Regulating the Searching Behaviour of Parked Vehicles Attempting to Locate Moving, Missing Entities

Matheus Souza, Wynita M. Griggs, Jakub Mareček and Robert N. Shorten

Abstract— **In this work, feedback regulation is added to a system that otherwise consists of a large, high-density network of parked vehicles. When awoken by an administrative centre, this network proceeds to search for moving, missing entities of interest using RFID-based techniques. RIFD readers and antennae are placed within the vehicles, while RFID passive tags are carried on the entity of interest via some means, e.g., a wrist band. Specifically, we seek to regulate the number and geographical distribution of parked vehicles that are "Switched On", and thus actively searching for the moving entity of interest. In doing so, we seek to conserve vehicular energy consumption while, at the same time, maintain good geographical coverage of the city such that the moving entity of interest is likely to be located within an acceptable time frame. Which vehicles are "Switched On" at any point in time is a matter determined periodically through the use of stochastic techniques. The regulated system is demonstrated through the use case of a missing Alzheimer's patient in inner-city Dublin, Ireland.**

I. INTRODUCTION

When a material object, pet, or loved ones go missing, it can be a stressful experience for all involved. Considering objects, pets and even people go missing every day, there are methods and systems to facilitate the location of missing entities. For example, we microchip our pets or equip them with cellular-based GPS collars, and applications on our computers allow us to track missing or stolen smartphones. Medical jewellery and community support networks exist to aid people with needs who wander, including Alzheimer's patients [1]. Meanwhile, general advances in information and communication technology, coupled with the emergence of the Internet of Things (IoT), means that these methods increasingly allow for automation of the search and therewith, improved response times.

For instance, in the context of IoT, the vehicles that we drive are becoming connected to each other, to the infrastructure, as well as to the internet [2]. With expanding on-board sensor complements, computing, and communication abilities, parked cars no longer need to be idle, to be of no service to us during the extended periods when they are not being driven. Recently [3]–[5], the use of networks of parked vehicles in dense urban areas has been suggested for the detection and localisation of moving, missing entities using RFID technology.

The RFID-based system, described in [3]–[5] and illustrated in Fig. 1, was envisioned as follows. Each participating parked vehicle has an RFID reader and antenna on board, and is able to communicate with an administrative centre. The missing entity is presumed to be carrying an RFID passive tag via some means, e.g., a wrist band. Passive RFID tags do not require a local power source, beyond the field created by the RFID reader, and thus need not contain batteries. When an entity is missing, an alarm is raised with the administrative centre. For example, the entity's carer or owner places a phone call with the police. Once the alarm has been raised, the administrative centre prompts the RFID-based application on board the parked vehicles participating in the service. The RFID technology enables those vehicles to attempt to locate the missing entity, and to inform the administrative centre when the missing entity is found, i.e., when the RFID equipment on board a parked vehicle detects and processes the presence of the unique RFID passive tag carried by the missing entity. The information sent to the administrative centre might include a time stamp, a GPS location of the parked vehicle, and the unique RFID passive tag ID carried by the missing entity that was detected by the equipment on board the parked vehicle. Once detected, the administrative centre is then able to invoke a procedure aimed at making contact with the missing entity. For example, police are able

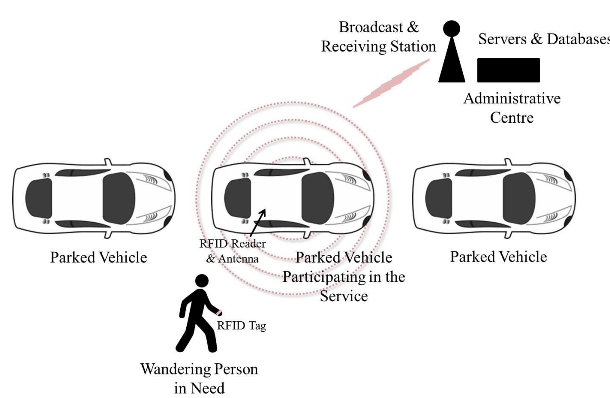

Fig. 1. RFID-based system illustration. (Some sub-images obtained from Openclipart [6], [7].)

This work was supported in part by the Science Foundation Ireland under Grant 16/IA/4610 and was co-funded under the European Regional Development Fund through the Southern & Eastern Regional Operational Programme to Lero - the Irish Software Research Centre. The work was also partially supported by Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP/Brazil) under Grant 2016/19504-7 and received funding from the European Union Horizon 2020 Programme (Horizon2020/2014-2020) under Grant 688380.

M. Souza is with the School of Electrical and Computer Engineering, University of Campinas, Brazil. msouza@fee.unicamp.br

W. M. Griggs and R. N. Shorten are with the School of Electrical and Electronic Engineering, University College Dublin, Belfield, Dublin 4, Ireland. wynita.griggs@ucd.ie, robert.shorten@ucd.ie

J. Marecek is with IBM Research – Ireland, IBM Dublin Technology Campus, Building 3, Damastown Industrial Estate, Mulhuddart, Dublin 15, Ireland. jakub.marecek@ie.ibm.com

to go to the location at which the entity was detected in order to refine the localisation and determine whether the entity needs assistance, and if required, aid the entity on its way home. See [3]–[5] for further details.

The work presented in [5] was largely simulation-based. The system was demonstrated through a use case scenario of a missing Alzheimer's patient in inner-city Dublin, Ireland. System parameters were varied, including: (i) the percentage of parking spaces on the Dublin map that were inhabited by vehicles participating in the service; (ii) the polling rate of the RFID equipment on board the participating parked vehicles; and (iii) the RFID equipment's detection range. Results were presented from thousands of simulations and consisted of: (a) average times that it took for the network of participating parked vehicles to detect the moving pedestrian; (b) population standard deviations from these average detection times; and (c) the number of times that the system failed to detect the pedestrian within a thirty-minute time frame. An interesting (albeit expected) observation that the results revealed was one of *redundancy*, in that the average detection times, and particularly the "failed to detect" totals, followed curves resembling the exponential. That is, the average detection times and "failed to detect" results remained relatively constant until a "threshold" participation percentage was reached. When numbers of parking spaces inhabited by searching vehicles fell below this threshold, detection times, and especially the "failed to detect" totals, increased sharply.

Clearly, a key question is: How can we distribute the searching agents to quickly locate the moving, missing entity, while also reducing redundancy in the system? There are a number of ways in which this problem can be approached, but one should keep in mind that the best search strategy can be formulated as the restless bandit problem, whose approximation to any non-trivial factor is complete for polynomial-space Turing machines [8], i.e., provably intractable. We hence propose to focus on fairness among participating vehicles in terms of energy consumption.

In this paper, we choose to put a feedback loop around our urban centre with the aim of regulating the number of cars looking for a missing, moving entity efficiently. Consider the following notion: the administrative centre broadcasts a signal to all vehicles capable of participating in our service. Then, each vehicle sends out a "ping" to determine how many neighbours they have that are also capable of participating. Probabilistic models of each vehicle switching on or off their RFID readers are associated with the numbers of neighbours. The agent uses the broadcast signal from the administration centre, together with the relevant probability model deduced by the number of his or her neighbours, to "flip a coin" and determine whether to "Switch On" their RFID reader over the next time interval. This process is repeated every time interval. In the next section, we provide some control-theoretic background for this approach. Then, we formalise the problem and present our algorithms. In Section V, we demonstrate the feedback regulation in action by revisiting the use case of a missing Alzheimer's patient in inner-city Dublin. Finally, conclusions and future work are presented in Section VI.

II. RELATED WORK

In this section, we explore related work on the interface of smart cities and control theory, which elucidates main mathematical features of the problem. Most of the theory discussed in this section is presented in [9], [10], who have introduced an abstract framework, blending practical aspects of intelligent transportation systems, smart cities, and techniques from classical control theory.

Let us consider a resource allocation problem in discrete time. In particular, consider the closed-loop system as depicted in Fig. 2, which comprises a (typically large) number of agents, a controller, and a filter. The controller, $\mathcal{C}$, broadcasts a signal $\pi(k)$ at time $k \in \mathbb{N}$; the $N \in \mathbb{N}$ agents $\mathcal{S}_1, \ldots, \mathcal{S}_N$ amend their use of a shared resource in response. The use $x_i(k)$ of the resource by agent i at time k is modelled as a random variable, as there is an inherent randomness in the reaction of each agent to the broadcast signal. The main design task is to *regulate* the aggregate resource utilisation

$$y(k) = \sum_{i=1}^{N} x_i(k), \tag{1}$$

which is also a random variable. In this setting, the controller usually does not have access to either x_i or y, but only to an estimate $\hat{y}$ of y, which is the output of a filter $\mathcal{F}$.

In addition to achieving regulation, the controller should also ensure that the agents have a sense of fairness and predictability. In control-theoretic terms, this can be cast as a particular flavour of ergodicity of the closed-loop system dynamics, known as the existence of a unique invariant measure [9], [10]. This completely removes effects of initial conditions on the long run.

The behaviour of the parked cars can be modelled in a number of ways. For simplicity, let us consider a model, where the state $x_i(k)$ of agent i at time k is in the set $\{0, 1\}$, as these variables model whether agent i allows for the search ($x_i(k) = 1$) or not ($x_i(k) = 0$). We also assume that both the controller $\mathcal{C} : e \mapsto \pi$ and the filter $\mathcal{F} : y \mapsto \hat{y}$ are linear and time-invariant dynamic systems. In particular, we shall explore, in the examples, the simple controller model given

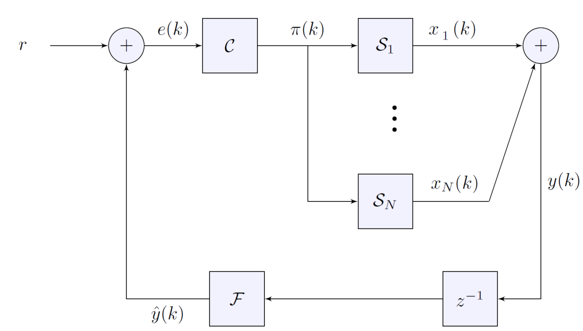

Fig. 2. A feedback model employed.

by the difference equation

$$\pi(k) = \beta\pi(k-1) + \kappa\left[e(k) - \alpha e(k-1)\right], \qquad (2)$$

for all $k \in \mathbb{N}$, in which $\alpha, \beta, \kappa \in \mathbb{R}$. This model includes, as particular cases, classical lead, lag and PI controller structures [11], [12]. A simple model that can be considered for the linear filter is the moving-average scheme

$$\hat{y}(k) = \frac{y(k) + y(k-1) + \cdots + y(k-m)}{m+1}, \qquad (3)$$

for some $m \in \mathbb{N}$.

Let us now describe the random behaviour presented by the agents considered in this paper. We assume that, at each time instant k, agent i has a probability p_{i1} of being on and a probability p_{i0} of being off at the following time instant. Both probabilities depend on the broadcast control signal π; that is,

$$\mathbb{P}(x_i(k+1) = 1) = p_{i1}\left(\pi(k)\right) \qquad (4)$$

and, thus,

$$\mathbb{P}(x_i(k+1) = 0) = p_{i0}\left(\pi(k)\right) = 1 - p_{i1}\left(\pi(k)\right), \qquad (5)$$

since both events are complementary. These probability functions must satisfy some assumptions, as we shall see in the sequel. One of these such assumptions is that p_{i1} and p_{i0} are bounded away from zero (and one), for all i and all π. The lack of this assumption can yield non-ergodic stochastic processes, since some agents may monopolise allocated resources. We are now able to state the following result, which is central for the theoretical framework we are considering.

Theorem 1 ([9]): Consider the feedback system depicted in Fig. 2, for some given finite-dimensional linear systems $\mathcal{C}$ and $\mathcal{F}$. Assume that each agent $i \in \{1, \ldots, N\}$ has state $x_i(k)$ governed by the following affine stochastic difference equation:

$$x_i(k+1) = w_{ij}\left(x_i(k)\right), \qquad (6)$$

where the affine mapping w_{ij} is chosen at each step of time according to a Dini-continuous probability function $p_{ij}(x_i(k), \pi(k))$, out of $w_{ij}(x_i) := A_i x_i + b_{ij}$, where A_i is a Schur matrix and for all i, $\pi(k)$, $\sum_j p_{ij}(x_i(k), \pi(k)) = 1$. In addition, suppose that there exist scalars $\delta_i > 0$ such that $p_{ij}(x_i, \pi) \geq \delta_i > 0$; that is, the probabilities are bounded away from zero. Then, for every stable linear controller $\mathcal{C}$ and every stable linear filter $\mathcal{F}$, the feedback loop converges in distribution to a unique invariant measure.

Note that this theorem addresses a more general dynamic model for the agents than the one we consider here. Indeed, for our case, we can take $A_i = 0$ and define $b_{i0} = 0$ and $b_{i1} = 1$ for all i. Note also that, from this result, it follows that our goal is to devise a stable, stabilising controller for the closed-loop system depicted in Fig. 2. This, together with stable filter dynamics, ensures ergodicity and, thus, fairness.

Some final remarks on this theorem are in order. First, Dini's condition on the probabilities may, obviously, be replaced by simpler, more conservative assumptions, such as Lipschitz or Hölder conditions [13]. Second, as we discussed previously, the requirement $p_{ij}(x_i, \pi) \geq \delta_i > 0$ in the theorem statement is necessary for ergodicity.

III. PROBLEM STATEMENT

As stated before, the main problem to be solved in this paper is the following:

> Regulate the number of cars looking for a missing, moving entity in an efficient way.

By efficiency, we mean that a solution to this problem must:

- Be **energy-efficient** compared to existent solutions that turn all cars on while the search is being conducted;
- Be **coverage-efficient**, i.e., the solution should encourage cars to spread through the city and should also orchestrate them to ensure that the city as a whole has a good coverage.

Our solution is embedded in the feedback theory for smart cities framework presented in the previous section and, as we shall see in the simulations, achieves both efficiency goals. In our solution, each parked car is assumed to be eligible for turning on a search sensor; therefore each car is an agent in our framework.

The controller regulates the number of simultaneously on cars around a pre-specified number r using the broadcast signal π, which affects the agents behaviour towards turning on or off. Broadly speaking, if the error signal $e = r - \hat{y}$ is large, then we expect a large value of π; their probabilities of turning on must be tuned so that large values of π induce more cars to turn on. The contrary effect should hold if $e = r - \hat{y}$ gets negative; that is, π should get negative and this should induce more cars to turn off.

The response of agent i to the broadcast signal π, namely its probabilities p_{i1} and p_{i0}, also plays a key role in the design. In addition to their explicit dependence on π, as discussed previously, these probabilities also depend on each agent's surroundings. Given that in our application, agent i represents a parked car whose probability of turning sensors on for the next time step is p_{i1}, this probability must clearly depend on the number of neighbouring vehicles. Indeed, clusters of cars can cooperate and take turns to cover one area whereas a sole car on a street must be almost always on. Hence, without any loss of generality, we consider three kinds of responses to the broadcast signal π, depending on whether a car has few (f_f), some/medium (f_s), or many (f_m) neighbours, as depicted in Fig. 3. Remember that these probabilities must satisfy the conditions of Theorem 1.

IV. AN ALGORITHM AND ITS IMPLEMENTATION

We are now able to present the main algorithm devised in this paper in Algorithm 1, and to showcase how it searches for missing entities.

To demonstrate the performance of our algorithm, we employed Simulation of Urban MObility (SUMO) Version 0.31.0. SUMO [14] is an open-source, microscopic traffic simulation package primarily being developed at the Institute

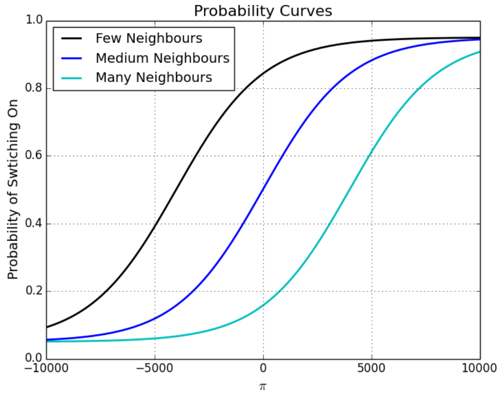

Fig. 3. Logistic functions used for probability models.

Algorithm 1: Main Algorithm

Data: Number of agents N; time step h; search time frame T; probability behaviours f_f, f_s and f_m; controller $\mathcal{C}$ and filter $\mathcal{F}$.
Result: Missing entity location **or** fail alert.
Initialise $k \leftarrow 0$; $\pi(0) \leftarrow 0$; $x_i(0) \leftarrow 0$; $\hat{y}(0) \leftarrow 0$;
for *each car* i **do**
  Determine the number of neighbouring cars;
  Decide whether N_i corresponds to *few*, *some* or *many* neighbours;
  Set p_{i1} as the corresponding probability behaviour;
end
while $k \cdot h \leq T$ **do**
  for *each car* i **do**
    if $x_i(k) = 1$ **then**
      Scan for missing entity and **return** its position if located;
    end
    'Toss a coin' and decide $x_i(k+1)$, according to (4);
  end
  $k \leftarrow k + 1$;
  Update $\hat{y}(k)$, $e(k)$, $\pi(k)$;
end
The agents have **failed** to locate the entity; **Return** an alert;

of Transportation Systems at the German Aerospace Centre (DLR). SUMO is designed to handle large networks, and comes with a "remote control" interface, TraCI (short for Traffic Control Interface) [15], which allows one to adapt the simulation and to control singular vehicles and pedestrians on the fly. Our goal was to simulate a pedestrian walking about in an urban scenario, and to regulate the number of parked vehicles actively searching for the pedestrian in an energy- and coverage-efficient manner using our algorithm.

A. A Setup of the Simulations

For our urban map, we considered the road, pedestrian sidewalk, and footpath networks in the region covered by the Dublin Parking Yellow Zone, in Dublin, Ireland [16]. See also Fig. 4. The Dublin Parking Yellow Zone is located in the centre of Dublin, and thus represents an area of very high demand for parking spaces. For our complete vehicle parking space set, we considered all of the on-street parking administered by Dublin's Pay-and-Display machines in the Dublin Parking Yellow Zone, as well as all of the public disabled parking spaces within this area.

To generate on the fly random walks for our pedestrian, we utilised the python script containing an algorithm that we had designed in [5]. That is, for each simulation, initially, the person begins on a random edge. The algorithm then continues by creating a list of neighbouring "next" edges, with respect to the current edge that the person is on, and randomly picks one of these "next" edges to be the next link on the person's route. For simplicity, we disallowed the person from performing U-turns, and set a maximum walking speed for the person at 1.25m/s. We used SUMO's *nonInteracting* pedestrian model [17] as the model for how the person otherwise interacted with our map.

In regard to the vehicle parking spaces, as we did in [5], we utilised Google Maps' satellite imagery to visually locate the approximate locations of all of the on-street parking spaces administered by Dublin's Pay-and-Display machines in the Dublin Parking Yellow Zone, as well as all of the public disabled parking spaces within this region. In total, 8,736 parking spaces were mapped from the Google Maps' satellite imagery onto our SUMO network. We represented all of these parking spaces as Points of Interest on our SUMO network. We specified the dimensions of each parking space as 5m $\times$ 2.5m for simplicity, using [18] as a guide on recommended parking space size, and otherwise used the satellite imagery to aid us in determining whether each parking space was to be placed in parallel or perpendicular to the curb.

Another parameter in our experiment was the proportion of parking spaces that would have cars parked in them that

Fig. 4. Dublin's Yellow Zone. (Imported for use in SUMO from OpenStreetMap.)

were capable of participating in our service. We elected to simulate a test case scenario where each of the 8,736 parking spaces had a 50% chance of being inhabited by a vehicle capable of participating in our service. At the beginning of each simulation, a "weighted coin' was flipped for each of the 8,736 parking spaces. The result of this "coin flip" was compared to the above desired percentage value, to determine whether that parking space would be inhabited by a parked vehicle capable participating in the service or not, over that particular simulation. Parking assignments then remained constant for the duration of a simulation, and parked, participating vehicles were "Switched On" or "Switched Off" according to Algorithm 1. At the beginning of each simulation, no participating vehicles were "Switched On". We chose our target number r of "Switched On" vehicles to be 2,000.

For our probability models, we employed the use of logistic functions which are illustrated in Fig. 3. We placed a circle with a radius of twenty metres around each parked vehicle capable of participating in our service, and let the number of other parked vehicles (capable of participating in our service, and) residing within this circle, equate to the number of neighbours that the vehicle at the centre of the circle had. For simplicity, we set $m = 0$ for our filter described by (3); and we let $\alpha = -4.01$, $\beta = 0.99$ and $\kappa = 0.1$ in (2). We set each vehicle's RFID polling rate (i.e. the frequency at which a car's RFID system is sampling at when the vehicle is "Switched On") as "Always On", meaning that once "Switched On", a vehicle is always polling as opposed to doing periodic, timed reads. We set a circular RFID field around each car with a radius of six metres. Moreover, we assumed that once a pedestrian entered this field, and if the vehicle was "Switched On", then the pedestrian would be detected. In other words, in this paper, we neglect some of the more complicated phenomena typically associated with RFID, such as the effects of tag placement, antenna orientation, cable length, reader settings, and environmental factors such as the existence of water or other radio waves [19].

For each simulation, then, our goal was to set the person down on a random edge, and have them walk until either: (i) they were detected by a parked vehicle that was "Switched On" and thus actively searching at the same time as when the pedestrian was passing by; or (ii) thirty minutes had lapsed and no detection event had occurred. The beginning of each simulation was intended to mimic the moment that the service application on board any parked vehicles capable of participating in our service was activated by an administrative centre, e.g., just after an alert has been raised by a carer to the police that a person in need and carrying a unique RFID tag was missing in the area. We permitted thirty minutes to lapse before a "fail-to-detect" event was recorded, keeping in mind that quickly finding a missing and potentially stressed person, and returning them to their home, for instance, is ideal. All simulations had time-step updates of 1s, while our control signals were sent only every 20s. For our test case scenario, 100 simulations were performed, in total.

V. RESULTS

To gather some preliminary data, we first permitted a single simulation to run for a full thirty minutes, i.e., with no pedestrian placement yet. From this simulation, Fig. 5 demonstrates that regulation of the system, such that approximately 2,000 parked vehicles were "Switched On" at any point in time, was achieved quite rapidly. Fig. 6 illustrates the evolution of the control signal π over time. Notice that π could then be used in association with Fig. 3, along with the known number of neighbours that a vehicle had, to determine the probability of that vehicle being "Switched On" over the next time step $k{\cdot}h$. Similarly, Fig. 7 shows how the evolution of the error signal e over time. Finally, Fig. 8 is a snapshot of part of our SUMO network, illustrating the geographical distribution of parked vehicles at the end of the simulation. In the snapshot, white circles represent empty parking spaces, purple circles represent parked vehicles capable of participating in the service, but currently "Switched Off", and blue circles represent currently "Switched On" vehicles. The white circles remained white throughout the entire simulation by design as we were considering static parking, but the purple and blue circles changed colour depending on the output of the control algorithm. Fig. 8 thus illustrates that our setup achieved a good geographical spread of "Switched On" vehicles.

Next, we performed our simulations proper, where a pedestrian was inserted onto the map at the beginning of each simulation, and the emulations ran until either: (i) the pedestrian was detected by a parked vehicle that was "Switched On" and thus actively searching at the same time as when the pedestrian was passing by; or (ii) thirty minutes had lapsed and no detection event had occurred. The data collected from our experiment comprised of: (i) the average time taken (in minutes) until detection of the missing entity occurred (provided that the detection occurred within thirty minutes from the beginning of an emulation, else a fail result was recorded); (ii) the population standard deviation (in minutes) from the average time taken until detection; and (iii) the total number of times that fail results were recorded over

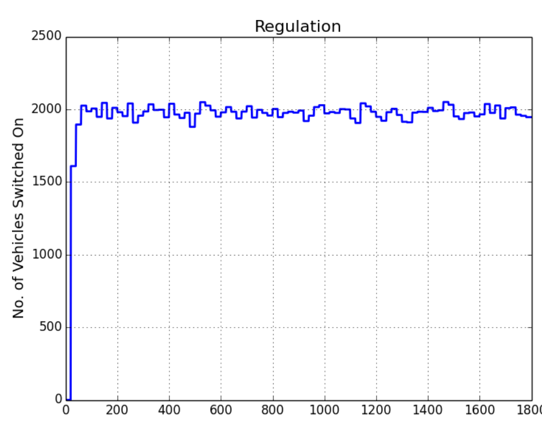

Fig. 5. Regulating to have 2,000 "Switched On" vehicles at any point in time.

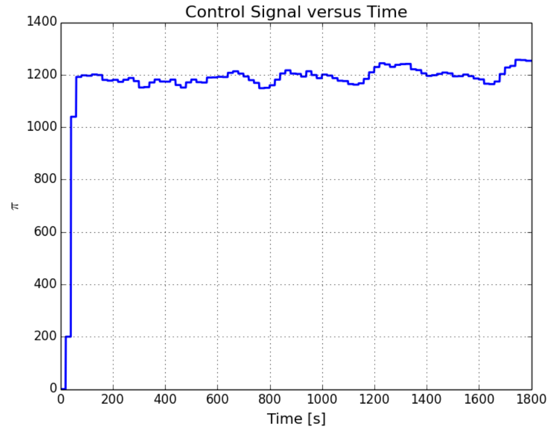

Fig. 6. The control signal, π, versus time.

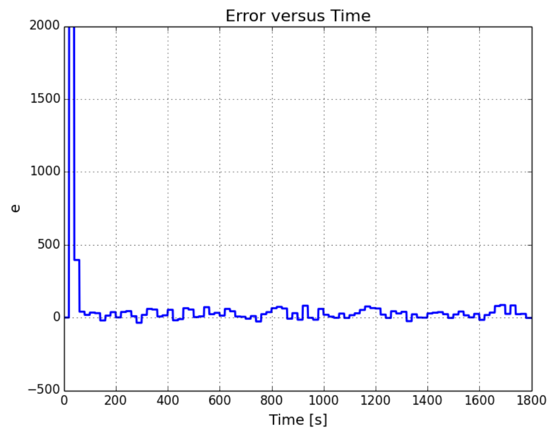

Fig. 7. The error signal, e, versus time.

the entirety of the experiment. To reiterate, 100 simulations in total were conducted over the course of our experiment. The results were as follows: (a) Average Detection Time $=$ 6.08 minutes; (b) Pop. Standard Deviation $=$ 6.65 minutes; and (c) Failed to Detect $=$ 7 times out of 100 simulations. In other words, the pedestrian was not detected, within a thirty minute time frame, 7% of the time. For the other 93 cases, the pedestrian was detected, on average, in approximately six minutes.

VI. CONCLUSIONS AND FUTURE WORK

We envisage a number of ways forward in regard to improving our experimental setup, including performing more simulations, and testing under different and more realistic parameters. Theoretically, we would like to consider dynamic parking, which would mean examining time-varying probability models. We are also interested in utilising Markov models to introduce a notion of mode-dependent probability, i.e., such that "switching on" over the next time step, $k \cdot h$, is dependent on whether a vehicle is currently "Switched On" or not.

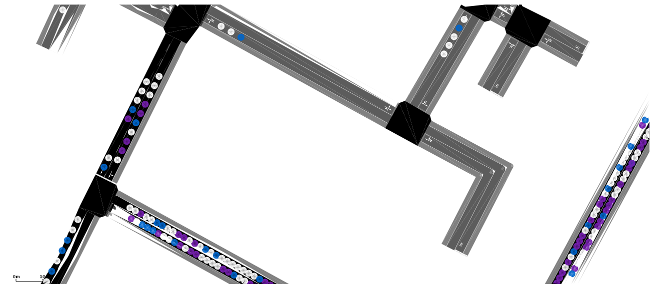

Fig. 8. Geographical distribution of parked vehicles from a subsection of the SUMO network.

REFERENCES

[1] Alzheimer's Association, *MedicAlert + Safe Return*. Accessed: Sep. 4, 2018. [Online]. Available: https://www.alz.org/help-support/caregiving/safety/medicalert-safe-return

[2] N. Lu, N. Cheng, N. Zhang, X. Shen and J. W. Mark, "Connected vehicles: solutions and challenges," *IEEE Internet of Things Journal*, vol. 1, no. 4, pp. 289-299, 2014.

[3] R. Cogill, O. Gallay, W. Griggs, C. Lee, Z. Nabi, R. Ordonez, M. Rufli, R. Shorten, T. Tchrakian, R. Verago, F. Wirth and S. Zhuk, "Parked cars as a service delivery platform," in *Proc. of the 3rd International Conference on Connected Vehicles and Expo*, Vienna, Austria, 2014, pp. 138-143.

[4] R. Verago, J. Naoum-Sawaya, W. Griggs and R. Shorten, "Localization of missing entities using parked cars," in *Proc. of the 4th International Conference on Connected Vehicles and Expo*, Shenzhen, China, 2015, pp. 40-41.

[5] W. M. Griggs, R. Verago, J. Naoum-Sawaya, R. H. Ordóñez-Hurtado, R. Gilmore and R. N. Shorten, "Localising missing entities using parked vehicles: an RFID-based system," *IEEE Internet of Things Journal*, doi: 10.1109/JIOT.2018.2864590.

[6] https://openclipart.org/detail/234440 Accessed: Apr. 16, 2018.

[7] https://openclipart.org/detail/201733 Accessed: Apr. 16, 2018.

[8] C. H. Papadimitriou and J. N. Tsitsiklis, "The complexity of optimal queuing network control," *Mathematics of Operations Research*, vol. 24, no. 2, pp. 293-305, 1999.

[9] A. R. Fioravanti, J. Mareček, R. N. Shorten, M. Souza and F. R. Wirth, "On classical control and smart cities," in *Proc. of the IEEE 56th Annual Conference on Decision and Control*, Melbourne, Australia, 2017, pp. 1413-1420.

[10] A. R. Fioravanti, J. Mareček, R. Shorten, M. Souza and F. Wirth, "On the ergodic control of ensembles." Accessed: Sep. 4, 2018. [Online]. Available: https://arxiv.org/abs/1807.03256

[11] G. F. Franklin, J. D. Powell and A. Emami-Naeini, *Feedback Control of Dynamic Systems*, 3rd ed. Reading, MA: Addison-Wesley, 1994.

[12] G. F. Franklin, J. D. Powell and M. Workman, *Digital Control of Dynamic Systems*, 3rd ed. Englewood Cliffs, NJ: Prentice Hall, 1997.

[13] M. F. Barnsley, S. G. Demko, J. H. Elton and J. S. Geronimo, "Invariant measures for Markov processes arising from iterated function systems with place-dependent probabilities," *Annales de l'Institut Henri Poincaré (B) Probabilités et Statistiques*, vol. 24, no. 3, pp. 367-394, 1988.

[14] D. Krajzewicz, J. Erdmann, M. Behrisch and L. Bieker, "Recent development and applications of SUMO – Simulation of Urban MObility," *International Journal on Advances in Systems and Measurements*, vol. 5, no. 3 & 4, pp. 128-138, 2012.

[15] A. Wegener, M. Piórkowski, M. Raya, H. Hellbrück, S. Fischer and J-P. Hubaux, "TraCI: an interface for coupling road traffic and network simulators," in *Proc. of the 11th Communications and Networking Simulation Symposium*, Ottawa, Canada, 2008, pp. 155-163.

[16] Dublin Street Parking Services, *Costs & Zones*. Accessed: Sep. 4, 2018. [Online]. Available: https://dsps.ie/costs-and-zones

[17] Simulation of Urban MObility – Wiki, *Simulation/Pedestrians*. Accessed: Sep. 4, 2018. [Online]. Available: http://sumo.dlr.de/wiki/Simulation/Pedestrians#Model_nonInteraction

[18] Irish Parking Association, *Frequently Asked Questions*. Accessed: Sep. 4, 2018. [Online]. Available: http://www.parkingireland.ie/car-parking/frequently-asked-questions

[19] S. Armstrong, *6 Factors That Affect RFID Read Range*, RFID Insider, Jul. 2013. Accessed: Sep. 4, 2018. [Online]. Available: http://blog.atlasrfidstore.com/improve-rfid-read-range