Contract No. 777596

INtelligent solutions 2ward the Development of Railway Energy and Asset Management Systems in Europe

D6.4 Operations Optimisation

DUE DATE OF DELIVERABLE: 31/08/2019

ACTUAL SUBMISSION DATE: 06/12/2019

Leader/Responsible of this Deliverable: Markos Anastasopoulos, UNIBRI

Reviewed: Y

Document status		
Revision	Date	Description
1.0	2018.12.07	Final
1.1	2019.12.02	Final for the TMT check
1.2	2019.12.06	Final after TMT approval and quality check

Project funded from the European Union's Horizon 2020 research and innovation programme		
Dissemination Level		
PU	Public	x
CO	Confidential, restricted under conditions set out in Model Grant Agreement	
CI	Classified, information as referred to in Commission Decision 2001/844/EC	

List of Contributions by Partner
UNIBRI
IASA
PURELIFI
DV

Start date of project: 01/09/2017 Duration: 26 Months

Executive Summary

The present study focuses on the development of a dynamically re-configurable Information Communication Technology (ICT) infrastructure to support the sustainable development of railway networks. Once data have been collected, the extracted knowledge is used to develop a set of applications that can improve the energy efficient operation of railway systems. A typical example includes the identification of the optimal driving profiles in terms of energy consumption. In the present study, this is achieved through the adoption of an offline optimisation framework based on Data Envelopment Analysis (DEA) and an online framework based on Markov Decision Processes (MDP). To accelerate convergence, the MDP is combined with neural network models to reduce the strategy space. However, to successfully apply these models in realistic environments, an additional challenge that should be also addressed is associated with the positioning problem especially, in environments where GPS is not available. In response to this, a Visible Light Communication (VLC) positioning scheme extending PureLifi solution has been also developed. The performance of the proposed scheme is evaluated based on actual data collected at an operation tramway system. Preliminary results illustrate that when the proposed method is applied, a 10% reduction in the overall power consumption can be achieved.

Abbreviations and Acronyms

Abbreviation	Description
3GPP	3rd generation partnership project
ACL	Access control list
AP	Access point
DEA	Data Envelopment Analysis
EPC	Evolved packet core
FDMA	Frequency division multiple access
FSO	Free space optics
Gbps	Giga bits per second
GRC	Gnu Radio Companion
HetNet	Heterogeneous network
HSS	Home subscriber service
ICT	Information and Communication Technology
IP	Internet protocol
KPI	Key performance indicator
LAN	Local area network
LED	Light emitting diode
LiFi	Light fidelity
LTE	Long term evolution
LTE-R	LTE-railway
LWA	LTE WLAN aggregation
MANO	Management and orchestration
MDP	Markov Decision Process
MIPS	Million instruction per second
MME	Mobility management entity

Abbreviation	Description
3GPP	3rd generation partnership project
ACL	Access control list
AP	Access point
DEA	Data Envelopment Analysis
MOP	Multi-objective optimization
QoS	Quality of service
SDN	Software defined network
SSID	Service set identifier
TCP	Transmission control protocol
UE	User equipment
VLAN	Virtual local area network
Wi-Fi	Wireless fidelity
WLAN	Wireless LAN

Table of Contents

List of Figures

List of Tables

1 Introduction

According to the International Union of Railways the length of tracks maintained by the European railway sector exceeds 300.000 km operating more than 5 billion train-kilometres and offering services for more than 400 billion passenger-kilometres. A steady increase is expected for the next 30 years making railways a key-asset in the European transportation ecosystem [1]. Railway systems are expected to increase their *share in transportation by expanding and geographical reach and deliver innovative and integrated travel solutions for people and goods meeting the highest service standards in terms of safety and security*" [2]. Besides safety, security and capacity, a key aspect that should be considered during the design of railway systems is environmental sustainability. According to the EU climate actions for sustainable transport "*greenhouse gas emissions should be reduced by 80-95% below 1990 levels by 2050 whereas by 2030, the goal for transport will be to reduce GHG emissions to around 20% below their 2008 level"*. If a radically different approach is not adopted, *GHG* emissions from transport would remain above their 1990 level by 2050, congestion costs will further increase and accessibility gap between central and peripheral areas will widen.

To avoid these unaffordable scenarios and achieve the 2030 targets for energy efficiency, the introduction of novel concepts for railway systems will be key to lower not only CO2 emissions bus also maximise capacity. Aligned with the flagship initiative "*Resource efficient Europe*" set up in the Europe 2020 Strategy, the paramount goal of European transport policy is to "*help establish a system that underpins European economic progress, enhances competitiveness and offers high quality and scalable mobility services while using resources more efficiently*". In practice, transport has to use less and cleaner energy, better exploit a modern infrastructure and reduce its negative impact on the environment. Hence, new transport patterns must emerge, according to which greater numbers of travellers are carried to their destination by the most efficient modes. To achieve this goal, future development must address the following topics:

1. Improvement of the energy efficiency performance through the adoption of electric railway systems. A typical railway system has better energy conservation features than other transportation systems as it is able to transfer passengers with an energy consumption less than 209 kJ per kilometre per person, which is much lower than the energy consumption of cars.
2. Optimisation of the performance of railways through the adoption of novel algorithmic approaches.
3. Coordination and cooperation of the different railway subsystems (rolling stock, stations, substations and the grid etc) enabling efficient usage of transport and infrastructure.

The adoption of new technologies and novel software solutions such as the Internet of Things (IoT) and Artificial Intelligence (AI) will be key to lower transport emissions. These technologies can be successfully used to address capacity limitations of current railway systems and improve overall system's performance. In response to these challenges, the present study focusing on improving the performance of railway systems through the identification of optimal driving profiles adopting the Data Envelopment Analysis (DEA) theory. It is shown that when the proposed approach is adopted better performance can be achieved as the railway system can transfer more passengers in less time and with less energy reducing operational expenditures.

However, to successfully apply this concept in a realistic railway environment an extensive set of measurements covering a broad range of kinematic, energy and environmental parameters needs to be stored, processed and analysed. To address this challenge, an operational data management (ODM) platform has been deployed over an operational tramway system utilising open source technologies. The installed ODM system was able to monitor the energy flows of the whole railway systems and identify the optimal performance/cost trade-offs on the fly. As discussed in [[3]]-[4], this platform comprises the following core elements:

1. A heterogeneous secure and resilient telecommunication system, consisting of both wireless (e.g. LTE, WiFi, LiFi) and wireline (e.g. optical) technologies converging energy and telecom services. This infrastructure is used to interconnect a plethora of monitoring devices located both on track and at the trackside and end-users to the Operational Control Centre (OCC).
2. A hybrid data storage and processing mechanism combining state-of-the art open source SQL/Non-SQL databases as well as batch and stream processing engines. Based on the characteristics of the collected data and the selected applications, data are dynamically forwarded to the most suitable storage/processing platform. A high-level view of this process is shown in Figure 1.

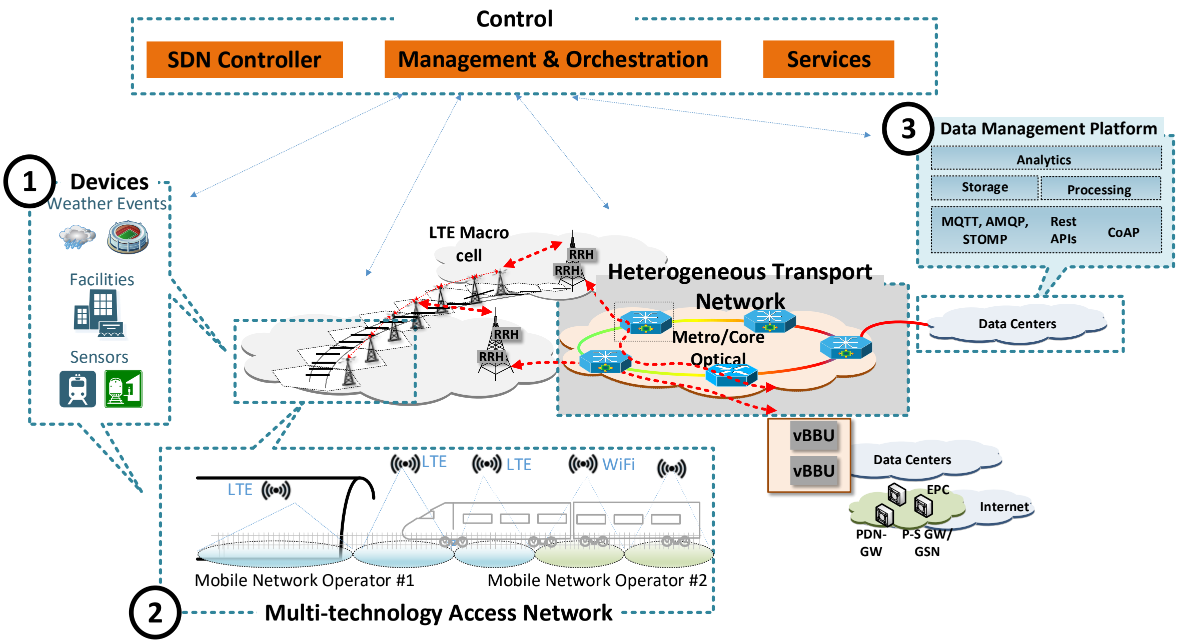

Figure 1. Converged Heterogeneous Network and Compute Infrastructures supporting railway services: Use case where data are collected from various devices (1) are transmitted over a 5G network (2) to the cloud-based data management platform (3).

The experimental campaign has been carried out over an actual tramway system operating at 750V. The rest of this study is organised as follows. The definition of the problem is outlined in Section 2, Section 3 gives a brief overview of the state of the art on the subject. The research methodology along with a description of the proposed off-line optimisation scheme is provided in Section 4. The online optimisation model based on Markov Decision Process is given in Section 5 whereas Section 5 concludes the deliverable.

2 Problem Definition

The transportation sector is a major contributor to air pollution and consumer of scarce non-renewable fossil fuels. Electric railway systems and vehicles powered by hybrid energy sources are expected to have high potentials to decrease fuel consumption due to their ability to regenerate kinetic energy in the braking phase. Despite the inherent energy efficient operation of electrified railway systems, reduction of the energy drawn from the power grid is still an overarching problem due to the following reasons:

1. The increasing costs of electric energy in some countries (in particular, where energy produced by non-renewable sources);
2. The increased electric energy demand due to several causes (lifestyle changes, which may be significant in developing countries; the need to supply more railway services, etc.);
3. The increased attention of the public and governments to the environment (air pollution) and to climate changes (greenhouse gas emissions).

A possible solution to reduce the energy consumption of railway systems is the identification of better driving styles. By optimising the way that a driver accelerates, maintains, slows or brakes, power consumption levels can be reduced. In the present study it is argued that the best performing driving profiles can be identified through the analysis of history measurements. To achieve this an experimental campaign has been carried out collecting data for more than three years over an operational tramway system. A smart metering system has been deployed monitoring energy, kinematic and environmental parameters based on sensing equipment installed both on-board and at the trackside. Once data have been collected and stored at the ODM system, two optimisation frameworks have been developed: an *offline* model based on DEA and an *online* based on Markov Decision Process (MDP) allowing the identification of the optimal driving styles that minimise the consumed energy subject to set of constraints related to scheduling, capacity and environmental conditions.

3 State-of-the-art

One very effective method of increasing the energy efficiency of railways is the optimised use of braking energy. As discussed in [7], regenerative braking of railway electric vehicles is effective when the electrical load exists near the regenerating train on the same electrified line. So, early in the morning and at midnight, or in the low-density district lines, regeneration cancellation phenomena often occur, and the regenerative brake force cannot be operated in accordance to the recommended value. Newly appeared high-performance energy storage devices press the issues of energy storage and reuse technologies on ground and on vehicles. Hybrid energy source is one effective solution especially in DC systems for city trams and urban railways it is not possible to coordinate all vehicle movements in such a way that a complete energy exchange between braking and accelerating phases can be reached. In [6] measurements and calculations have shown that modern vehicles are able to feed up to 40 percent of the consumed energy back into the grid. This is only possible, however, if the braking phase of one vehicle coincides with the accelerating phase of another vehicle in the vicinity, i.e. not more than one km away. Otherwise the energy must be wasted on the braking resistor. An energy storage unit installed in a DC substation can store the surplus braking energy that cannot be directly supplied to other vehicles and feed the energy back into the grid for subsequent accelerations. It has been proven in several installations that one energy storage unit is able to save up to 340,000 kWh per year [7].

Energy storage units can be installed both in existing and in new substations, whereby track section extensions can be built and operated at lower costs if such units are integrated between the substations as points of supply. The main problem of these methods is that they need some new hardware to be installed on the already existing trams. So, a more efficient and economy approach to reduce the energy consumption is the optimisation of driving profiles, speed and timetables. However, the regenerating braking and storing methods can be used in addition to optimised driving profiles. Most other methods provide optimised driving profile in a very different way. They divide tram's route into four steps, acceleration, cruising, coasting and braking. Energy consumption during the cruising and braking is less than during the acceleration phase. Therefore, the key to achieving energy-savings during train operation lies in making the correct choice of transition points for different states, that is, to find the optimal strategies of train operation to minimise energy consumption. The optimal strategies can be derived from the Hamiltonian function and the Pontryagin principle. According to the above two mathematical methods, the optimal strategies were further summarised as follows: maximum acceleration and braking during the beginning and end of a train journey, particle acceleration or braking during cruising, and starting to coast as early as possible. In a method, which have been developed in [8] the speed profile definition module identifies the speed profile that is able to respect some constrains, having fixed the values of cruising speed and average acceleration. The energy consumption estimation module calculates the consumed traction energy corresponding to the defined speed profile. The optimisation module operates on the decision variables, on cruising speed and average acceleration, so as to minimise the energy consumed. Evaluation of the objective function requires calculation of the speed profile and the energy consumed. The energy consumed is estimated by an equation, solving the differential equation of the motion by the finite difference approach. In [8] they've made a similar approach with [9] in order to solve the problem. The main difference is that in [9] they've made their own Energy and Running Time Simulator with different describing parameters of the driving operation. In [10] there is a different approach. The purpose of that study is to optimise the energy consumption by adjusting the timetable. As the timetable creation process is based on the predefined running times between stations, the definition of running times will directly impact not only the traffic planning but also the entire energy consumption. Generally, the running time can be adjusted within a certain range by using the running time supplements, which are extra running time on top of the technical minimum running time between every two stations; the longer the running time is, the lower the energy consumption for the same distance as we see in [11]. However, the longer the running time is, the higher the time cost of the railway sector and passengers; thus, a trade-off between the running time and energy consumption is necessary. In addition, the train order may change after adjusting the running time based on energy consumed, and then the train operation needs to be adjusted to maintain safety constraints, and the adjustment of train operation will result in a further change in energy consumption [12].

Table 1. Sample of the collected dataset.

Timestamp	External Temp	Speed	Current HVAC C2	Voltage (catenary)	Current (Ventilation)	Voltage HVAC	Total Energy Pantograph
	°C	km/h	A	V	A	V	kWh
1442729913	10.8	43	15.6	892	38.7	449.32	37.0573402
1442729914	10.8	40.8	15.6	891	37.9	449.28	37.00736674
1442729915	10.7	38.9	15.6	869	38.2	449.55	36.95579201
1442729916	10.7	36.9	15.6	874	38.2	449.64	36.90263086
1442729917	10.8	35.1	15.6	855	39.5	449.73	36.85689206

4 Off-Line optimisation model

4.1 *Data Collection Process*

To improve energy efficient operation of railway systems, initially, an ODM platform has been deployed enabling data collection and processing of information obtained from a variety of sensors and devices. This platform comprises a communication segment that relies on a set of optical and wireless network technologies to interconnect a variety of end-devices and compute resources. Through this approach, data obtained from various sources (monitoring devices, users and social media) can be dynamically and in real-time directed to the OCC for processing. The wireless technologies comprise cellular WiFi, LiFi and LTE networks to provide the on-board and on-board to trackside connectivity. For the trackside the to the OCC segment, information is transferred over an optical network. The overall solution is shown in **Error! Reference source not found.**. As mentioned above, this platform is used to monitor a variety of parameters. An indicative sample of the collected measurements is provided in **Error! Reference source not found.**. This dataset includes information related to the geographic location of the rolling stock, on-board CO2 levels that is used to estimate the number of passengers, internal and external temperature that is important for the evaluation of the Heating Ventilation and Air-conditioning system's (HVAC) performance, kinematic parameters (including acceleration and speed) etc.

The smart metering solution also comprises an Information Technology (IT) segment that is responsible for the storage and processing of the measurements. Storage is accommodated by hybrid mechanism combining state-of-the-art open source SQL/NoSQL databases while processing is executing based on Apache Spark. Using purposely developed algorithms, knowledge can be extracted from the dataset which can assist railway system operators to identify optimal train driving and scheduling profiles.

4.2 *Optimisation based on data envelopment analysis*

In the present study, identification of the optimal driving profiles is performed using DEA. DEA is a very powerful service management and benchmarking technique originally developed by Chames, Cooper and

Rhodes (1978) [13] to evaluate non-profit and public sector organisations. This is achieved by measuring the productive efficiency of the construction elements of these organisations, namely, decision-making units (DMUs). DEA can measure how efficiently a DMU uses the resources available to generate a set of outputs. The performance of DMUs is assessed using the concept of efficiency or productivity defined as a ratio of total outputs to total inputs. Note that efficiencies estimated using DEA are relative, that is, relative to the best performing DMU or DMUs (if multiple DMUs are the most efficient). The most efficient DMU is assigned an efficiency score of 1, and the performance of other DMUs vary between 0 and 1 relative to the best performance.

Table 2. Sample of the collected dataset. Sample of 10 routes used for the identification of the optimal driving profiles

StyleID	Inter-station Travelling Time (sec)	Total Energy (KW)	HVAC (KW)	CO2 (Average ppm)	Temperature °C
1	73	3139.2726	44.872627	47.979189	11.301351
2	77	2665.796	47.293833	38.555385	13.503846
3	73	4601.6475	29.122982	42.172973	14.404054
4	74	3397.467	45.642488	41.707368	14.797368
5	73	3146.8157	44.755127	45.322297	14.97973
6	77	3549.4091	307.04326	42.435443	15.134177
7	78	3334.6836	48.084032	42.387342	14.173418
8	75	3090.6305	45.894299	54.406974	13.892105
9	68	4883.8277	41.379654	46.720725	13.031884

To apply DEA in railway environments, driving styles are treated as DMUs. Now, let S be the set of driving styles extracted from the dataset with $\mathbf{X}_i, i \in S$, being the vector of inputs of style i, with N elements $x_{ij}, j \in N$. Let $\mathbf{Y}_i, i \in S$ be corresponding vector of outputs with size M ($\mathbf{Y}_i = [y_{i1}, y_{i2}, \dots, y_{iM}]$. Let also $\mathbf{X}_0 = [x_{01}, \dots x_{0N}]$ be the inputs of the driving style that we want to evaluate and $\mathbf{Y}_k = [y_{01}, .., y_{0M}]$ the output vector. Introducing parameter λ_i indicating the weight given to driving style i in its attempt to dominate Style 0, the measure of efficiency θ of Style 0 is determined through the solution of the following optimisation problem:

$$Min\ \theta$$

Subject to

$$\sum_{i \in S} \lambda_i x_{ij} \leq \theta x_{0j}\ , \forall j \in N \quad (1)$$

$$\sum_{i \in S} \lambda_i y_{ij} \geq y_{0j} \, , \forall j \in M \quad (2)$$

$$\lambda_i \geq 0 \forall i \in S$$

Constraint (1) limits the inputs of all other driving styles below the inputs used by the reference model 0, while equation (2) selects the driving styles that outperform style 0. The above problem is solved for all driving styles to identify the most efficient one.

In the present study, the optimal driving styles have been calculated taking as inputs parameters related to the in-cabin CO2 levels, the external temperature, the total driving time between adjacent stations, the total power consumption as measured by the pantograph and the power consumed by the HVAC system.

An indicative sample of the parameters characterising the driving styles is provided in **Error! Reference source not found.**, while the corresponding linear programming (LP) formulation considering only the first two styles is given below:

- *LP for evaluating Style 1:*

$$\min \theta$$

subject to

$$47.979189\lambda_1 + 38.555385\lambda_2 + 42.172973\lambda_3 \geq \; 47.9791890\theta \;\; (3.1)$$

$$11.301351 \, \lambda_1 + 13.503846\lambda_2 + 14.404054\lambda_3 \leq \; 11.3013510\theta \;\; (3.2)$$

$$3139.2726\lambda_1 + 2665.796\lambda_1 + 4601.6475 \, \lambda_3 \leq 3139.2726 \;\; (3.3)$$

$$44.872627\lambda_1 + 47.293833\lambda_2 + 291.22982\lambda_3 \leq 44.872627 \;\; (3.4)$$

$$73\lambda 1 + 77\lambda_2 + 73\lambda_3 \leq 73 \; (3.5)$$

$$\lambda_1, \lambda_2, \lambda_3 \geq 0$$

- *LP for evaluating Style 2:*

$$\min \theta$$

subject to

$$47.979189\lambda_1 + 38.555385\lambda_2 + 42.172973\lambda_3 \geq \; 38.5553850\theta \;\; (4.1)$$

$$11.301351\,\lambda_1 + 13.503846\lambda_2 + 14.404054\lambda_3 \leq \; 13.503846\theta \;\; (4.2)$$

$$3139.2726\lambda_1 + 2665.796\lambda_1 + 4601.6475L3 \leq 2665.796 \;\; (4.3)$$

$$44.872627\lambda_1 + 47.293833\lambda_2 + 291.22982\lambda_3 \leq 47.293833(4.4)$$

$$73\lambda 1 + 77\lambda_2 + 73\lambda_3 \leq 77 \; (4.5)$$

$$\lambda_1, \lambda_2, \lambda_3 \geq 0$$

4.3 Numerical Results

Solving the LP model for the styles shown in **Error! Reference source not found.**, the efficiency scores can be readily determined. The relevant results are provided in A similar set of results is shown in Figure 3 where the optimal profile that minimises the power consumption under end-to-end scheduling and passengers' constraints is illustrated. When the proposed method is applied, a 10% reduction in the overall power consumption can be achieved.

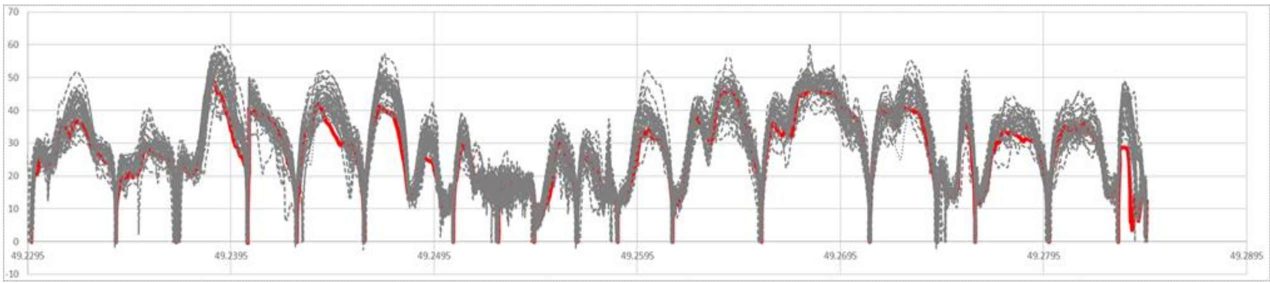

Figure 3: Optimal driving profile obtained when the DEA method is applied (red line) and comparison with styles obtained from measurements (grey lines)

Table 3.

An indicative set of results indicating the driving styles obtained when the DEA approach is adopted is shown in Figure 2. When the system is optimised for energy efficiency (green curve) the obtained driving style is smooth. On the other hand, when the system is optimised for shorter travelling times a higher average speed and steeper acceleration levels are observed.

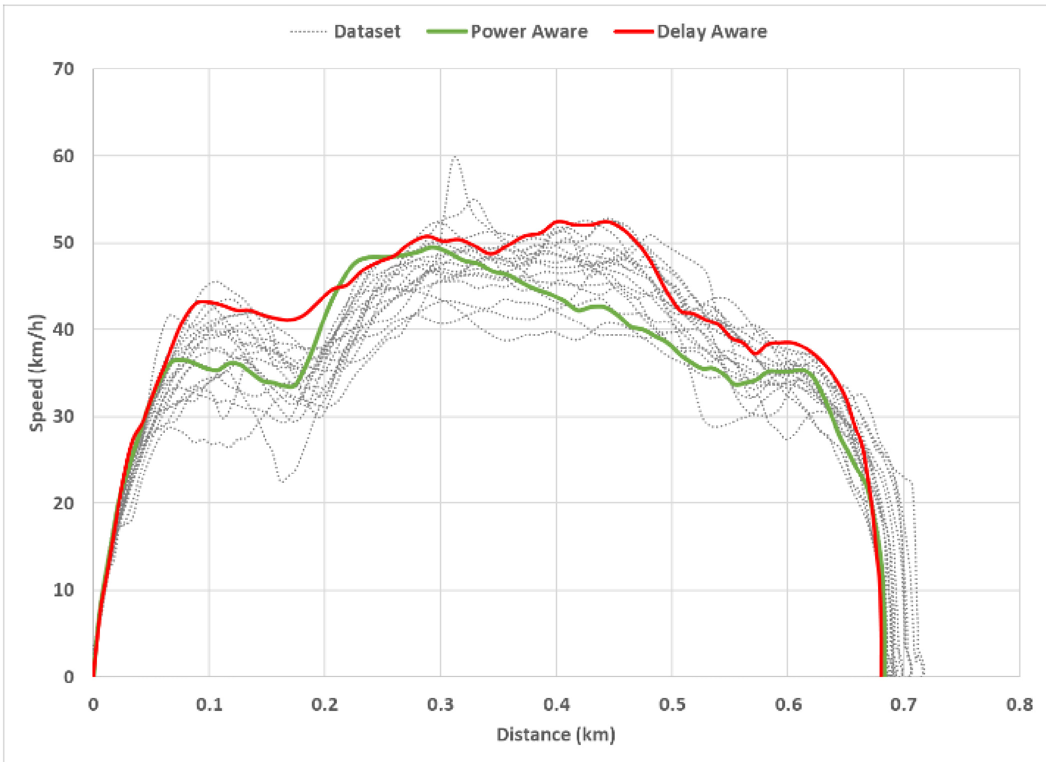

Figure 2: Tramway speed as a function KM distance for various driving profiles

A similar set of results is shown in Figure 3 where the optimal profile that minimises the power consumption under end-to-end scheduling and passengers' constraints is illustrated. When the proposed method is applied, a 10% reduction in the overall power consumption can be achieved.

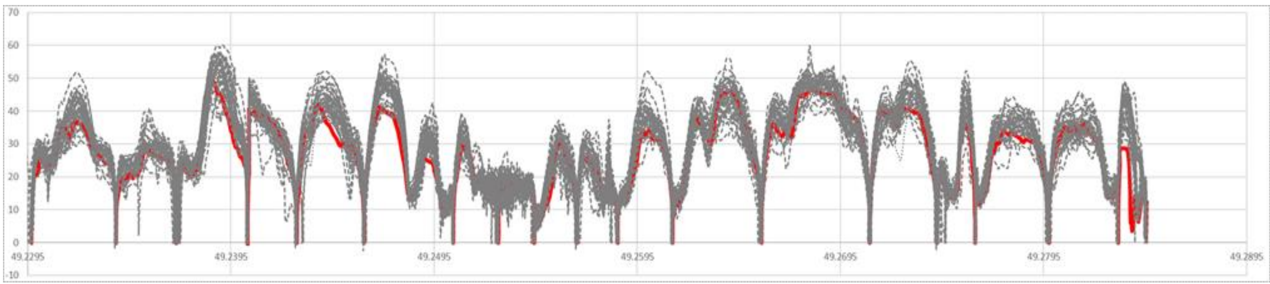

Figure 3: Optimal driving profile obtained when the DEA method is applied (red line) and comparison with styles obtained from measurements (grey lines)

Table 3: Efficiency scores for the driving styles shown in Table 2.

Style ID	Efficiency score
1	0.8099
2	0.7312
3	0.6887

4	0.6902
5	0.7649
6	0.6746
7	0.67
8	0.8975
9	0.819

In the method followed, the fastest routes were compared, those with the highest consumption and those with the slowest routes, respectively. In addition, similar time routes were compared to each other to arrive at the above results. For example, we notice that routes 1 and 5 reach their destination at the same time and have almost the same consumption, total and ventilation. However, CO2 levels in the cabin are higher in the case of the first route, so more passengers are transferred. Therefore, it is reasonable to get the result that route 1 is more efficient than route 5. Additionally, we notice that route 8 is more efficient than route 1. Also, in these routes the consumptions are similar, but we observe a considerable increase CO2. As a result, the tramway on route 8, with more passengers and lower consumption, arrived later to the station compared to route 1.

5 Real time optimisation based on Artificial Intelligence techniques

This section provides an overview of a modelling framework that can be used to dynamically optimise the performance of railway systems. Combining AI and machine learning techniques originally used in the development of self-driving cars, (Buehler et al., 2009) [23], a scalable optimisation framework has been developed that in the long run can be used in fully autonomous railway and tramway systems. To achieve this, the autonomous tramway/railway system relies into three components that perform *environment perception*, *decision-making* and *dynamic control* actions. Specifically, *environmental perception* is responsible to detect surroundings in real time via radar, lidars, GPS, and computer vision. In IN2DREAMS these actions are performed by the ODM system and the sensing devices that have been deployed on the train and at the trackside monitoring environmental, kinematic and energy related parameters. At this point it should mentioned that a key aspect that has been taken into consideration is positioning not only for areas where GPS is available but also in-tunnels where this service is blocked. To achieve this, a novel technology based on Visible Communications has been developed extending PureLiFi's solution that allows positioning inside tunnels. The *decision-making* framework understands the environment and predicts the optimal actions that should be considered. In railway systems, these actions are associated with the optimal acceleration/breaking that should be applied.

This deliverable focuses on the optimal decision-making for railway systems which is a challenging problem due the uncertainties that are introduced regarding environmental conditions, number of passengers transferred etc. Currently this type of decision-making processes is taken through finite state machines (FSM) and Convolutional Neural Networks (CNNs). FSMs where adopted in by (Urmson et al., 2008, Montemerlo et al., 2008) [24],[25] to prescribe rules that can be applied under different situations. Although the FSM is simple and effective in given situations it does not explicitly consider environment uncertainties and thus cannot be applied in a dynamic traffic scenario. In addition to this it requires manual classification and fails to take decision under uncommon situations. Chen et al. (2015) trained a convolutional neural network (CNN) model on 484,815 images collected and labeled when playing a car racing video game TORCS for 12 h. The model mapped an input image to a small number of key perception indicators that is then sent in the designed controller. Bojarski et al. (2016) [26] trained a convolutional neural network to map raw pixels from a single front-facing camera directly to steering commands.

In IN2DREAMS a probabilistic decision-making method, which can be applied in a dynamic railway system has been developed. The driving task was first formulated as the Markov decision process (MDP) by defining the environment state space, agent action space. Then, it built a state transition model and a reward model by using a prediction model of the energy grid and the surrounding environment. The optimal policy was then automatically deduced adopting three different techniques including the value iteration method of dynamic programming (DP), the Gauss-Seidel method and an LP approach. The simulation results show the preset goal can be achieved. The framework of the proposed method is shown in Figure 6.

5.1 *Preliminaries in Markov Decision Processes*

According to the Markov Decision Processes (MDP) theory, a decision agent, referred to agent, has the opportunity to affect the behaviour of a probabilistic evolutionary system by choosing its strategy. In general, by choosing a sequence of actions, the agents aim at optimising the system performance with regards to a certain criterion. MDPs are models for sequential decision making when outcomes are uncertain. Actually, MDPs are the extension of Markov Chains that add two extra elements to it, the decision and the reward. The main advantage of MDP is that it is an abstract and flexible framework which can be applied to many real-world problems in many ways.

An MDP may be a finite or an infinite horizon process and is fully characterised by the sets $S, A_{s_i}, p(.|s_i, a_{ij})$ and $r(s_i, a_{ij})$ where S denotes the set of system states. At any state $s_i \in S$, an agent, which is the learner and decision maker, has to choose an action a_{ij} from the set of actions A_{s_i} available at this state. As a result of choosing action $a_{ij} \in A_{s_i}$ at state s_i, an agent receives a reward $r(s_i, a_{ij})$ which may be negative (cost) or positive (benefit) depending on the current state of the system under consideration. The system performs a transition to a new state s_j with a transition probability $p(.|s_i, a_{ij})$. To optimise system performance, an optimal policy π^* must be determined. During the investigation of the optimal policy, the interaction between the agent and the environment (system) takes place in discrete time steps (t = 0, 1, 2, 3…). At each time step the agent receives some representation of the environment's state, and on this basis, it executes an action. One-time step later, the agent receives a numerical reward, known as immediate reward, from the environment and finds itself a new state. Therefore, the MDP and the agent generate a sequence or trajectory like this: So, Ao, R1, S1, A1…, Rt, St, At, Rt+1, St+1, At+1..., where the symbols R, S and A represent the set of actions, states and rewards respectively. The previously described interaction between the environment and the agent is illustrated in Figure 4.

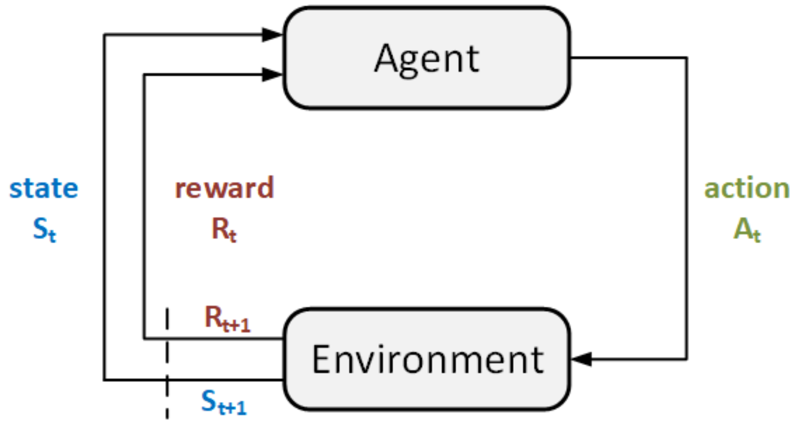

Figure 4: The agent – environment interaction in an MDP [27]

A policy is a sequence of decision rules, i.e. a sequence of functions at instances t, when decisions are made $d_t: S_t \rightarrow A_{s_{it}}$, providing the agents with a set of actions to choose when in state s_{it}. An optimal policy is a policy that satisfies a specific performance criterion. A common optimality criterion used in infinite horizon MDPs is to maximise the expected total discounted reward

$$\mathrm{u}_{\lambda}^{\pi}(\mathrm{s_i}) = E_S^{\pi} \sum_{t=1}^{\infty} \lambda^{\{t-1\}} r(s_i, d_t(s_{it}))$$

achieved when policy π is followed. The expected total discounted reward is known as the value of a state s under a policy π and it is called value function $\mathrm{u}_{\lambda}^{\pi}(\mathrm{s_i})$. $\lambda, 0 \leq \lambda < 1$, is a discount factor used to measure at instance t the reward received at instance t+1. Small values of λ emphasise near-term gain, whereas large values assign significant weight to later rewards. An immediate reward is the reward received immediately after the choice of an action a_{ij} at state s_i.

Therefore, the optimal policy π^* is the one satisfying

$$u_{\lambda}^{\pi^*}(s_i) \geq u_{\lambda}^{\pi}(s_i), \forall s_i \in S$$

To determine the optimal policy when at state s_i, an agent must determine the action a_{ij} that maximises the sum of the immediate reward, $r(s_i, a_{ij})$, and the expected reward. That is, an agent seeks solution to the Bellman equations:

$$u(s_i) = \sup_{a_{ij} \in A_{s_i}} \{r(s_i, a_{ij}) + \sum_{s_j \in S} \lambda\, p(s_j | s_i, a_{ij}) u(s_j)\}$$

The solution to the Bellman equations is the set of optimal value functions $u_{s_i}^*$ of the system.

The optimal action a_{ij}^* at state s_i is the one that satisfies

$$a_{ij}^* = arg \max_{a_{ij} \in A_{s_i}} u(s_i) = \operatorname*{argmax}_{a_{ij} \in A_{s_i}} \{r(s_i, a_{ij}) + \sum_{s_j \in S} \lambda\, p(s_j | s_i, a_{ij}) u(s_j)\}$$

The algorithms most commonly used to determine the optimal value functions and thus the optimal policy of the system are the policy iteration, the value iteration, and the linear programming. These algorithms are going to be described in the following sections.

5.1.1 Policy Iteration

According to policy iteration method, a policy π is improved using the value function $v_{\pi}(s)$ to a better policy π'. Then the value function $v_{\pi'}(s)$ is computed and it is used in order to improve the policy π' to a better policy π''. This process is illustrated in Figure 5, where the symbols E and I above the arrow denotes the policy evaluation and the policy improvement respectively. In a finite MDP problem, the policy iteration process converges to optimal policy and optimal value function in a finite number of iterations.

$$\pi_0 \xrightarrow{\mathrm{E}} v_{\pi_0} \xrightarrow{\mathrm{I}} \pi_1 \xrightarrow{\mathrm{E}} v_{\pi_1} \xrightarrow{\mathrm{I}} \pi_2 \xrightarrow{\mathrm{E}} \cdots \xrightarrow{\mathrm{I}} \pi_* \xrightarrow{\mathrm{E}} v_*$$

Figure 5: Policy Iteration process [27]

Figure 5 shows that the policy iteration method is separated into two basic steps; the iterative **policy evaluation** and the iterative **policy improvement**.

As already mentioned, we consider computing the value function $v_{\pi}(s)$ for an arbitrary policy π using the Bellman equation:

$$v_{\pi}(s) = \sum_{\alpha} \pi(\alpha|s) \sum_{s',r} p(s', r|s, \alpha)[r + \gamma v_{\pi}(s')]$$

In the iterative policy evaluation, instead of computing the accurate value function for an arbitrary policy π, we approximate it using the Bellman equation as an updating rule. We consider a sequence of approximate value functions v_0 , v_1, v_2, …, we arbitrarily select the initial approach v_0 and we obtain the approximation of the successive value functions by using the updating rule:

$$v_{k+1}(s) = \sum_{\alpha} \pi(\alpha|s) \sum_{s',r} p(s', r|s, \alpha)[r + \gamma v_k(s')],$$

for all $s \in S$. To produce the approximation of value v_{k+1} from v_k, the iterative policy evaluation applies the same operation to each state s. Specifically, the old value of s is replaced by a new value obtained by the old values of the next state s', and the immediate rewards, along all the one-step transitions possible under the policy being evaluated. At each iteration of the iterative policy evaluation the value of every state is updated once to produce the new approximate value function v_{k+1}. The sequence of the approximate values $\{ v_k\}$ converge to a v_{π} as $k \to \infty$.

In the iterative policy improvement, given the value function $v_{\pi}(s)$ for an arbitrarily policy π, the original policy π is updated to a new policy π' since the following condition:

$$q_{\pi}\big(s, \pi'(s)\big) \geq v_{\pi}(s)$$

is fulfilled. The value $q_{\pi}(s, a)$ is computed by the equation:

$$q_\pi(s,a) = \sum_{s',r} p(s',r|s,a)[r + \gamma \upsilon_\pi(s')]$$

The iterative policy algorithm is given in Table 4.

Table 4: Policy Iteration Algorithm [27]

1. Initialisation
   $\boldsymbol{V(s) \in R \; and \; \pi(s) \in A(s) \; arbitarily \; for \; all \; s \in S}$

2. Policy Evaluation
   Repeat
     Δ ⟵ 0
     for each s ∊ S:
       υ ⟵ $\boldsymbol{V(s)}$
       $\boldsymbol{V(s) \longleftarrow \sum_{s',r} p(s',r|s,\pi(s))[r + \gamma V(s')]}$
       Δ ⟵ $\boldsymbol{max(\Delta, |\upsilon - V(s)|)}$
     Until Δ < θ (θ is a small positive number)

3. Policy Improvement
   Policy-stable ⟵ true
   for each s ∊ S:
     old-action ⟵ $\boldsymbol{\pi(s)}$
     $\boldsymbol{\pi(s) \longleftarrow argmax_a \sum_{s',r} p(s',r|s,a)[r + \gamma V(s')]}$
     If old-action ≠ $\boldsymbol{\pi(s)}$, then policy-stable ⟵ false
   if policy-stable, then stop and return V ≈ υ* and π ≈ π*; else go to 2

5.1.2 Value Iteration

The value iteration process can be totally described by the following update operation that combines both the previously described policy improvement and policy evaluation steps:

$$\upsilon_{k+1}(s) = \max_a \sum_{s',r} p(s',r|s,a)[r + \gamma \upsilon_\pi(s')],$$

for all s ∈ S.

The value iteration method update is similar to the policy evaluation update which has been already described above. Like the policy evaluation, the value iteration converges to an optimal value function after an infinite number of iterations. Table 5 summarises the value iteration algorithm.

Table 5: Value Iteration Algorithm

Initialise $V(s) \in R \;\; arbitarily \; for \; all \; s \in S$
Repeat
  Δ ⟵ 0
  for each s ∊ S:
    υ ⟵ $V(s)$

$V(s) \leftarrow max_a = \sum_{s',r} p(s',r|s,a)[r + \gamma V(s')]$
$\Delta \leftarrow max(\Delta, |v - V(s)|)$
Until Δ < θ (θ is a small positive number)
Output a deterministic policy π ≈ π*, such that
$\pi(s) = argmax_a \sum_{s',r} p(s',r|s,a)[r + \gamma V(s')]$

5.1.3 Linear Programming

A linear programming (LP) problem can be defined as a problem of maximising or minimising a linear function subject to linear constraints (equalities or inequalities) [28]. In the previous paragraph we discussed that the value iteration method guarantees convergence to an optimal value $V^*(s) = \max_a \sum_{s',r} p(s',r|s,a)[r + \gamma V^*(s')], \forall\, s \in S$. Linear programming can also be used to find the optimal value function $V^*(s)$ for all $s \in S$ in an MDP problem. The formulation to find the $V^*(s)$:

$$\min_V \mu_0(s)V(s)$$

$$s.t. \forall s \in S, \forall a \in A:$$

$$V(s) \geq \sum_{s',r} p(s',r|s,a)[r + \gamma V^*(s')],$$

where $\mu_0(s)$ is the probability distribution over S, with $\mu_0(s) > 0, \forall s \in S$.

Although in some cases the LP's convergence is better than it of policy iteration and value iteration methods, it becomes impractical for high dimensional problems.

5.2 Optimal train driving profiles using MDP

In this section, the general theoretical framework outlined in the previous section is employed to identify optimal driving profiles. The system comprises the possible locations of the rolling stock together with the acceleration policies. The set of possible system states of the MDP is denoted by

$$S = \left(\times_{i \in [1,N] s_i}\right) \cup s_0$$

where i is a location of the train (kilometric distance from a reference point), s_0 is the accident state and $s_i = [x_i, v_i]$ is a vector containing the kilometric position x_i of the train from the starting station and its speed v_i. s_0, is included in S to describe the "accident state" which occurs when some constraints are violated. This includes the cases where the speed of the train exceeds a specific limit. The destination node (station) $D(x_D)$ represents the "terminal state", s_D of the MDP.

At any state $s_i \in S$, the train has to select one action from the set of possible MDP actions $a_{ij} \in A_{s_i}, s_i \in S$ available at this state. The elements of this set determine the acceleration policy (and consequently the traction effort) that can be selected at any location. Based on the acceleration profile adopted and the current speed of the train, the location of the rolling stock for the upcoming time instant can determined. The transition probabilities of the MDP are given by

$$p(s_j|s_i, a_{ij}) = \begin{cases} p, & failure\ probabilty \\ 1-p, & success\ probability \end{cases}$$

where the failure probability, p, is the probability that movement of the train from state i to state j fails if action a_{ij} is selected; in this case, the system remains at state s_i and the driving profiles of the train must be readjusted. The success probability, $1-p$, is the probability that a train successfully moves from from position i to position j if action a_{ij} is selected; in this case, the system makes a transition to state s_j.

The terminal state s_D of the system is assumed to be an absorbent state. That is, once the train reaches the location of a station D, the system stays at this state with probability 1.

$$p(s_D|s_D, a_{Dj}) = 1 \qquad \forall a_{Dj} \in A_{s_D}$$

It is also assumed that, when the system is at the accident state, the system reaches the terminal state with probability 1 and the selected driving profile fails, i.e.

$$p(s_D|s_0, a_{0j}) = 1 \qquad \forall a_{0j} \in A_{s_0}$$

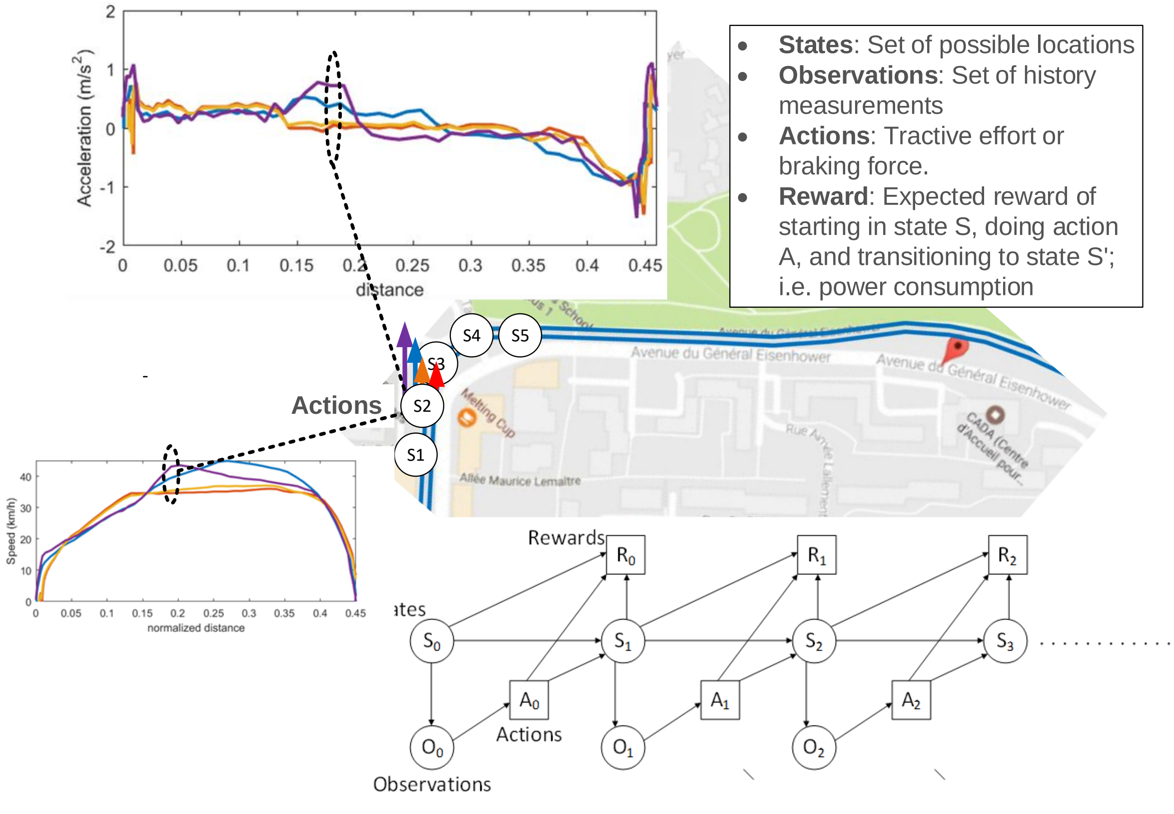

Figure 6 – Optimal driving profiles based on MDP: Decisions are taken every time instant. This allows different driving profiles to be combined.

The transition matrix of the system corresponding to the decision rule d is denoted by $\mathbf{P_d}$ and is a two dimensional matrix |S| × |S| with its $(i,j)^{th}$ element being the transition probability $p(s_j|s_i, a_{ij})$. In the framework of the present work, once an action a_{ij} has been chosen at state s_i an immediate reward R_{ij} is applied. This reward may be associated with the energy consumed or time spent for the rolling stock to be transferred from location i to location j.

5.2.1 Numerical Example

To keep the analysis tractable we assume the simple scenario where the train driver can select one of the following actions: to accelerate with rate $a_1 = 1m/s^2$, to keep a constant speed ($a_0 = 0m/s^2$) or to decelerate with $a_{-1} = -1m/s^2$. Increased granularity for the acceleration strategies can be also adopted at the cost of increased computational complexity. Initially the train located at the train station (x_0) and its position is described through state S_0 with $v_0 = 0m/s$. During the first stage of the problem the driver selects an acceleration strategy from the set $A_{s_0} = \{a_{-1}, a_0, a_1\}$. It is obvious that if the driver is at x_0 and selects either a_{-1} or a_0 will remain in the same position. In the second stage, after observing its status a new acceleration strategy is adopted and the value function is calculated. The same process is repeated over a finite time horizon.

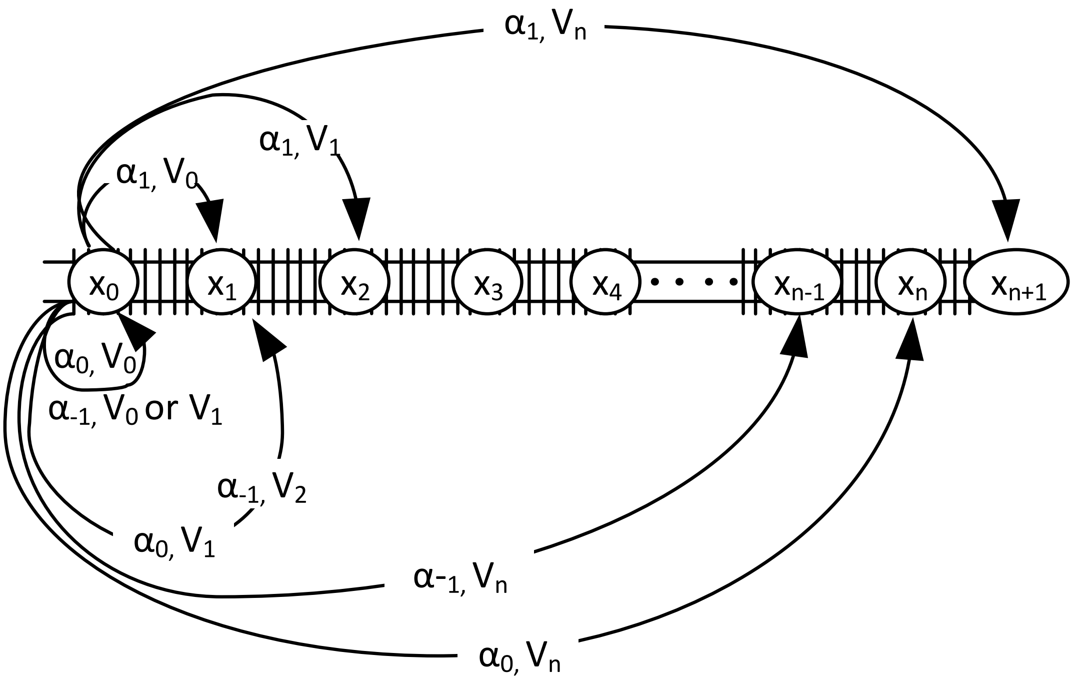

Figure 7 – State transition diagram.

In order to solve the MDP problem, the transition matrix P is first defined. As shown in **Error! Reference source not found.** when the train is in position $x_i, i = 0, \dots, n$ its speed tales values in the $v_0, v_1, \dots, v_m$ where for simplicity we assume that $v_j, j = 0, \dots, m$ with $v_j \in \mathbb{Z}$. In the next step all states defined through the tuple (x_i, v_j) are numerated. For example, for $n = 4$ and $m = 2$ the possible states are defined as follows:

$S_1 \leftarrow x_0, v_0$ | $S_6 \leftarrow x_0, v_1$

$S_2 \leftarrow x_1, v_0$ | $S_7 \leftarrow x_1, v_1$

$S_3 \leftarrow x_2, v_0$ | $S_8 \leftarrow x_2, v_1$

$S_4 \leftarrow x_3, v_0$ | $S_9 \leftarrow x_3, v_1$

$S_5 \leftarrow x_4, v_0$ | $S_{10} \leftarrow x_4, v_1$

$$S_{11} \leftarrow x_0, v_2$$

$$S_{12} \leftarrow x_1, v_2$$

$$S_{13} \leftarrow x_2, v_2$$

$$S_{14} \leftarrow x_3, v_2$$

$$S_{15} \leftarrow x_4, v_2 \; \ldots\ldots$$

In the general case, where the system is in position x_i with speed v_j its state will be denoted as S_k with $k = i + 1 + (n + 1) * j$.

When the action $a_{-1} = -1m/s^2$ is adopted the transition matrix will be given by:

$P(:,:,1) =$

$$\begin{array}{cc}
 & \begin{array}{cccccccccccccccc} S_1 & S_2 & S_3 & S_4 & S_5 & S_6 & S_7 & S_8 & S_9 & S_{10} & S_{11} & S_{12} & S_{13} & S_{14} & S_{15} & \end{array} \\
\begin{array}{c} S_1 \\ S_2 \\ S_3 \\ S_4 \\ S_5 \\ S_6 \\ S_7 \\ S_8 \\ S_9 \\ S_{10} \\ S_{11} \\ S_{12} \\ S_{13} \\ S_{14} \\ \\ . \end{array} &
\begin{bmatrix}
1-p & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & . \\
0 & 1-p & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & . \\
0 & 0 & 1-p & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & . \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & . \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & . \\
0 & 1-p & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & . \\
0 & 0 & 1-p & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & .. \\
0 & 0 & 0 & 1-p & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & . \\
0 & 0 & 0 & 0 & 1-p & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & . \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & . \\
0 & 0 & 0 & 0 & 0 & 0 & 1-p & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & . \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1-p & 0 & 0 & 0 & 0 & 0 & 0 & 0 & . \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1-p & 0 & 0 & 0 & 0 & 0 & 0 & . \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1-p & 0 & 0 & 0 & 0 & 0 & . \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1-p & . \\
. & & . & . & .. & . & . & . & . & & . & . & . & . & . & .
\end{bmatrix}
\end{array}$$

It should be noted that for simplicity the termination state is not depicted.

A similar matrix can be determined for a_0:

$P(:,:,2) =$

$$
\begin{array}{c|ccccccccccccccc|}
 & S_1 & S_2 & S_3 & S_4 & S_5 & S_6 & S_7 & S_8 & S_9 & S_{10} & S_{11} & S_{12} & S_{13} & S_{14} & S_{15} \\
S_1 & 1\text{-}p & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
S_2 & 0 & 1\text{-}p & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
S_3 & 0 & 0 & 1\text{-}p & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
S_4 & 0 & 0 & 0 & 1\text{-}p & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
S_5 & 0 & 0 & 0 & 0 & 1\text{-}p & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
S_6 & 0 & 0 & 0 & 0 & 0 & 0 & 1\text{-}p & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
S_7 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1\text{-}p & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
S_8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1\text{-}p & 0 & 0 & 0 & 0 & 0 & 0 \\
S_9 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1\text{-}p & 0 & 0 & 0 & 0 & 0 \\
S_{10} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1\text{-}p & 0 & 0 & 0 & 0 & 0 \\
S_{11} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1\text{-}p & 0 & 0 & 0 \\
S_{12} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1\text{-}p & 0 & 0 \\
S_{13} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1\text{-}p & 0 \\
S_{14} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1\text{-}p \\
 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1\text{-}p
\end{array}
$$

and a_1:

$P(:,:,3) =$

$$
\begin{array}{c|ccccccccccccccc|}
 & S_1 & S_2 & S_3 & S_4 & S_5 & S_6 & S_7 & S_8 & S_9 & S_{10} & S_{11} & S_{12} & S_{13} & S_{14} & S_{15} \\
S_1 & 0 & 0 & 0 & 0 & 0 & 0 & 1\text{-}p & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
S_2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1\text{-}p & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
S_3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1\text{-}p & 0 & 0 & 0 & 0 & 0 & 0 \\
S_4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1\text{-}p & 0 & 0 & 0 & 0 & 0 \\
S_5 & 0 & 0 & 0 & 0 & 1\text{-}p & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
S_6 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1\text{-}p & 0 & 0 & 0 \\
S_7 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1\text{-}p & 0 & 0 \\
S_8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1\text{-}p & 0 \\
S_9 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1\text{-}p \\
S_{10} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1\text{-}p & 0 & 0 & 0 & 0 & 0 \\
S_{11} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
S_{12} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
S_{13} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
S_{14} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 & 0 & 0 & 0 & 0 & 0 & 0 & 1\text{-}p & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{array}
$$

The reward matrix can be easily determined by assigning a positive value when an action is successfully applied and a large negative value when the same action results in an event that violates normal operating conditions (speed limit violation). If the system is optimised for minimum traveling time, the positive rewards may be associated with the speed of the train. In case where the system is optimised for energy efficiency rewards are associated with the traction effort (i.e minimise the total number of instances where a positive acceleration has been applied).

From the above it is clear that the number of states increases with the available actions leading to increased computational cost. A comparison between the MDP linear programming model, the MDP value iteration approach modified to handle sparse matrices and the MDP Gauss-Seidel models is given in **Table** 6

Table 6: CPU time for various MDP models ($n = 4$ and $m = 2$)

	MDP Linear programming	MDP value iteration modified for sparse matric	MDP Gauss-Seidel
CPU time	0.8125	0.011	0.0156

We observe that the MDP approach modified for sparse matric manipulation outperforms the others.

5.2.2 Accelerating convergence by reducing the state space

Although the MDP model can be successfully used to identify the optimal actions that should be adopted it is suffers increased computational complexity. To reduce this complexity, the available strategy can be reduced selecting only those acceleration profiles that appear in the system with high probability. To achieve this clustering techniques can be applied. Clustering is unsupervised learning that assigns labels to objects. Sets, partition matrices, and/or cluster prototypes may be mathematically represented by cluster partitions. Sequential clustering (single linkage, complete linkage, average linkage, Ward's method, etc.) is easily implemented but computationally expensive. Partitional clustering can be based on hard, fuzzy, probabilistic, or noise clustering models. Cluster prototypes can take many forms such as hyperspheric, ellipsoidal, linear, circles, or more complex shapes. Relational clustering models find clusters in relational data. Complex relational clusters can be found by kernelisation. Cluster tendency assessment finds out if the data contain clusters at all, and cluster validity measures help identify an appropriate number of clusters. Clustering can also be done by heuristic methods such as the self-organising.

For example, driving profiles with similar characteristics with similar characteristics can be clustered. To be able to group routes that are similar requires each route to be represented by a feature vector in the same n-dimensional space. For each route the mean values for Temperature (T1 and T2), Velocity, and CO2 Level were computed. Three different feature vectors ([T1, T2], [T1, T2, CO2], [T1, T2, Velocity, CO2]) were used as inputs into four different clustering Algorithms (KMeans, Birch, Mean Shift, DBSCAN). Based on this approach the MDP model will include in the action set values that belong in the same cluster.

5.2.2.1 KMeans

KMeans is a general-purpose clustering algorithm and it is considered as one of the best. KMeans needs to be given the number of clusters. As experimentation it was decided to run three different scenarios. One for 2 clusters, another one for 3, and lastly, one for 4 clusters. The results produced from this algorithm are presented in Figure 8. Because one of the feature vectors has four attributes and a 4-dimensional space cannot be visualised, Principal Component Analysis (PCA) was used, for dimensionality reduction, so visualisation of the clustering could be possible.

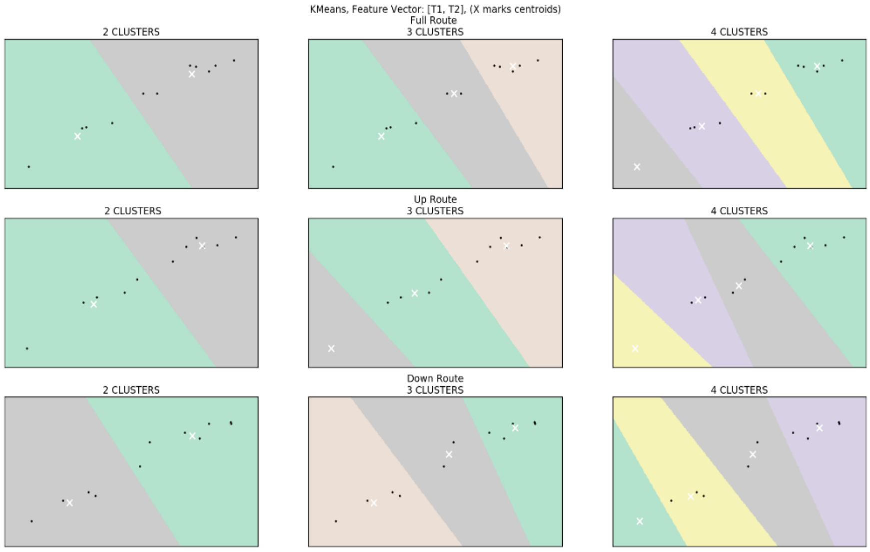

Figure 8: KMeans, Feature Vector: [T1, T2]

5.2.2.2 Birch

Birch was the second clustering algorithm that was chosen to experiment with. It has been proven scalable and even although it is slower than KMeans, it is faster than almost any other clustering algorithm. As Birch algorithm can be fed with the number of clusters, the same methodology as the one for Kmeans was used. The plots of the clustering results with Birch algorithm, for each different “route” category, are presented in Figure 9.

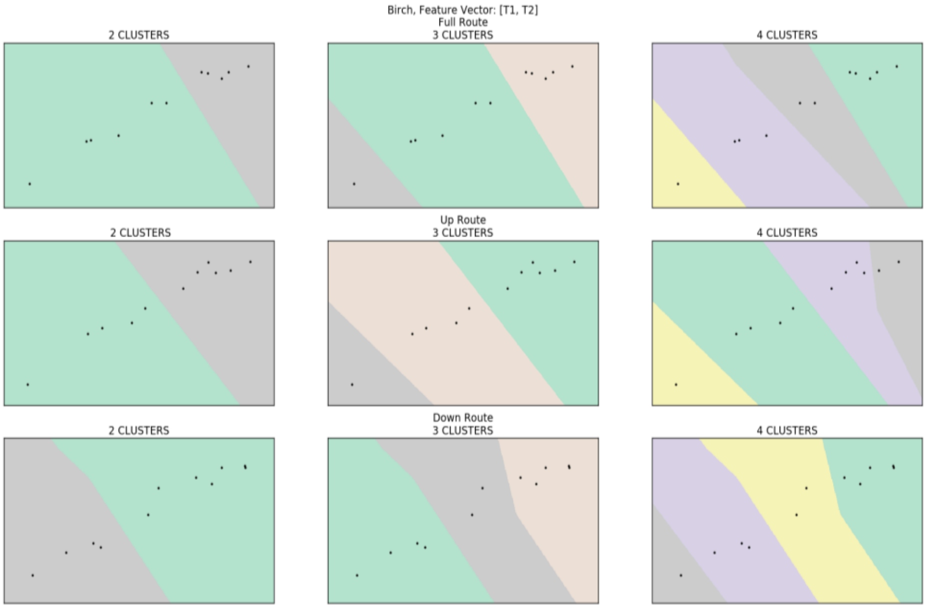

Figure 9: Birch, Feature Vector: [T1, T2]

5.2.2.3 Mean Shift

Mean shift was the next clustering algorithm that it was used. The main difference observed when compared to KMeans and Birch is that the number of clusters is not provided to the algorithm. On the contrary, the number of clusters has to be discovered by the algorithm. For that reason, the experiments with Mean Shift algorithm were restricted only to different feature vectors and different route categories. The results produced by the the Mean Shift clustering algorithm are shown in Figure 10.

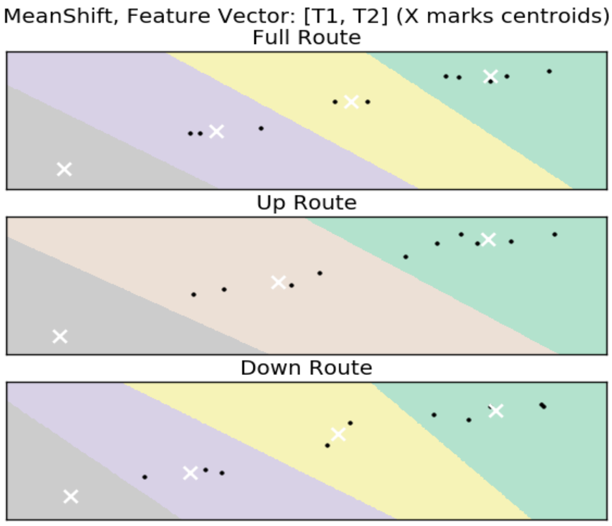

Figure 10 Mean Shift, Feature Vector: [T1, T2]

5.2.3 Numerical Example

The optimal driving profiles using the MDP approach for the Reims tramway use case are shown in Figure 11 and Figure 12. To keep the analysis tractable, we assume that the possible set of acceleration levels for the rolling stock ranges between -1-1m/s^2 with step 1m/s2^.We observe that the driving profiles calculated through the MDP model and the measurements have similar trends. The difference lies in the acceleration granularity step which for the MDP model was 1m/s^2. By increasing granularity better matching is expected.

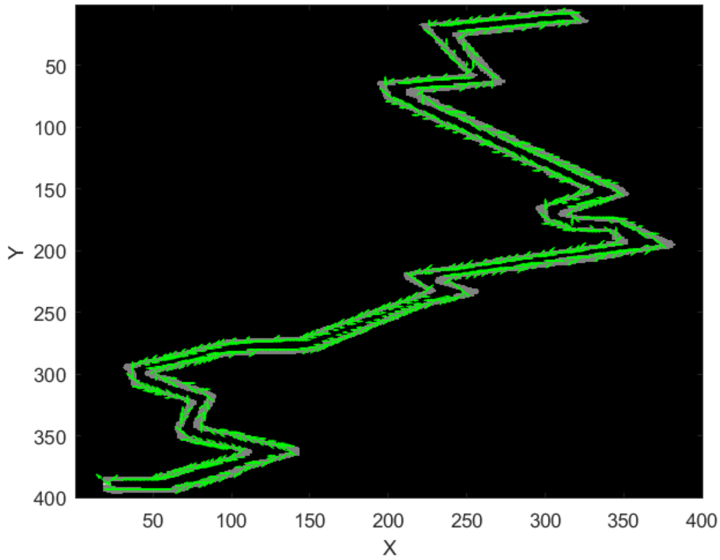

Figure 11 The REIMS tramway map along with the trajectory of the train

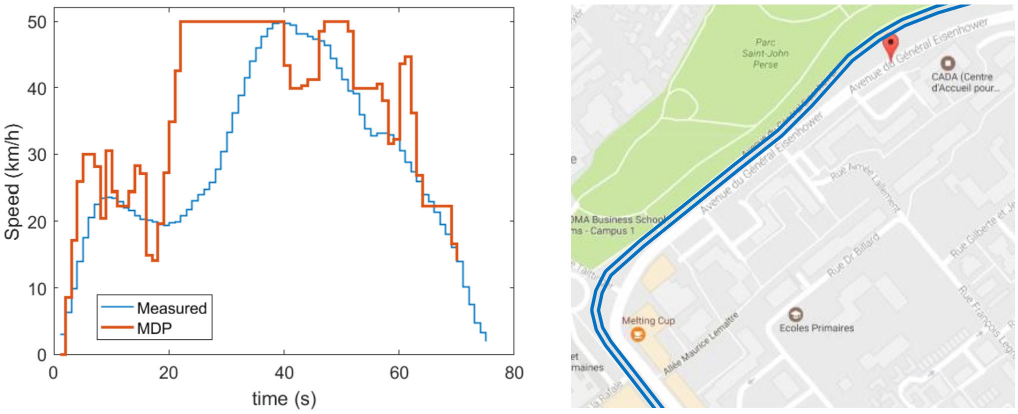

Figure 12 Comparisons between measurements and optimal driving profiles using the MDP framework for the part route from station 8 to station 9.

5.3 *In-tunnel Localisation system*

The above models require knowledge of the position of train at every time instant so that the appropriate action is applied. PureLiFi has designed and implemented an LiFi based, in-tunnel localisation system. This section presents the system specification, test plan and test results collected.

5.3.1 System requirements

The localisation system is required to determine the position of the train within a tunnel and relay this information to a subscriber located on-board of the train. Table 7 shows the system requirements of the PoC system.

Table 7: System requirements

Parameter	**Value**	**Notes**
Distance between tunnel ceiling and top of the carbody	5m	-
Maximum speed of the train	50 km/h	During the testing, the maximum permissible speed shall be evaluated.
Number of LiFi APs	3	Mounted on the tunnel ceiling
Number of LiFi STAs	1	Mounted on top of the train
Communication protocol to forward location information	MQTT	

5.3.2 System specification

The LiFi based localisation solution is based on pureLiFi's LiFi XC system. The commercial system is specified in detail in Deliverable 2.2.

5.3.2.1 Principle of operation

For this solution, the APs are mounted on the ceiling of the tunnel. The luminaires installed with the APs produce a 66 degree cone angle downlight.

On the train side, a Raspberry Pi computer and the LiFi STA are mounted on top of the train. As the train passes under a LiFi AP, the STA captures a beacon periodically transmitted by the AP. This beacon includes the MAC address of the AP, which is unique to each AP. This MAC address information is received at the MQTT subscriber where the location can be determined based on this.

In order to accurately detect the location of the train, it is crucial that at least one beacon is transmitted while the LiFi STA is within the LiFi cell.

5.3.2.2 Localisation accuracy

The accuracy of the system is dependent of the cone angle of the downlight. In principle, the system is to be designed such that there are no overlaps between the coverage areas of each lamp. This ensures that only one AP MAC address can be reported at any given location. Figure 13 below shows the geometry of the proposed system.

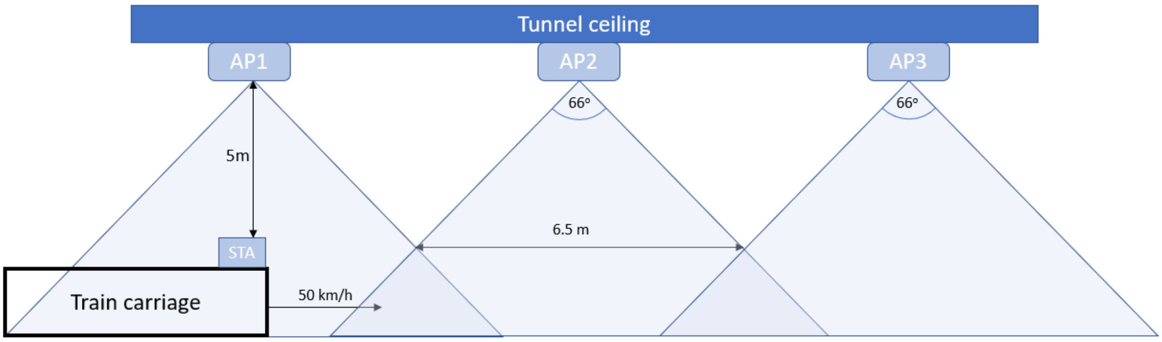

Figure 13: Geometry of localisation system

It can be seen that with the standard LiFi XC lamp, the location can be determined with a 6.5m resolution.

In order to ensure that the STA always identifies the location of the train successfully, the beaconing period of the AP has been reduced to 1.28ms. As a result, a train travelling at a speed of 50km/h will see at 365 beacons from a single AP, worst case. Therefore the requirement to capture at least one beacon/AP at 50km/h is satisfied.

5.3.2.3 System architecture

Figure 14 below illustrates the system architecture.

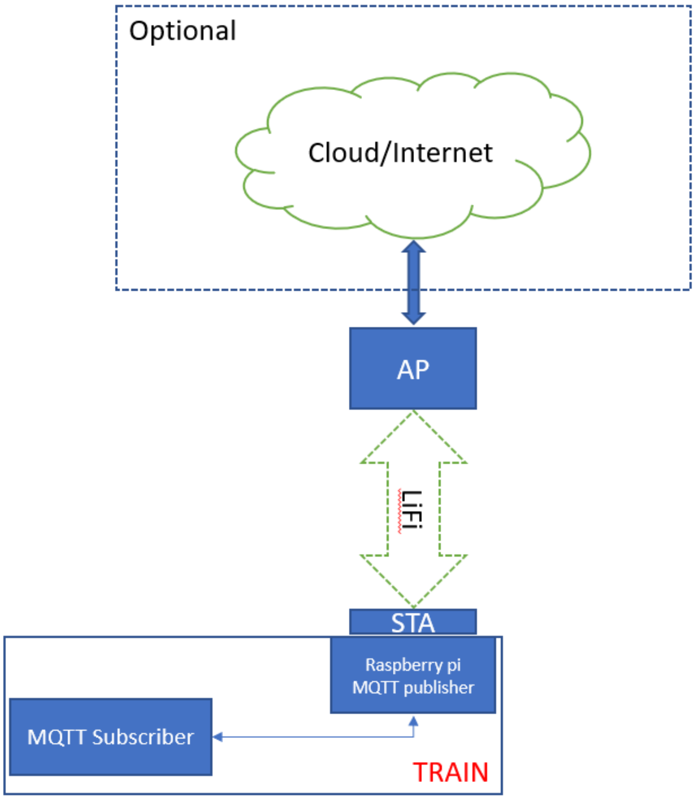

Figure 14: System architecture

The LiFi Station is connected with a Raspberry Pi computing device that is capable of detecting the MAC address of access point and update that information via MQTT commands. A python script (msg_send.py) is the command used to publish the message to the MQTT subscriber.

The MQTT subscriber is used as broker of the system. mosquitto_sub -t "Topic" command can be used to subscribe the messages.

JSON is an open standard format that uses human-readable text to transmit data objects consisting of attribute–value pairs. The detected MAC address information is packed with this JSON format and this can be processed in the subscriber side of the system.

5.3.3 Test plan

5.3.3.1 Test 1 – Functional test

The purpose of this test is to verify that the STA can successfully detect and report the MAC address of the AP it is currently under.

The test set-up is shown in Figure 15 below.

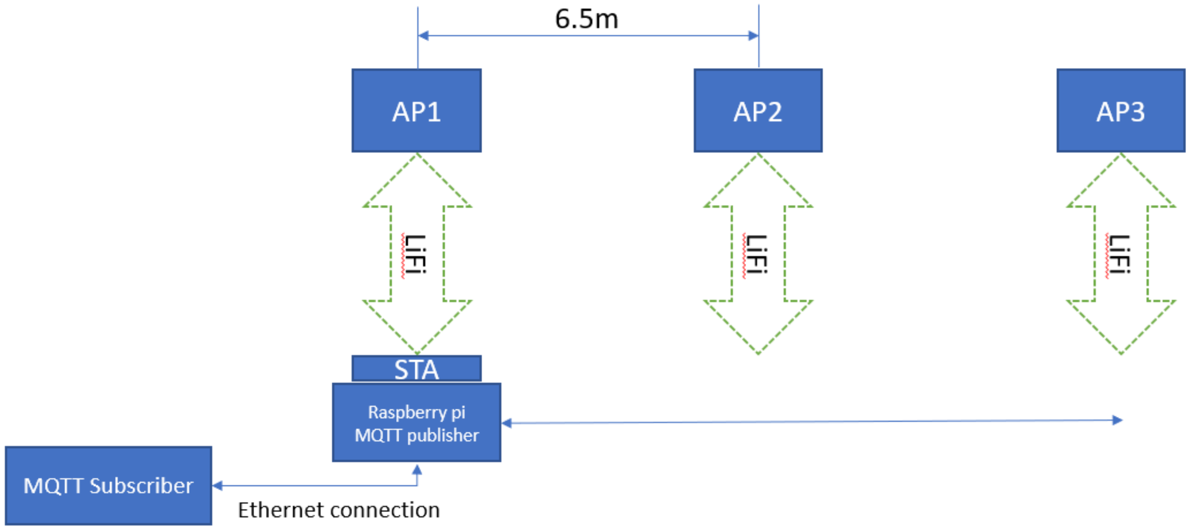

Figure 15: Functional test set-up

3 APs are set-up with 6.5m spacing between the centre points. The LiFi enabled Raspberry Pi is connected to a laptop acting as the MQTT subscriber. Using a simple script, the MAC address of the AP can be requested and printed.

To verify the system functionality, the STA must be moved between the coverage areas of the APs, and it must be verified that the MQTT subscriber can pull the correct MAC address.

5.3.3.2 Test 2 – Maximum achievable speed

This test is used to confirm the maximum speed of the train that can be detected by the tracking system. The train must spend a certain time under an AP to establish the MAC L2 LiFi connection. The aim of this test is to find the minimum time of required for this connection establishment process.

The following test procedure shall be used:

1. Disable LiFi functionality of AP1
2. Move the STA under AP1
3. To simulate the train movement, enable and disable LiFi functionality (beaconing) for a short period of time
4. Verify that the location information was updated
5. Repeat the above steps with different “ON” times in step 3. After finding the shortest “ON” time that results in detection, the maximum speed of the train can be determined.

Following command can be used for step 3:

Lifictl -p2; sleep “duration in s”; lifictl -p0

5.3.4 Test results

5.3.4.1 Test 1 – Functional test

The STA was moved between AP1, 2 and 3. In total 30 handovers were executed. After each handover, a successful MAC address update was observed.
The functional test was successful.

5.3.4.2 Test 2 – Maximum speed test

The STA was placed under an AP, the LiFi functionality of the AP was enabled for a short period of time before disabling again. The actual on duration was measured (due to expected variation between the requested and the actual ON duration).

For each duration, the ON-OFF test was repeated 20 times. The Table 8 summarises the observed results. The output of the test script for the 25 and 20 ms cases is shown in the Appendix.

Table 8: Maximum speed test, test results

Requested duration (ms)	Measured duration (ms)	Corresponding train speed (km/h)	Successful detection	Failed detection	Pass/Fail
150	181	129	20	0	Pass
100	135	173	20	0	Pass
75	105	223	20	0	Pass
50	80	292	20	0	Pass
25	59	397	20	0	Pass
20	30	780	19	1	Fail

6 Conclusions

The objective of this deliverable was to develop solutions that are in line with the “Intelligent train” paradigm enabling the operation of the rolling stock in an autonomous mode (driverless trains). To achieve this, an off-line modelling framework based on Data Envelopment Analysis and an online based on Markov Decision Process that aims at identifying the optimal driving styles in terms of energy efficiency of an operational tramway system has been developed. A main limitation of these approaches is related to their increased computational complexity. To address this, DEA and MDP methods were coupled with machine learning techniques to reduce the complexity of the ILP formulations. An additional challenge was associated with the positioning of the rolling stock under both line of sight and non-line of sight scenarios. This problem has been resolved through the development of a positioning solution based on low cost LiFi technology assist in optimising the operation of the railway system. Preliminary results indicate that the proposed approach can reduce the energy consumption in railway systems by 10%.

7 References

[1] https://uic.org/support-activities/statistics/#UIC-statistical-indicators

[2] CYbersecurity in the RAILway sector, D2.1 – Safety and Security requirements of Rail transport system in multi-stakeholder environments, cyrail.eu

[3] Anastasopoulos, M., Tzanakaki A., Iordache, M., Langlois, O., Pheulpin, Jean-Francois, Simeonidou D. ICT platforms in support of future railway systems, In proc of TRA 2018, Vienna. 16-19 April 2018.

[4] Achilleos A., Anastasopoulos M., Tzanakaki, A., Iordache, M. Langlois, O., Pheulpin J.F, Simeonidou D., "Optimal Driving Profiles in Railway Systems based on Data Envelopment Analysis". VEHITS 2019: 254-259

[5] Ogasa M.: "Energy Saving and Environmental Measures in Railway Technologies:Example with Hybrid Electric Railway Vehicles"

[6] Gunselmann, W.: "Technologies for Increased Energy Efficiency in Railway Systems", European Conference on Power Electronics and Applications, Dresden 2005

[7] Gunselmann, W., Godbersen, Ch.: "Double-layer capacitors store surplus braking energy. In: Railway Gazette International", November 2001 S. 581 ff

[8] Gallo, M., Simonelli, F., De Luca, G., and V. De Martinis, "Estimating the effects of energy-efficient driving profiles on railway consumption," 2015 IEEE 15th International Conference on Environment and Electrical Engineering (EEEIC), Rome, 2015, pp. 813-818.

[9] P. Lukaszewicz, J. Allan, C. A. Brebbia, R. J. Hill, G. Sciutto & S. Sone, "Energy saving driving methods for freight trains", Div. of Railway Technology, KTH (Royal Inst. of Technology), Sweden Computers in Railways IX, © 2004 WIT Press, www.witpress.com, ISBN 1-85312-715-9

[10]Huiru Zhang, Limin Jia, Li Wang, Xinyue Xu. (2019) "Energy consumption optimization of train operation for railway systems: Algorithm development and real-world case study". Journal of Cleaner Production 214, pages 1024-1037.

[11]Scheepmaker, G. M., Goverde, R. M. P., & Kroon, L. G. (2017). "Review of energy-efficient train control and timetabling" European Journal of Operational Research, 257(2), pp. 355–376.

[12][13]C. and Sim, S. "Optimising train movements through coast control using genetic algorithms," IEE Proceedings on Electric Power Applications, 1997, vol. 144, pp. 65–73.

[13]Charnes, A., Cooper W., Rhodes. E., (1978) "Measuring the efficiency of decision making units", European Journal of Operational Research, vol. 2, issue 6, 429-444

[14]Cornuejols, G., Trick, M., (1998). Quantitative Methods for the Management Sciences, Course Notes, Chapter 12, "Data Envelopment Analysis, [Online] https://mat.gsia.cmu.edu/classes/QUANT/

[15]Huiru Zhang, Limin Jia, Li Wang, Xinyue Xu. (2019) "Energy consumption optimization of train operation for railway systems: Algorithm development and real-world case study". *Journal of Cleaner Production* 214, pages 1024-1037.

[16]Mensing, F., Trigui, R., Bideaux, E. "Vehicle trajectory optimization for application in ECO-driving", *Vehicle Power and Propulsion Conference (VPPC), 2011, pp. 1-6*.

[17]Powell, J.P. Palacín R. (2014) "A comparison of modelled and real-life driving profiles for the simulation of railway vehicle operation", *NewRail – Centre for Railway Research, Newcastle University, Newcastle upon Tyne, UK*

[18]H. Strössenreuther: "Energy Efficient Driving" -DB AG. 2nd UIC Railway

[19]Energy Efficiency Conference UIC, Paris 4-5 February 2004.

[20]P. Lukaszewicz: "Driving techniques and strategies for freigth trains". Computers in Railways VII. COMPRAIL 2000 Bologna.

[21]P. Lukaszewicz: "Energy Consumption and Running Time for Trains". KTH Stockholm 2001. TRITA-FKT 2001:25. ISSN1103-470X.

[22]Toledo T., Lotan T.: "In-Vehicle Data Recorder for Evaluation of Driving Behavior and Safety"

[23]Buehler, M., Iagnemma, K., & Singh, S. (2009). The DARPA urban challenge: Autonomous vehicles in city traffic (Vol. 56). Berlin, Heidelberg: Springer.
[24] C. Urmson, "A Robust Approach to High-Speed Navigation for Unrehearsed Desert Terrain", J. Field Robotics, vol. 23, no. 8, pp. 467-508, 2006.
[25]M. Montemerlo, J. Becker, S. Bhat, H. Dahlkamp, D. Dolgov, S. Ettinger, D. Haehnel et al., "Junior: The Stanford Entry in the Urban Challenge", Journal of Field Robotics, vol. 25, no. 9, pp. 569-597, 2008.
[26] M. Bojarski, D. Del Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L. D. Jackel, M. Monfort, U. Muller, J. Zhang et al., End to end learning for self-driving cars, 2016.
[27]S. Richard, G. B. Sutton and G. B. Andrew, Reinforcement Learning: An Introduction, London, England: The MIT Press, 2017.
[28]T. S. Ferguson, Linear Programming: A Concise Introduction.

Appendix

In2Dreams Beacon Detection Script
==================================
Test Date: 2019-09-05T16:34:40
The AP's are swithced from lifictl -p2 to lifictl -p1
Test Iteration with on off time interval set to 0.020
===
Test 1 expected 0:4:7d:ad:53:c9
Test 1 recorded 0:4:7d:36:76:41
Test 1 Test Status FAILED
Test 2 expected 0:4:7d:36:76:41
Test 2 recorded 0:4:7d:36:76:41
Test 2 Test Status PASSED
Test 3 expected 0:4:7d:ad:53:c9
Test 3 recorded 0:4:7d:ad:53:c9
Test 3 Test Status PASSED
Test 4 expected 0:4:7d:36:76:41
Test 4 recorded 0:4:7d:36:76:41
Test 4 Test Status PASSED
Test 5 expected 0:4:7d:ad:53:c9
Test 5 recorded 0:4:7d:ad:53:c9
Test 5 Test Status PASSED
Test 6 expected 0:4:7d:36:76:41
Test 6 recorded 0:4:7d:36:76:41
Test 6 Test Status PASSED
Test 7 expected 0:4:7d:ad:53:c9
Test 7 recorded 0:4:7d:ad:53:c9
Test 7 Test Status PASSED
Test 8 expected 0:4:7d:36:76:41
Test 8 recorded 0:4:7d:36:76:41
Test 8 Test Status PASSED
Test 9 expected 0:4:7d:ad:53:c9
Test 9 recorded 0:4:7d:ad:53:c9
Test 9 Test Status PASSED
Test 10 expected 0:4:7d:36:76:41
Test 10 recorded 0:4:7d:36:76:41
Test 10 Test Status PASSED
Test 11 expected 0:4:7d:ad:53:c9
Test 11 recorded 0:4:7d:ad:53:c9
Test 11 Test Status PASSED
Test 12 expected 0:4:7d:36:76:41
Test 12 recorded 0:4:7d:36:76:41
Test 12 Test Status PASSED
Test 13 expected 0:4:7d:ad:53:c9
Test 13 recorded 0:4:7d:ad:53:c9
Test 13 Test Status PASSED
Test 14 expected 0:4:7d:36:76:41
Test 14 recorded 0:4:7d:36:76:41
Test 14 Test Status PASSED
Test 15 expected 0:4:7d:ad:53:c9
Test 15 recorded 0:4:7d:ad:53:c9
Test 15 Test Status PASSED

Test 16 expected 0:4:7d:36:76:41
Test 16 recorded 0:4:7d:36:76:41
Test 16 Test Status PASSED
Test 17 expected 0:4:7d:ad:53:c9
Test 17 recorded 0:4:7d:ad:53:c9
Test 17 Test Status PASSED
Test 18 expected 0:4:7d:36:76:41
Test 18 recorded 0:4:7d:36:76:41
Test 18 Test Status PASSED
Test 19 expected 0:4:7d:ad:53:c9
Test 19 recorded 0:4:7d:ad:53:c9
Test 19 Test Status PASSED
Test 20 expected 0:4:7d:36:76:41
Test 20 recorded 0:4:7d:36:76:41
Test 20 Test Status PASSED
The Requested On Off Interval 0.020 equates to an actual interval of 30ms

Test Iteration with on off time interval set to 0.021
==

Test 1 expected 0:4:7d:ad:53:c9
Test 1 recorded 0:4:7d:36:76:41
Test 1 Test Status FAILED
Test 2 expected 0:4:7d:36:76:41
Test 2 recorded 0:4:7d:36:76:41
Test 2 Test Status PASSED
Test 3 expected 0:4:7d:ad:53:c9
Test 3 recorded 0:4:7d:ad:53:c9
Test 3 Test Status PASSED
Test 4 expected 0:4:7d:36:76:41
Test 4 recorded 0:4:7d:36:76:41
Test 4 Test Status PASSED
Test 5 expected 0:4:7d:ad:53:c9
Test 5 recorded 0:4:7d:ad:53:c9
Test 5 Test Status PASSED
Test 6 expected 0:4:7d:36:76:41
Test 6 recorded 0:4:7d:36:76:41
Test 6 Test Status PASSED
Test 7 expected 0:4:7d:ad:53:c9
Test 7 recorded 0:4:7d:ad:53:c9
Test 7 Test Status PASSED
Test 8 expected 0:4:7d:36:76:41
Test 8 recorded 0:4:7d:36:76:41
Test 8 Test Status PASSED
Test 9 expected 0:4:7d:ad:53:c9
Test 9 recorded 0:4:7d:ad:53:c9
Test 9 Test Status PASSED
Test 10 expected 0:4:7d:36:76:41
Test 10 recorded 0:4:7d:36:76:41
Test 10 Test Status PASSED
Test 11 expected 0:4:7d:ad:53:c9
Test 11 recorded 0:4:7d:ad:53:c9
Test 11 Test Status PASSED
Test 12 expected 0:4:7d:36:76:41
Test 12 recorded 0:4:7d:36:76:41

Test 12 Test Status PASSED
Test 13 expected 0:4:7d:ad:53:c9
Test 13 recorded 0:4:7d:ad:53:c9
Test 13 Test Status PASSED
Test 14 expected 0:4:7d:36:76:41
Test 14 recorded 0:4:7d:36:76:41
Test 14 Test Status PASSED
Test 15 expected 0:4:7d:ad:53:c9
Test 15 recorded 0:4:7d:ad:53:c9
Test 15 Test Status PASSED
Test 16 expected 0:4:7d:36:76:41
Test 16 recorded 0:4:7d:36:76:41
Test 16 Test Status PASSED
Test 17 expected 0:4:7d:ad:53:c9
Test 17 recorded 0:4:7d:ad:53:c9
Test 17 Test Status PASSED
Test 18 expected 0:4:7d:36:76:41
Test 18 recorded 0:4:7d:36:76:41
Test 18 Test Status PASSED
Test 19 expected 0:4:7d:ad:53:c9
Test 19 recorded 0:4:7d:ad:53:c9
Test 19 Test Status PASSED
Test 20 expected 0:4:7d:36:76:41
Test 20 recorded 0:4:7d:36:76:41
Test 20 Test Status PASSED
The Requested On Off Interval 0.021 equates to an actual interval of 30ms

Test Iteration with on off time interval set to 0.022
==

Test 1 expected 0:4:7d:ad:53:c9
Test 1 recorded 0:4:7d:ad:53:c9
Test 1 Test Status PASSED
Test 2 expected 0:4:7d:36:76:41
Test 2 recorded 0:4:7d:36:76:41
Test 2 Test Status PASSED
Test 3 expected 0:4:7d:ad:53:c9
Test 3 recorded 0:4:7d:ad:53:c9
Test 3 Test Status PASSED
Test 4 expected 0:4:7d:36:76:41
Test 4 recorded 0:4:7d:36:76:41
Test 4 Test Status PASSED
Test 5 expected 0:4:7d:ad:53:c9
Test 5 recorded 0:4:7d:ad:53:c9
Test 5 Test Status PASSED
Test 6 expected 0:4:7d:36:76:41
Test 6 recorded 0:4:7d:36:76:41
Test 6 Test Status PASSED
Test 7 expected 0:4:7d:ad:53:c9
Test 7 recorded 0:4:7d:ad:53:c9
Test 7 Test Status PASSED
Test 8 expected 0:4:7d:36:76:41
Test 8 recorded 0:4:7d:36:76:41
Test 8 Test Status PASSED
Test 9 expected 0:4:7d:ad:53:c9

Test 9 recorded 0:4:7d:ad:53:c9
Test 9 Test Status PASSED
Test 10 expected 0:4:7d:36:76:41
Test 10 recorded 0:4:7d:36:76:41
Test 10 Test Status PASSED
Test 11 expected 0:4:7d:ad:53:c9
Test 11 recorded 0:4:7d:ad:53:c9
Test 11 Test Status PASSED
Test 12 expected 0:4:7d:36:76:41
Test 12 recorded 0:4:7d:36:76:41
Test 12 Test Status PASSED
Test 13 expected 0:4:7d:ad:53:c9
Test 13 recorded 0:4:7d:ad:53:c9
Test 13 Test Status PASSED
Test 14 expected 0:4:7d:36:76:41
Test 14 recorded 0:4:7d:36:76:41
Test 14 Test Status PASSED
Test 15 expected 0:4:7d:ad:53:c9
Test 15 recorded 0:4:7d:ad:53:c9
Test 15 Test Status PASSED
Test 16 expected 0:4:7d:36:76:41
Test 16 recorded 0:4:7d:36:76:41
Test 16 Test Status PASSED
Test 17 expected 0:4:7d:ad:53:c9
Test 17 recorded 0:4:7d:ad:53:c9
Test 17 Test Status PASSED
Test 18 expected 0:4:7d:36:76:41
Test 18 recorded 0:4:7d:36:76:41
Test 18 Test Status PASSED
Test 19 expected 0:4:7d:ad:53:c9
Test 19 recorded 0:4:7d:ad:53:c9
Test 19 Test Status PASSED
Test 20 expected 0:4:7d:36:76:41
Test 20 recorded 0:4:7d:36:76:41
Test 20 Test Status PASSED
The Requested On Off Interval 0.022 equates to an actual interval of 58.8ms

Test Iteration with on off time interval set to 0.023
==

Test 1 expected 0:4:7d:ad:53:c9
Test 1 recorded 0:4:7d:ad:53:c9
Test 1 Test Status PASSED
Test 2 expected 0:4:7d:36:76:41
Test 2 recorded 0:4:7d:36:76:41
Test 2 Test Status PASSED
Test 3 expected 0:4:7d:ad:53:c9
Test 3 recorded 0:4:7d:ad:53:c9
Test 3 Test Status PASSED
Test 4 expected 0:4:7d:36:76:41
Test 4 recorded 0:4:7d:36:76:41
Test 4 Test Status PASSED
Test 5 expected 0:4:7d:ad:53:c9
Test 5 recorded 0:4:7d:ad:53:c9
Test 5 Test Status PASSED

Test 6 expected 0:4:7d:36:76:41
Test 6 recorded 0:4:7d:36:76:41
Test 6 Test Status PASSED
Test 7 expected 0:4:7d:ad:53:c9
Test 7 recorded 0:4:7d:ad:53:c9
Test 7 Test Status PASSED
Test 8 expected 0:4:7d:36:76:41
Test 8 recorded 0:4:7d:36:76:41
Test 8 Test Status PASSED
Test 9 expected 0:4:7d:ad:53:c9
Test 9 recorded 0:4:7d:ad:53:c9
Test 9 Test Status PASSED
Test 10 expected 0:4:7d:36:76:41
Test 10 recorded 0:4:7d:36:76:41
Test 10 Test Status PASSED
Test 11 expected 0:4:7d:ad:53:c9
Test 11 recorded 0:4:7d:ad:53:c9
Test 11 Test Status PASSED
Test 12 expected 0:4:7d:36:76:41
Test 12 recorded 0:4:7d:36:76:41
Test 12 Test Status PASSED
Test 13 expected 0:4:7d:ad:53:c9
Test 13 recorded 0:4:7d:ad:53:c9
Test 13 Test Status PASSED
Test 14 expected 0:4:7d:36:76:41
Test 14 recorded 0:4:7d:36:76:41
Test 14 Test Status PASSED
Test 15 expected 0:4:7d:ad:53:c9
Test 15 recorded 0:4:7d:ad:53:c9
Test 15 Test Status PASSED
Test 16 expected 0:4:7d:36:76:41
Test 16 recorded 0:4:7d:36:76:41
Test 16 Test Status PASSED
Test 17 expected 0:4:7d:ad:53:c9
Test 17 recorded 0:4:7d:ad:53:c9
Test 17 Test Status PASSED
Test 18 expected 0:4:7d:36:76:41
Test 18 recorded 0:4:7d:36:76:41
Test 18 Test Status PASSED
Test 19 expected 0:4:7d:ad:53:c9
Test 19 recorded 0:4:7d:ad:53:c9
Test 19 Test Status PASSED
Test 20 expected 0:4:7d:36:76:41
Test 20 recorded 0:4:7d:36:76:41
Test 20 Test Status PASSED
The Requested On Off Interval 0.023 equates to an actual interval of 59ms

Test Iteration with on off time interval set to 0.024
==

Test 1 expected 0:4:7d:ad:53:c9
Test 1 recorded 0:4:7d:ad:53:c9
Test 1 Test Status PASSED
Test 2 expected 0:4:7d:36:76:41
Test 2 recorded 0:4:7d:36:76:41

Test 2 Test Status PASSED
Test 3 expected 0:4:7d:ad:53:c9
Test 3 recorded 0:4:7d:ad:53:c9
Test 3 Test Status PASSED
Test 4 expected 0:4:7d:36:76:41
Test 4 recorded 0:4:7d:36:76:41
Test 4 Test Status PASSED
Test 5 expected 0:4:7d:ad:53:c9
Test 5 recorded 0:4:7d:ad:53:c9
Test 5 Test Status PASSED
Test 6 expected 0:4:7d:36:76:41
Test 6 recorded 0:4:7d:36:76:41
Test 6 Test Status PASSED
Test 7 expected 0:4:7d:ad:53:c9
Test 7 recorded 0:4:7d:ad:53:c9
Test 7 Test Status PASSED
Test 8 expected 0:4:7d:36:76:41
Test 8 recorded 0:4:7d:36:76:41
Test 8 Test Status PASSED
Test 9 expected 0:4:7d:ad:53:c9
Test 9 recorded 0:4:7d:ad:53:c9
Test 9 Test Status PASSED
Test 10 expected 0:4:7d:36:76:41
Test 10 recorded 0:4:7d:36:76:41
Test 10 Test Status PASSED
Test 11 expected 0:4:7d:ad:53:c9
Test 11 recorded 0:4:7d:ad:53:c9
Test 11 Test Status PASSED
Test 12 expected 0:4:7d:36:76:41
Test 12 recorded 0:4:7d:36:76:41
Test 12 Test Status PASSED
Test 13 expected 0:4:7d:ad:53:c9
Test 13 recorded 0:4:7d:ad:53:c9
Test 13 Test Status PASSED
Test 14 expected 0:4:7d:36:76:41
Test 14 recorded 0:4:7d:36:76:41
Test 14 Test Status PASSED
Test 15 expected 0:4:7d:ad:53:c9
Test 15 recorded 0:4:7d:ad:53:c9
Test 15 Test Status PASSED
Test 16 expected 0:4:7d:36:76:41
Test 16 recorded 0:4:7d:36:76:41
Test 16 Test Status PASSED
Test 17 expected 0:4:7d:ad:53:c9
Test 17 recorded 0:4:7d:ad:53:c9
Test 17 Test Status PASSED
Test 18 expected 0:4:7d:36:76:41
Test 18 recorded 0:4:7d:36:76:41
Test 18 Test Status PASSED
Test 19 expected 0:4:7d:ad:53:c9
Test 19 recorded 0:4:7d:ad:53:c9
Test 19 Test Status PASSED
Test 20 expected 0:4:7d:36:76:41
Test 20 recorded 0:4:7d:36:76:41
Test 20 Test Status PASSED

The Requested On Off Interval 0.024 equates to an actual interval of 59.4ms

Test Iteration with on off time interval set to 0.025
===

Test 1 expected 0:4:7d:ad:53:c9
Test 1 recorded 0:4:7d:ad:53:c9
Test 1 Test Status PASSED
Test 2 expected 0:4:7d:36:76:41
Test 2 recorded 0:4:7d:36:76:41
Test 2 Test Status PASSED
Test 3 expected 0:4:7d:ad:53:c9
Test 3 recorded 0:4:7d:ad:53:c9
Test 3 Test Status PASSED
Test 4 expected 0:4:7d:36:76:41
Test 4 recorded 0:4:7d:36:76:41
Test 4 Test Status PASSED
Test 5 expected 0:4:7d:ad:53:c9
Test 5 recorded 0:4:7d:ad:53:c9
Test 5 Test Status PASSED
Test 6 expected 0:4:7d:36:76:41
Test 6 recorded 0:4:7d:36:76:41
Test 6 Test Status PASSED
Test 7 expected 0:4:7d:ad:53:c9
Test 7 recorded 0:4:7d:ad:53:c9
Test 7 Test Status PASSED
Test 8 expected 0:4:7d:36:76:41
Test 8 recorded 0:4:7d:36:76:41
Test 8 Test Status PASSED
Test 9 expected 0:4:7d:ad:53:c9
Test 9 recorded 0:4:7d:ad:53:c9
Test 9 Test Status PASSED
Test 10 expected 0:4:7d:36:76:41
Test 10 recorded 0:4:7d:36:76:41
Test 10 Test Status PASSED
Test 11 expected 0:4:7d:ad:53:c9
Test 11 recorded 0:4:7d:ad:53:c9
Test 11 Test Status PASSED
Test 12 expected 0:4:7d:36:76:41
Test 12 recorded 0:4:7d:36:76:41
Test 12 Test Status PASSED
Test 13 expected 0:4:7d:ad:53:c9
Test 13 recorded 0:4:7d:ad:53:c9
Test 13 Test Status PASSED
Test 14 expected 0:4:7d:36:76:41
Test 14 recorded 0:4:7d:36:76:41
Test 14 Test Status PASSED
Test 15 expected 0:4:7d:ad:53:c9
Test 15 recorded 0:4:7d:ad:53:c9
Test 15 Test Status PASSED
Test 16 expected 0:4:7d:36:76:41
Test 16 recorded 0:4:7d:36:76:41
Test 16 Test Status PASSED
Test 17 expected 0:4:7d:ad:53:c9
Test 17 recorded 0:4:7d:ad:53:c9

Test 17 Test Status PASSED
Test 18 expected 0:4:7d:36:76:41
Test 18 recorded 0:4:7d:36:76:41
Test 18 Test Status PASSED
Test 19 expected 0:4:7d:ad:53:c9
Test 19 recorded 0:4:7d:ad:53:c9
Test 19 Test Status PASSED
Test 20 expected 0:4:7d:36:76:41
Test 20 recorded 0:4:7d:36:76:41
Test 20 Test Status PASSED
The Interval 0.025 equates to an actual interval of 59ms