. Especially, when the parcels are delivered with drones. Per vehicle the noise in decibel is provided in a table.

However, the measurement decibel is hard to add together since that would result in an unfair representation of the noise actually produced. For example, the use of three streetrobots that produce a sound with 50 decibel is not the same as the noise of the diesel delivery van and a quadcopter with hoist. This is why there are weights per km assigned to the decibel level.

Weight	Noise(dB)
0	$W \leq 50$
1	$51 \leq W \leq 64$
2	$65 \leq W \leq 74$
3	$W \geq 75$

TABLE I: Weights of dB level

Vehicle	Noise(dB)(W)
Diesel delivery van	70(2)
Electric delivery van	65(2)
QuadcopterWithHoist	80(3)
StreetRobot	50(0)
MobileParcelLocker	60(1)
QuadcopterNoHoist	75(3)

TABLE II: dB per vehicle

2) Planet

Energy consumption per vehicle

It is important to know how much energy the autonomous vehicles use, considering environmental impact, cost-effectiveness, and operational efficiency. This KPI is calculated for every vehicle separately to see how all vehicles perform and a total is calculated to see the overall use of energy for a certain scenario and configuration of the vehicles. This energy consumption per vehicle is used to calculate the cost and emissions for KPIs discussed later in this section.

Vehicle	kWh/km
Electric delivery van	0,28
QuadcopterWithHoist	0,4
StreetRobot	0,08
MobileParcelLocker	0,25
QuadcopterNoHoist	0,3

TABLE III: Energy use per vehicle per km

CO_2 emissions

To see how much emission a vehicle has is very important to track and also what the total emissions are of the configuration. The university strives to be as green as possible and that is also why all electricity used is "green". Since 2022, certificates are bought to mitigate the emissions of electricity at the university. These certificates do cost money, so with fewer emissions the university can save money. To be able to compare the benchmark scenario and different configurations of the model an emission of 210 CO_2 g/kWh is used [2] [3].

Vehicle	CO_2/km
Diesel delivery van	147
Electric delivery van	58,8
QuadcopterWithHoist	84
StreetRobot	15,75
MobileParcelLocker	52,5
QuadcopterNoHoist	63

TABLE IV: CO_2 emission per vehicle

Vehicles idle time

To see how well the system behaves the idle time per vehicle needs to be determined. This KPI indicates how busy the vehicles are. This is important to see per vehicle, but also in total. If the KPI is close to 1 it is probably a good idea to add some vehicles to relieve the relieve the system. If it is low it can be an idea to remove a vehicle to save costs. First, the idle time per vehicle is calculated, added together and divided by the total vehicles.

$$IdleTime = (1 - WorkingPortion) \tag{2}$$

3) Profit

This KPI is used to determine what situation is more profitable than the other. This KPI consists of three components. Per vehicle the maintenance and other costs per km, the energy costs [1] per km and the costs of not delivering a parcel. Initial investment costs of the vehicles are not considered, since that would be to much of an estimate and not quite in the scope of this research. After running the model with a certain configuration of the vehicles a total costs is determined and this can be used to evaluate and compare scenarios and make recommendations. Per undelivered parcel the cost is 15 euros.

Vehicle	Euro/km
Diesel delivery van	0,15
Electric delivery van	0,12
QuadcopterWithHoist	0,11
StreetRobot	0,04
MobileParcelLocker	0,1
QuadcopterNoHoist	0,10

TABLE V: Maintenance and other costs per vehicle per km

Vehicle	Euro/km
Diesel delivery van	0,17
Electric delivery van	0,07
QuadcopterWithHoist	0,11
StreetRobot	0,02
MobileParcelLocker	0,07
QuadcopterNoHoist	0,08

TABLE VI: Energy costs per vehicle per km

Cost per parcel
To have an even better overview of the system the total costs are also divided by the number of delivered parcels.

C. Conceptual model

The conceptual model outlines the flow and processes that are underlying the simulations. In this case it means how parcels are managed, routed, delivered in different scenarios. These flowcharts are the same for both the decoupling scenario as the benchmark scenario. The flowcharts are located in Appendix C.

The conceptual model consists out of:

- System Creation and Delivery Method
- Parcel Delivery Process
- Vehicle Arrival Process

System Creation and Delivery Method
The simulations starts with initializing and resetting the system to clear previous data. It then creates delivery pools. Once parcels are created, they get time windows, which specify when a parcel must be delivered and a destination. The simulation then checks for the delivery method. If Benchmark Scenario is set on "True", an external delivery van will handle all the deliveries. If set on "False" it will be the Decoupling Scenario which brings all parcels to a central point from where the autonomous vehicles will do the final last-mile delivery.

Parcel Delivery Process
The delivery process is activated by 2 events, when a parcel arrives at a destination and when a parcel window opens. First, a few checks need to be performed to see how and if the parcel can be delivered. If a person is not needed at the drop-off we check if the parcel is in the fraction of failed deliveries, if this is not the case we deliver the parcel. Then, a parcel might need to be delivered within a time window and if the benchmark scenario is used if the delivery van is too early it waits until the time-window is open, otherwise the delivery cannot be made. This is a discrete event simulation, so this is represented in the flowchart by ending the event when the vehicle needs to wait.

Vehicle Arrival Process
This event is triggered by 2 instances. When a vehicle arrives at a destination and when a vehicle is empty. Then, a few checks need to be made to find out what needs to happen next. If the last marker is not reached, which is the entrance marker of the decoupling point, the vehicle moves to the next destination. Otherwise, a check is made whether it is a delivery van and if the vehicle is empty. If the delivery van is not empty, it needs to unload the parcels, else the event ends. If it is not a delivery van, but an autonomous vehicle, and the vehicle is not empty, it also needs to unload. If an autonomous vehicle is empty, it needs to move to the home location.

D. Assumptions

To be able to simulate the real-world, a few assumptions need to be made. Weather effects are ignored, and it is assumed that all autonomous vehicles, including drones, can operate under all weather conditions. All vehicles are assumed to operate without any mechanical or technical failures, except empty batteries. Routes are assumed to be direct and optimized, with no deviations or obstacles along the way. The vehicles move in straight lines to the markers, with a decreasing factor of the movement speed of the vehicle. The number of people across different locations on campus is assumed to be uniformly distributed. Parcels are assumed to arrive in batches rather than continuously, to simplify scheduling and routing. All parcels are decoupled at the decoupling point for final delivery by autonomous vehicles, with no direct delivery from the van to destinations. Campus residents are assumed to exhibit parcel delivery patterns and behaviors that resemble the average Dutch online consumer. If a parcel's time window is already closed when the vehicle arrives, it is assumed that the delivery fails. We assume that 5% of the deliveries fail.

E. Dispatching rule

To make sure that delivering parcels is done as efficient as possible, a dispatching algorithm was created. This section describes the logic the algorithm follows and the reasoning behind it. To start, a distinction is made between vehicles that can handle multiple parcels (MobileParcelLocker) and vehicles that can only handle one parcel at a time (StreetRobot, QuadcopterNoHoist, QuadcopterWithHoist).
At first, vehicles that can handle multiple parcels are loaded. This is done because a route has to be created which is both fast and allows the vehicle to deliver all parcels within the time window. Thus, it is the best choice to have as many parcels available to create the route from. For each parcel available in the decoupling point, the time it takes for the vehicle to reach the parcel's destination is calculated. This is done by dividing the distance between the decoupling point and the destination with the vehicle's speed. After this, the estimated time of arrival is compared to a parcel's time window. If the ETA is within the time window, the parcel is a viable option to be included in the vehicle's route. If there are multiple viable parcels, the parcel with the shortest travel time is chosen as next parcel to be included in the route. If a parcel gets added to the route, the beginning point for the next iteration is set to

the destination of the last parcel which is added to the route. The time which it takes the vehicle to reach the destination from the previous destination, along with the loading time and drop off time, is added to the current time. For the next iteration, this time will be used as the time at which the vehicle leaves the last destination.

Determining if a parcel can be delivered by a vehicle in time is done in the following way. The method which contains the decoupling algorithm (ScheduleParcels) calls another method (TimingPossible) for each possible parcel that can be assigned to the vehicle. This method calculates the time it will take for the vehicle to reach the destination of the parcel from its current point (this is either the decoupling point or the destination of the last parcel in the route). This time is calculated by taking the speed of the vehicle and multiplying it by the distance the vehicle needs to cover. Next to that, the time it takes to deliver a parcel when the vehicle is at the location is calculated. This time includes the time it takes to load the parcel from the decoupling point to the vehicle, the average unload time for the vehicle, and a 3-minute buffer. This buffer is there since the unload time is determined by a distribution and is not a fixed time. Once the time it takes the vehicle to move to the destination and the time it takes to drop off the parcel is determined, another calculation is done to see if the parcel can be delivered within the time window. The rule is set that a vehicle can deliver a parcel in time if the current time (or, in the case of a vehicle with a capacity greater than 0, the time at which the last parcel in the route is expected to be delivered) + the time it takes to travel to the destination is larger than the start time of the time window. Next to this, the time + the travel time + the time to deliver the parcel (loading time + unload time + buffer) must be smaller than the end time of the time window.

The process of appending parcels to the route of the vehicle is repeated until either no parcels can be added to the route (because the parcel can not be delivered within its time window if it were included in the route) or the vehicle is full. The vehicle will be sent away to deliver parcels if it can deliver at least 10 parcels. If less than 10 parcels can be delivered, sending away the vehicle is deemed not worth it. In this case, the vehicle must often wait until a new batch of deliveries arrives, after which (most of the time) a route can be created which is long enough.

After the vehicles that can handle multiple parcels is either sent away, or the choice was made that the vehicle should not execute the calculated route, the dispatching algorithm moves on to the vehicles that can only handle one parcel.

The most important function for these vehicles is that they should deliver parcels that need to be delivered fast. As these vehicles only deliver one parcel, they can go there in a straight line and don't have to make detours to deliver other parcels. Next to that, the quadcopters have high speed, which allows them to be able to reach a destination quickly.

At first, all parcels at the decoupling point which are still available are sorted based on when their time window ends. Next, the algorithm checks if any of the parcels can be delivered by a quadcopter with hoist. As these types of quadcopters can not transport all parcels, they are dispatched first. This is to prevent the scenario where another vehicle is transporting a parcel suitable for the quadcopter with hoist, while there are no other parcels available for the quadcopter with hoist, resulting in the quadcopter staying idle in the decoupling point. For each quadcopter with hoist, the parcel with the closest end time to the current time is assigned to it, given that the parcel is suitable for the quadcopter with hoist and that by the time the quadcopter reaches the parcel's destination, the parcel's time window has started.

After the quadcopters with hoist are dealt with, the other vehicles with single capacity (Quadcopters without hoist & Streetrobots) get parcels assigned to them. These vehicles will only transport parcels whose time window ends within an hour. This choice is made because we do not want to have the situation where all vehicles are gone delivering parcels with low priority (the end of their time window is still far away), while a new batch of parcels arrives with many parcels having a time window which ends soon.

As quadcopters are the faster vehicle, they are preferred to be utilized. The parcels with the highest priority (so with a time window which ends the soonest) are assigned to the available quadcopters. If there are still parcels left whose time window ends within an hour, these parcels are loaded into streetrobots, given that the streetrobots are able to deliver them in time.

To conclude this section, it must also be noted that the level of battery charge available to a vehicle before it needs to be charged is increased. While simulation different scenarios, sometimes vehicles would run out of battery while travelling through the campus. A direct result of this is that the parcel it currently is transporting can not be delivered anymore. However, this also means that throughout the whole day, the vehicle would also be unable to transport any more parcels. To prevent this from happening the battery reserve attribute was increased from 20% of its total capacity to 40% of its total capacity.

IV. Experimental design

A. Warm-up Time

The simulation model simulates one day, after the run of the day the system is reset. This means all undelivered

parcels are removed and the vehicles are ready to be used again. This indicates the simulation is terminating and has transient behavior. For this type of simulation the system will not have a steady-state, so there is no warm-up period. How the system performs with initial values is what we want to know, in this case the configuration of the vehicles and the delivered parcels. Based on the selected KPIs the system can be evaluated.

B. Replications

To determine the amount of replications we use a replication and deletion approach. We look at the error of our run data and search for one below 95% confidence and an alpha of 0.0425. This is important to know since we want to perform experiments and the data needs to be of sufficient quality. The run data consists of data on how long a parcel stays in the system. This data is used to determine the amount of replications. For determining the number of replications, a vehicle configuration of 1 MobileParcelLocker, 5 StreetRobots and 5 QuadcopterNoHoist was used. This choice was made since this configuration can be seen as a configuration with average performance. The amount of vehicles was not very low nor very high, and the performance was good, while not being one of the configurations with the best performance.

The mean of the replication is calculated by finding the average time in system of all deliveries of all replications done. After a new replication is done the mean gets updated for that replication. The same is done for the Variance and the error rate.

The formula for determining the optimal number of replications [4], n^*, is given by:

$$n^* = \min \left\{ i \geq n : \frac{t_i \sqrt{\frac{S_n^2}{i}}}{\overline{X}} \leq \gamma' \right\}$$

where:

- i: Current replication number,
- S_n^2: Variance of the sample after n replications,
- $\overline{X}$: Mean of the sample,
- t_i: t-score for $i - 1$ degrees of freedom, based on your desired confidence level,
- γ': Acceptable error threshold.

At 13 replications, the error rate was smaller than the acceptable error threshold. Thus, throughout the experiments, 13 replications were done per experiment.

C. Experimental Setup

To conduct experiments and find the ideal configuration of vehicles, the experiment manager tool of plant simulation was initially used. however, after experiencing many issues in trying to get this to work, the decision was make the create the experimental setup ourselves in Plant Simulation. This gave us more control in what steps were taken and in what order they were executed. The experimental setup that we designed is the following:
At first, the amount of experiment which will take place is determined. The model has 4 input variables, maxMobileParcelLockers, maxStreetRobots, maxQuadcopterWithHoist & maxQuadcopterNoHoist. These variables correlate to the maximum amount of that vehicle that should be available in an experiment. Based on these values all possible combinations of vehicles are calculated and stored in the DataTable "Input". For each configuration of vehicles, the simulation runs 13 times (the reason for this is explained in the previous section IV-B). For each run, the outcome for each relevant KPI is stored in the DataTable "RunResults". After each experiment, the average value for each KPI is computed and stored in the DataTable "Output".

After all experiments are finished, the best configuration of vehicles must be determined. This is done by calculating a score from all KPI values. The calculation of this score is further explained in the next section. The configuration of vehicles with the highest score is chosen as the best configuration. After, a test is done to determine how low the timewindow duration can be while maintaining a good delivery rate. This process is further explained in the section "Optimization time window and delivery system"".

D. Experimental Factors

In the plant simulation model, in the class library we used the models.model to do the experiments and simulate the configurations. How this looks can be found in Appendix A . Based on what vehicle configurations we want to test, variables can be adjusted. The number of runs replications to get a good enough representation of the data is 13 runs with the same vehicle configuration(determined in the section "Replications"). To make sure the configurations can be compared with each other the same random number generator seed is used for each run in the different experiments.

Not every KPI has the same level of importance, so weights are assigned. Noise pollution, emission, and energy have a weight of 0.5. The average time in the system has a weight of 1. The costs have a weight of 2, and the delivery rate has a weight of 3. These KPIs have been normalized and added together. The configuration with the highest score wins, this means

it has the best overall performance. We ran the type of vehicle with the ranges of the number of that vehicle. Based on these experiments we found the configuration of number of quadcopter with hoist, streetrobots, mobile parcel lockers, and quadcopters without hoist.

When running a random experiment with the upper and lower bound ranges as shown in the table VII

Vehicles	Range
quadcopter with Hoist	1-5
Street robot	3-8
Mobile Parcel Locker	1-3
Quadcopter without hoist	3-8

TABLE VII: Random Experiment Ranges

E. Optimization of time window

After all experiments are executed, the best configuration is determined by which experiment has the best score. This configuration of vehicles is then used to determine how low the duration of a time window can go while maintaining a good delivery rate (delivery rate of 90%). For adjusting the time window duration, the DataTable "TimeWindowDuration". For each row in this table, the column "duration" is multiplied by the variable WindowRate. First, an experiment runs with WindowRate set to 1.00. While the deliveryrate is above 90%, WindowRate is decreased by 0.01 and the DataTable containing time window durations is updated. Once the deliveryrate falls below 0.90, the simulation is finished and the last WindowRate where a sufficient delivery rate is achieved is saved to the variable TimeWindow. For each value of WindowRate, the achieved DeliveryRate is stored in the DataTable "TimeWindowResults"

V. Results

When running the random experiment we get a the following configuration as output, see Table VIII.

Vehicle	Amount
Mobile Parcel Locker	3
Street robot	8
Quadcopter with hoist	1
Quadcopter without hoist	3

TABLE VIII: Best configuration

For this optimal configuration we got these KPI Values IX:

KPI	Value
Noise pollution	141.09
CO_2 emission	4755.08 (gCO2)
Energy Consumption	22.70 (kWh)
Parcel's time in system	3680 (s)
Delivery rate	92%
Total costs	208.56 (€)

TABLE IX: KPIs optimal experiment

This configuration is logical since the streetrobots have low energy usage, low noise pollution and are in general with regards to the KPIs have a good performance. This is why a lot are used. The mobile parcel locker also has a good performance and since it has a large capacity can operate with the route more efficiently. The quadcopters have the highest energy consumption but do deliver parcels very fast, since they perform worse than the other vehicles with regards to the KPIs it is logical not that much are used. Looking at these KPIs we see that 92% of parcels get delivered with this configuration.

A. Statistic comparison results

to test our model and configurations we perform a paired two sample t-test with the best, and second best configuration. This statistical test is used to test if there is a significant difference between the two samples. This is done by looking at the average time in the system and the emissions.

Statistic	**Experiment 1**	**Experiment 2**
Mean	3683.9727	3682.7613
Variance	18153.7236	18009.4433
Observations	13	13
Pearson Correlation	0.9997641	

TABLE X: Paired Two-Sample t-Test: Average Time in System

Metric	**Value**
Z $(x - y)$	1.2114
Var[w]	0.6783
CI (Lower, Upper)	(0.4199, 2.0028)

TABLE XI: Paired Two-Sample t-Test: Z, Var[w], and CI for Average Time in System

For the average time in the system, the confidence interval for the mean difference does not include 0, meaning that the difference between the two experiments is statistically significant. If we look at the Z we see such a small difference in performance that this metric for the experiment might not be substantial.

Statistic	Experiment 1	Experiment 2
Mean	4699.4404	4718.5519
Variance	683208.8288	720380.0772
Observations	13	13
Pearson Correlation	0.9987788	

TABLE XII: Paired Two-Sample t-Test: Emissions

Metric	Value
Z $(x - y)$	-19.1115
Var[w]	169.6708
CI (Lower, Upper)	(-31.6286, -6.5945)

TABLE XIII: Paired Two-Sample t-Test: Z, Var[w], and CI for Emissions

For the emissions we find a mean difference of (-31.6286, -6.5945). This also has no 0 in the interval and indicates statistical significance between the tests. Here the negative difference in Z shows that the second experiment has higher emissions on average. This difference does seem more substantial. This difference could be explained by the fact that the emission of the StreetRobot is lower then other vehicles.

VI. Conclusions

This study investigated the feasibility and efficiency of implementing autonomous last-mile delivery systems on the University of Twente campus. Using discrete event simulation in Siemens' Plant Simulation software, we modeled, tested, and optimized various configurations of autonomous delivery vehicles under two scenarios: the Benchmark Scenario and the Decoupling Scenario.

The experiments identified the optimal configuration of vehicles to be 3 Mobile Parcel Lockers, 8 Street Robots, 1 Quadcopter with Hoist, and 3 Quadcopters without a hoist. This configuration achieved a delivery rate above the threshold of 90% while allowing the time window duration to be reduced by 19%. These results demonstrate a significant improvement in service quality and operational efficiency. Furthermore, the scoring method used in the study confirms that emissions, utilization, and overall system performance can be effectively optimized and balanced. The flexibility of the scoring framework allows Campus & Facility Management (CFM) to tailor the system to their evolving needs and priorities. The CFM should invest in drones considering at least this set-up. As demands can increase we recommend stress testing the system and continuosly revise the simulation model used for insights.

The proposed configuration aligns well with the university's goals of sustainability, efficiency, and reliability. It ensures a balance between the key dimensions of People, Planet, and Profit:

The optimized configuration ensures high service quality by minimizing parcel delivery times while adhering to strict time windows, improving overall reliability and customer satisfaction. The use of autonomous vehicles significantly reduces CO_2 emissions and noise pollution compared to traditional delivery vans, supporting the university's sustainability objectives. By optimizing vehicle utilization and operational costs, the proposed system minimizes expenses while maintaining high delivery success rates.

VII. Future research

For this research, we limited the scope to finding the best configuration of vehicles for last-mile delivery on the University of Twente campus. However, there are several opportunities to extend this work and investigate other aspects of the system. One potential direction is to explore alternative delivery options, such as a combined model where students could choose between having their parcels delivered to personal lockers or centralized stationary lockers on campus. This approach could help address the issue of undelivered parcels at the end of the day, provided their number remains manageable.

The financial feasibility of autonomous systems is another area worth investigating, particularly in terms of cost configurations and depreciation. Additional research into these aspects would help determine whether investing in autonomous vehicles is a cost-effective long-term solution for the campus. Uniform distribution assumptions in the current model also limit its realism. For example, certain buildings, such as dorms, may have a higher density of parcel deliveries than others. Incorporating data from delivery companies could help refine these assumptions and align the distribution patterns with real-world scenarios, similar to the approach taken for analyzing delivery times.

The model could also be extended to simulate multiple consecutive days of operation to assess the system's behavior over time. This is especially important for parcels that remain undelivered at the end of the day and need to be delivered the following day. Multi-day simulations could provide valuable insights into how the system performs in a steady state, including fleet utilization and long-term operational efficiency.

Additionally, external factors such as weather conditions, varying parcel arrival patterns, and unexpected delays could be incorporated to increase the model's realism. These factors would allow for a more comprehensive evaluation of the system's robustness under diverse scenarios. Lastly, further statistical analysis could be conducted to refine and optimize the Key Performance Indicators (KPIs).

References

[1] ANWB. Wat kost 1 kwh?, 2024. Accessed: 2024-11-08.

[2] E. Commission. Co2 emission performance standards for cars and vans, 2019. Accessed: 2024-11-08.

[3] Klimaatmonitor. Co2-uitstoot dashboard, 2024. Accessed: 2024-11-08.

[4] A. M. Law. *Simulation Modeling and Analysis*. McGraw-Hill Education, 5th edition, 2014.

Appendix A
Experimental lay-out

In figure 1, the experimental lay-out can be found. In order to conduct an experiment, the following steps need to be taken.

First of all, the settings for the experiment need to be determined. The limit for the amount of vehicles can be adjusted by changing the value of the variables maxMobileParcelLockers, maxStreetRobots, maxQuadcopterNoHoist and maxQuadcopterWithHoist, with each variable corresponding to its vehicle. Do make sure to run the CreateInput method in order for the changes to apply. If you wish to change the minimum amount of vehicles it can be done within the CreateInput method. It speaks for itself, but do not give the variable indicating the maximum amount of vehicles a lower value than the variable indicating the minimum amount of vehicles.

After this, the simulation can be started by starting the EventController. While the simulation is running, the current experiment (and the total amount of experiments) are shown as the variables currExp and totalExp in the Experiment Configuration. After each experiment, the results of the experiment are stored in the DataTable Output.

After each experiment, the KPI variables and relevant DataTables are reset. This is done in the method ResetValues. This method is called in the Reset method.

At the end of the whole simulation, a score is calculated for each experiment and the experiments are ranked based on this score. The calculation of the scores is done in CalculateResults and the scores are stored in the DataTable Results. The scores are calculated based on the values for the KPIs which are used and the weights given to the KPIs. The weights for each KPI can be found in the DataTable Weights.

The vehicle configuration with the best score is shown in the box on the right ("Best configuration"). For now this box shows the results shown in the results section of this report, but these will change once a new simulation has finished.

Using the best configuration of vehicles, another simulation will be conducted to determine how small the time window duration of parcels can become while still maintaining a good delivery rate (greater than 90%). This process is started at the bottom of the method EndSim. When a simulation is finished, the method GetTimeWindowRate will be called. This methods starts the new simulation to determine how small a time window can get. The variable WindowRate is used to change the size of time windows. While a good delivery rate is being achieved, WindowRate keeps decreasing by 0.01. WindowRate indicates the percentage of timewindow size which is used for the current experiment, with a

WindowRate of 1.00 indicating that the timewindow size is at 100% (so the same as normal), while lower values indicate a lower percentage of the orignial time window range. For each value of WindowRate, the results of the experiment is stored in the DataTable TimeWindowResults. Both the WindowRate and the DeliveryRate are stored. This process keeps going until the delivery rate falls below 90%. After this, everything is finished. The lowest value for WindowRate where a good delivery rate was achieved is stored in the variable TimeWindow in the "Best Configuration" section.
If you end a simulation preemptively, make sure that the variables currRun and currExp are set to 1 before starting a new simulation. This does not happen automatically, but not doing this may result in the new simulation not starting at the first experiment.

APPENDIX B
TECHNICAL DESCRIPTION

Frames and Objects:

- **Model:** The control panel on which we run experiments and host essential methods (e.g., Init, Reset, EndSim). This serves as the central hub for managing the simulation.
- **Campus:** The main frame that includes the physical layout of the campus, delivery routes, and the EventController. It also contains instances of DecouplingPoint and Destination.
- **Inputs:** This frame holds all input parameters, tables, and experimental factors required for running the simulation, such as fleet composition, time windows, and delivery batch configurations.
- **Outputs:** This frame stores the results of the simulation, including KPIs like noise pollution, emissions, energy consumption, delivery success rates, and costs.
- **Methods:** This frame contains the main methods used in the simulation, such as CreateDeliveries, AssignParcelsToVehicle, and PerformRoute. It is inherited by the Campus frame to synchronize methods with the EventController.
- **DecouplingPoint:** Models the central delivery hub where parcels are transferred from delivery vans to autonomous last-mile vehicles.
- **Destination:** Represents delivery locations on campus. Each destination is modeled as a frame and visualized as a grey cube on the Campus frame.
- **Model:** An empty frame that can be duplicated for extending the simulation model as needed.

Key Objects and Attributes

- **Parcel:** Represents parcels with the following attributes:
  - *AssignedToVehicle:* Boolean indicating whether a parcel is assigned to a vehicle.
  - *DeliveryDestination:* The destination frame where the parcel should be delivered.
  - *DeliveryMarker:* The marker closest to the destination.
  - *StartWindow, EndWindow:* The parcel's delivery time window.
  - *Status:* Indicates the current state of the parcel in the delivery process.
- **DeliveryModePool:** Manages vehicle instances. Each type of vehicle (e.g., StreetRobot, Quadcopter) has a dedicated DeliveryModePool.
- **Vehicles:** Different types of vehicles are modeled, including:
  - *DeliveryVanDiesel* and *DeliveryVanElectric*: Represent benchmark delivery vans.
  - *StreetRobot:* Handles one parcel at a time.
  - *QuadcopterNoHoist* and *QuadcopterWithHoist:* Drones with and without hoist systems for delivering parcels.
  - *MobileParcelLocker:* Can carry up to 24 parcels.
- **RoadMarker:** Used for routing vehicles. Vehicles travel from one marker to another, stopping at the marker closest to the parcel's destination.

Simulation design

- Simulation Day: Each simulation run represents one day, starting at 09:00 and ending at midnight.
- Map Scale: The scale of the campus map is 1:14, meaning 1 meter in the model equals 14 meters in reality.
- Initialization and Reset: The simulation initializes vehicles, parcels, and routes based on the input parameters. The Reset method clears the state for a new run.

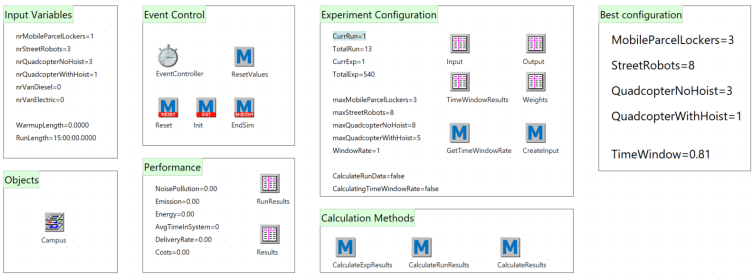

Fig. 1: Experiment Model

APPENDIX C
PROCESS FLOWCHARTS

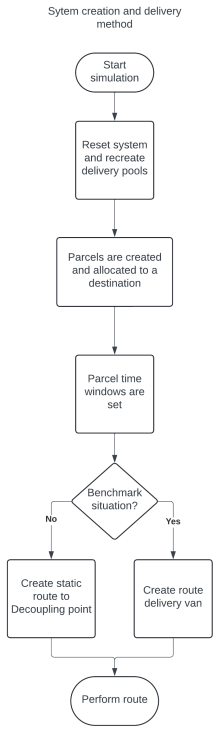

Fig. 2: Flowchart system creation

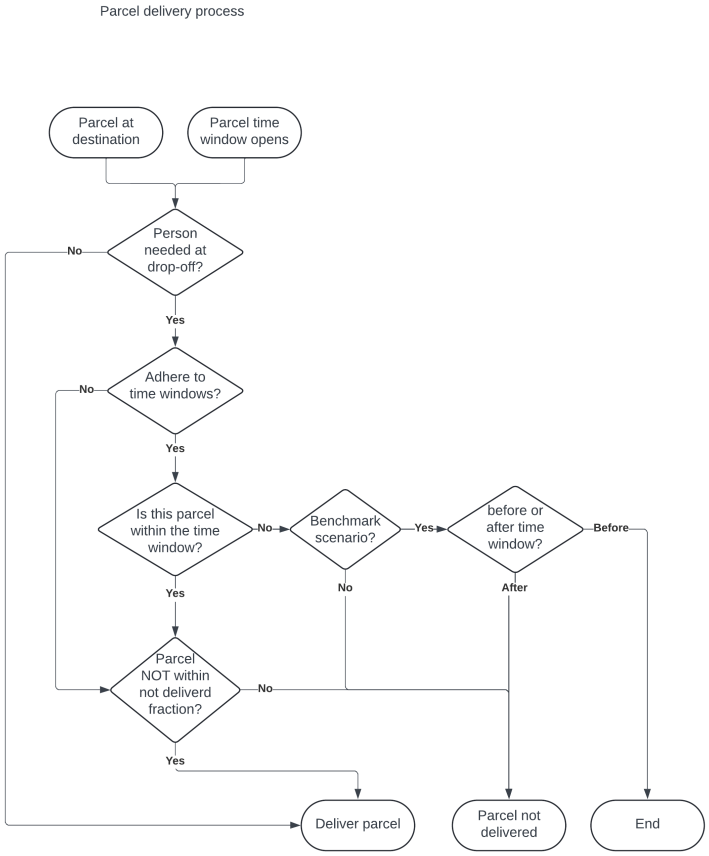

Fig. 3: Flowchart Delivery Process

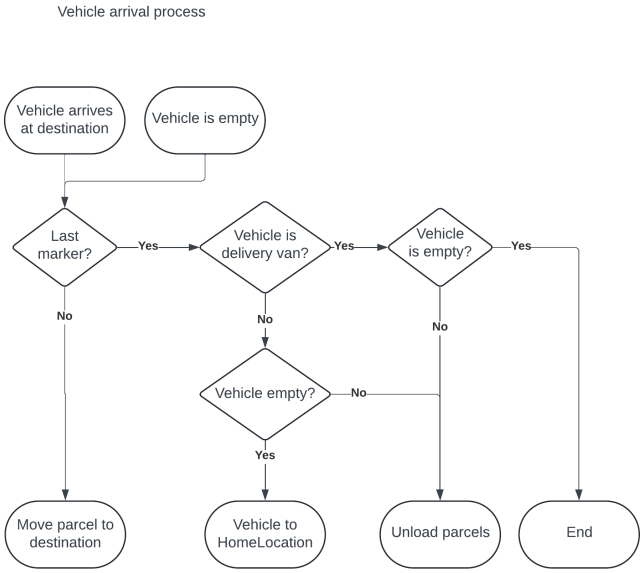

Fig. 4: Vehcle arrival process