Neuromodulated Patience for Robot and Self-Driving Vehicle Navigation

Jinwei Xing*, Xinyun Zou†, and Jeffrey L. Krichmar*†
*Department of Cognitive Sciences
†Department of Computer Science
University of California, Irvine
Irvine, California, USA, 92697
Emails: {jinweix1, xinyunz5, jkrichma}@uci.edu

***Abstract*—Robots and self-driving vehicles face a number of challenges when navigating through real environments. Successful navigation in dynamic environments requires prioritizing subtasks and monitoring resources. Animals are under similar constraints. It has been shown that the neuromodulator serotonin (5-HT) regulates impulsiveness and patience in animals. In the present paper, we take inspiration from the serotonergic system and apply it to the task of robot navigation. In a set of outdoor experiments, we show how changing the level of patience can affect the amount of time the robot will spend searching for a desired location. To navigate GPS compromised environments, we introduce a deep reinforcement learning paradigm in which the robot learns to follow sidewalks. This may further regulate a tradeoff between a smooth long route and a rough shorter route. Using patience as a parameter may be beneficial for autonomous systems under time pressure.**

***Index Terms*—autonomous vehicle, deep reinforcement learning, impulsiveness, navigation, neuromodulation, road following, serotonin**

I. Introduction

Real-world environments can change due to the season, time of day, construction, or the behavior of other agents. Furthermore, goals, motivations, or context can change due to altered conditions. Uncertainty can arise due to sensor noise, unforeseen obstacles or uncertain goals. An autonomous system needs to cope with these challenges and have the ability to rapidly adapt its behavior based on the current situation.

For successful behavior in a dynamic world, an agent may need to tradeoff between patience and assertiveness. For example, a self-driving car may get stuck at a four-way stop sign because human drivers are not waiting their turn. A self-driving car that became impatient would eventually assert itself, and move into the intersection. On the other hand, in a dangerous driving situation (e.g., icy roads), an autonomous vehicle may need to slow down and possibly delay its arrival time for safe travel. In this case, patience is a virtue. Or if a search and rescue robot's task is to locate as many injured people as possible, even if it means the robot could run out of energy, patient search would be a priority. In these cases, a signal dynamically regulating the patience, or impatience, of the autonomous system would be beneficial.

Biological inspiration for regulating patience in autonomous systems could be obtained from the mammalian nervous system, which has a number of neuromodulators that regulate context, signal changes, and direct actions. The neuromodulator serotonin (5-HT) is thought to have a role in harm aversion, anxious states, and temporal discounting [1]. Recently, Miyazaki and colleagues showed that optogenetically increasing 5-HT levels caused mice to be more patient, especially when the timing of a reward was uncertain [2]. Based on these results, they developed a Bayesian decision model for the probability to wait or quit.

Although great progress has been made in the robotics community for path planning, there are still a number of open issues when it comes to flexible navigation under dynamic conditions [3]. Classic path planning algorithms include Dijkstra's algorithm, A Star (A*), and D*. Dijkstra's algorithm uses a cost function from the starting point to the desired goal. A* additionally considers the distance from the start to the goal "as the crow flies" [4]. D* extends the A* algorithm by working backward from the goal toward the start position, and can readjust costs, allowing it to replan paths in the face of obstacles [5]. However, these cost functions are typically fixed or deterministic. Neurobiolgically inspired algorithms have demonstrated the ability to readjust paths depending on cost, such as our work on adaptive path planning [6], and Erdem and Hasselmo's work that demonstrated the ability to take shortcuts [7]. The above algorithms do not consider motivation or context, and do not reflect the flexibility observed in animal navigation.

In order to add context and flexibility to path planning, we apply the rodent model of patience [2] to a ground robot. Specifically, our robot navigates through a series of waypoints. The level of 5-HT dictates how patiently the robot will search for a waypoint. We show that changing the 5-HT level can have dramatic effects on the robot's behavior. Such a system may be beneficial for adjusting autonomous behavior depending on the context and uncertainty of a situation.

II. Methods

A. Navigation task

Robot navigation tasks were carried out in two different outdoor parks with varying terrain and features. Figure 1 shows satellite images of the two parks. Waypoints were GPS coordinates placed on sidewalks in the park. The park on the left of Figure 1, Encinitas Community park, was

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

relatively flat. Waypoints were placed along the perimeter of the test area on either sidewalks or the paved parking lot. In the middle of the test area was a grassy region with some trees. The park on the right of Figure 1, Aldrich park at the University of California, Irvine, was hilly with numerous obstacles (e.g., bushes, benches, and buildings). It should be noted that the Aldrich park test area was in a sunken bowl surrounded by tall buildings and trees. These features made GPS signals unreliable. For this reason, a road following algorithm, which will be discussed below, was introduced to assist with navigation. The waypoints were placed on the sidewalk that surrounded the inner grassy region.

In both parks, the robot's task was to proceed to each waypoint in order. If the robot became impatient, it would skip searching for the present waypoint and randomly choose a future waypoint. However, the robot had to reach the last waypoint for a trial to be complete.

B. Robot and Software Design

For the robot experiments, we used the Android-Based Robotic platform [6], a mobile ground robot constructed from off-the-shelf commodity parts and controlled through an Android smartphone (see Figure 2). An IOIO-OTG microcontroller communicated with an Android smartphone via a Bluetooth connection and relayed motor commands to a separate motor controller for steering the Dagu Wild Thumper 6-Wheel Drive All-Terrain chassis. Three ultrasonic sensors, which were used for obstacle avoidance, were connected to the robot through the IOIO-OTG. A software application, which controlled the robot, was written in Java using Android Studio and deployed on a Google Pixel XL smartphone. The application utilized the phone's built-in camera, accelerometer, gyroscope, compass, and GPS for navigation.

For waypoint navigation, a GPS location was queried using the Google Play services location API. The bearing direction from the current GPS location of the robot to a desired waypoint was calculated using the Android API function bearingTo. A second value, the heading, was calculated by subtracting declination of the robot's location to the smartphone compass value, which was relative to magnetic north. This resulted in an azimuth direction relative to true North. The robot traveled forward and steered in attempt to minimize the difference between the bearing and heading. The steering direction was determined by deciding whether turning left or turning right would require the least amount of steering to match the bearing and heading. The navigation procedure continued until the distance between the robot's location and the current waypoint was less than 20 meters, at which point the next waypoint in the list was selected.

C. Waypoint Navigation and Model of Neuromodulated Patience

The robot proceeded through a list of waypoints as described above. However, if the robot became impatient, it skipped the present waypoint and randomly chose a waypoint closer to the final destination.

The likelihood to skip a waypoint was based on the Bayesian Decision Model given by [2]. Specifically, we calculated the probability to wait given the time elapsed:

$$p(wait|t) = \frac{1}{1 + exp^{\beta \cdot 5HT \cdot L(t)}}, \tag{1}$$

where β was equal to 50, and L(t) was the likelihood of reaching the waypoint at time t, and 5HT denoted the serotonin level. The likelihood was calculated with a Normal cumulative distribution function having a mean of 40 seconds and a standard deviation of 20 seconds. The likelihood function was multiplied by a scalar that represented the probability of receiving a reward. As in [2], we assumed that increasing 5-HT levels caused an overestimation of the prior probability. Therefore, in our experiments low 5-HT equated to a probability of a reward of 0.50 and high 5-HT equated to probability of a reward of 0.95 (see [2] for details). Figure 3 shows the resulting probability to wait, $p(Wait|t)$, curves.

The $p(Wait|t)$ curves in Figure 3 were used to decide whether to keep searching for a waypoint or to forego the desired waypoint and choose another. A random number between 0 and 1 was generated and if the number was greater than $p(Wait|t)$, where t was the time elapsed that the robot had been searching for a waypoint, the robot stopped searching for this waypoint. A new waypoint was randomly chosen that was closer to the final destination. Note that if the robot was searching for the final destination waypoint or for a waypoint after a skip, the $p(Wait|t)$ curve was not referenced. That is, the robot had to reach the shortcut waypoint or had to reach the final waypoint for a successful trial. See Algorithm 1 for implementation details.

D. Road Following with Deep Reinforcement Learning

A road following algorithm based on deep reinforcement learning was used in the experiments carried out in Aldrich park. This became necessary due to poor GPS reception in this environment. We used a Deep Q-Network (DQN) for online learning of a driving policy on the Aldrich park sidewalks [8].

1) Road Following DQN States and Actions: In reinforcement learning, an agent is acting in an environment. At each time step t, the agent chooses an action $a_t \in A$ in response to the current state $s_t \in S$. The system makes the transition from s_t to s_{t+1} with a reward r_t based on the reward function $R(s_t, a_t)$. The goal of reinforcement learning algorithms is to learn a policy that maps a state s to an action a , such that the expected sum of rewards $\mathbb{E}_\pi[\sum_t^\infty \gamma^t r^t | s_t, a_t]$ is maximized where π is the agent's behavior function. $\gamma \in [0, 1]$ is a discounting factor used to penalize the rewards in the future. As a value-based deep reinforcement learning method, the DQN learns a state-action value function $Q_\theta(s, a)$ which outputs the expected discounted sum of future rewards that will be received by following the policy. Some recent works used deep reinforcement learning in robot navigation tasks [9], [10], but all of them are set in ideal indoor environments. To the best of our knowledge, our project is the first work that trained the

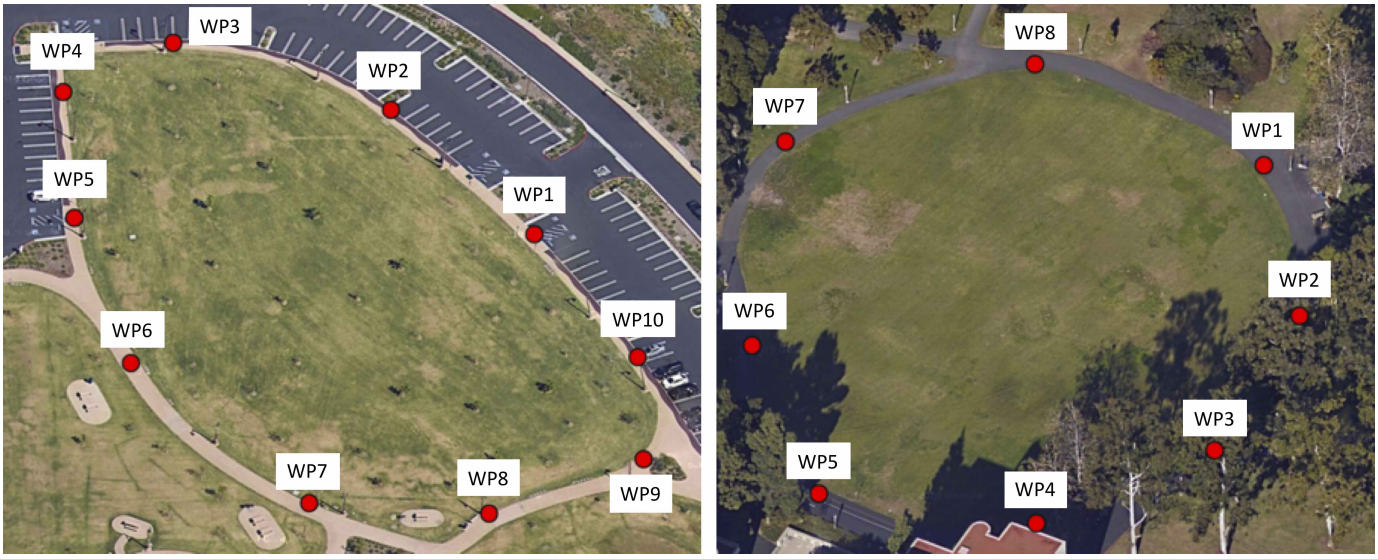

Fig. 1: Parks where robot navigation experiments were carried out. The left is an image of Encinitas Community park and the right is an image of Aldrich park at the University of California, Irvine. The labels denote the waypoints (i.e., WP1...WP10). Waypoints were approximately 50-60 meters apart. Imagery from Google Maps, 2019.

(a) Android Based Robot in Encinitas Park.

(b) Android Based Robot in Aldrich Park.

Fig. 2: Android Based Robot used for the experiments.

robots to navigate in complicated outdoor environments with deep reinforcement learning.

In our experiments that utilized road following, the agent was the Android-Based Robot and the environment was Aldrich Park (see figure 6). The state was represented by an annotated camera image, as will be described in Section II-D2, from the smartphone that was mounted on the robot. The reward was either 0.5 when the robot stayed on road or 0 when the robot went off road. In the beginning of each training episode, the robot was initialized in the center of the road. When the robot was not on road, the episode ended and the robot was reset to the center of the road for the next episode. In each step, the robot moved forward for 0.6 seconds with a constant speed but a different steering angle ranged from sharp left to slight left to straight to slight right and to sharp right. During the training, the robot was reinforced by staying on the road. After around 15 episodes and 2 hours of training, roughly 2000 training steps, the robot learned to follow the road.

2) Semantic Segmentation of Images: To evaluate the states of the robot in the environment and then generate rewards for the deep reinforcement learning module, we used ENet [11], a pixel-wise real-time semantic segmentation neural network. ENet labeled each pixel of the image as road or non-road. We used middle-bottom portion of the segmented image to evaluate if the robot was on road. The image size in the experiment is 320x240 pixels and the size of middle-bottom portion for evaluation is 80x32 pixels. If most pixels in that portion were labeled as road, we judged that the robot was on road. Otherwise, the robot was thought to be off-road.

The environment of Aldrich Park and camera setting in this project were very different from those of popular datasets

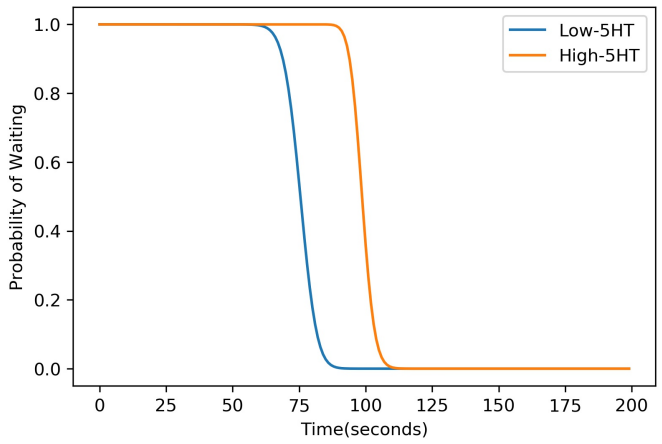

Fig. 3: Probability of waiting function. Higher 5-HT levels shifted the curve to the right resulting in longer wait times.

such as Kitti [12] where road detection was also involved. Therefore, we created a scene understanding dataset for the robot from data collected in Aldrich Park. Smartphone camera frames were collected in Aldrich Park at different times (i.e., 2pm to 7pm) of day. We selected 418 distinct and representative pictures and did binary (road and non-road) pixel-wise labeling for these using the PixelAnnotationTool from [13]. Figure 4 shows examples of semantic segmentation taken from the Aldrich Park dataset. The ENet model trained on the Aldrich Park dataset allowed us to rapidly label road and non-road portions of a scene and generate rewards for the deep reinforcement learning module.

Besides its necessity for reward generation, the semantic segmentation module provided two other benefits. First, the segmented observation, which was fed to the deep reinforcement module, removed noisy information from the original image and kept the most important features (road or non-road). This simplified the task for deep reinforcement learning and thus the training of the DQN was faster. Second, the semantic segmentation module increased the generalization and adaptability of the self-driving navigation to handle dynamic characteristics of outdoor environments such as lighting changes due to time of day or weather. Examples in Figure 4 show some of the various lighting conditions in the park. Without the semantic segmentation module, the DQN trained at 2pm could not work at 7pm because sunlight changed. To solve this problem without the semantic segmentation module, we would have needed to train under all different environment situations, which would be time consuming and would need to deal with potential problems such as catastrophic forgetting. Another case that demonstrates the advantage of semantic segmentation is that, the robot could avoid a pedestrian automatically because the pedestrian would be labeled as non-road and the robot would try to stay on road. The robot trained without semantic segmentation could not achieve this and would instead take random action since the appearance of a pedestrian was a novel state for it. By

Algorithm 1: Waypoint Navigation with Neuromodulated Patience

Input: GPS and compass readings, N waypoints;
Initialize waypoint index w = 0;
Initialize time count t = 0;
Initialize $shortcut$ = false;
Initialize $finished$ = false;
while *not* $finished$ **do**
  get the current GPS and compass readings;
  if *robot is within 20m of waypoint*(w) **then**
    if $w == N$ **then**
      $finished$ = true;
      break;
    else
      $w = w + 1$;
      $t = 0$;
    end
    $shortcut$ = false;
  end
  if *not* $shortcut$ *and* w *!=* N **then**
    generate a random number $rand\_num$ and update $p(Wait|t)$;
    if $rand\_num > p(Wait|t)$ **then**
      update w with a random integer in the range of $[w + 1, N)$;
      $t = 0$;
      $shortcut$ = true;
    end
  end
  if *not* $shortcut$ *and in Aldrich park and on road* **then**
    move forward toward waypoint(w) based on road following algorithm;
  else
    use GPS and compass to get bearing to waypoint(w);
    navigate toward waypoint(w);
  end
  $t = t + 1$;
end

separating the scene understanding task from reinforcement learning, semantic segmentation enables faster training and better generalization capability [14].

3) Road Following Data Pipeline: Figure 5 shows the data pipeline. The Android Based Robot took pictures with the smartphone's camera. Using a WiFi "hotspot", the image was sent to a nearby laptop, which performed real-time image segmentation of "road" versus "non-road". The laptop also ran a deep reinforcement learning network, based on the DQN, which processed the image and outputted action values used by the agent to choose actions. The actions ranged from sharp left to slight left to straight to slight right and to sharp right. The reward was also based on the segmented state. A detailed illustration of the road following deep reinforcement neural

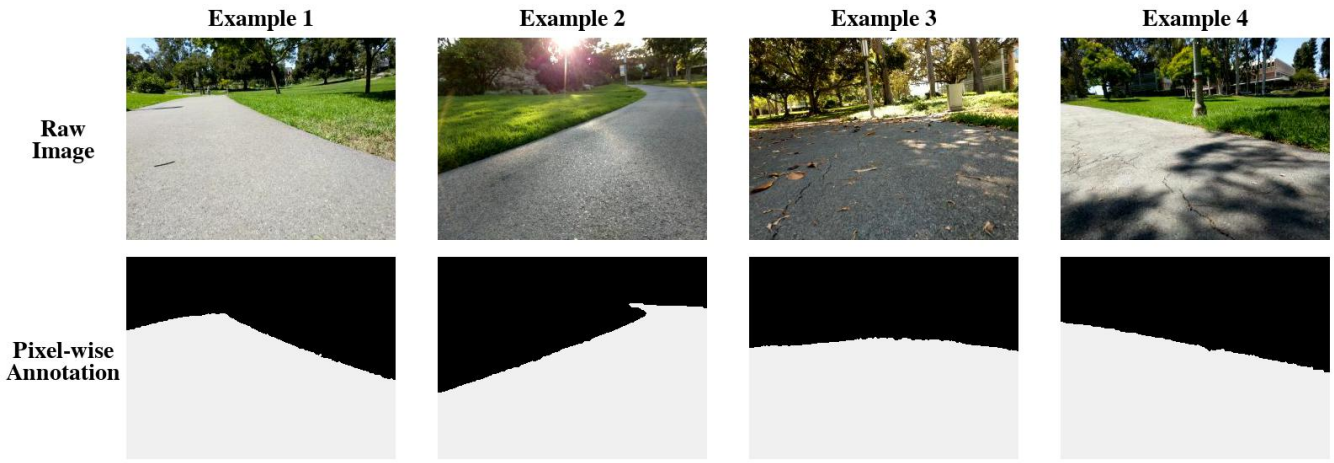

Fig. 4: Examples of raw images and pixel-wise annotations in Aldrich Park dataset. The size of each image is 320x240 pixels and each pixel has a label of either 1 (road) or 0 (non-road). In visualized annotations (lower half), the non-road portion is shown in black and the road portion is in gray.

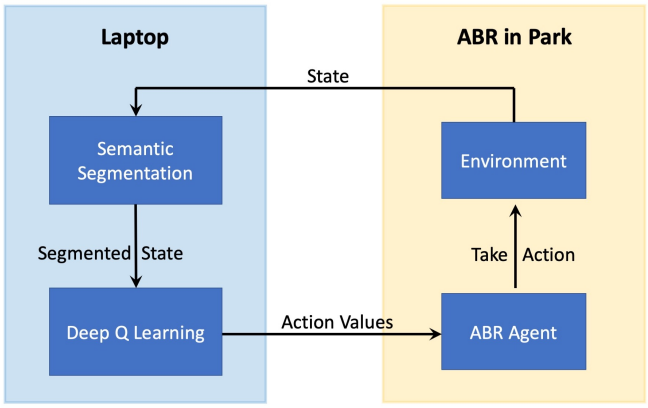

Fig. 5: High level illustration of the data pipeline for the road following algorithm. Images from the Android Based Robot's (ABR) smartphone camera are sent to a nearby laptop via a socket. The laptop runs a deep reinforcement learning algorithm, which rewards staying on the road, and generates steering actions for the robot.

network is given in Figure 6. The laptop took about 400 ms to process the information, generate an action, and update the network. This was adequate for online learning in real-time.

The robot learned to follow the road after roughly 2000 training steps. The road following algorithm was used in Aldrich park where waypoints were set along the sidewalk that surrounded a hilly grass region (Figure 1, right). The robot could move to the next waypoint by following the road. In the present experiments, we segmented road and non-road. But, potentially, we could also segment people, trees, benches, etc. These object classes could be used as further inputs for training the network and implementing more complex behavior.

III. Results

Two sets of robot navigation experiments were carried out. One set was in the Encinitas Community park (see Figure 1 left) and the other was in Aldrich park (see Figure 1 right). In both cases, the robot navigated through a set of waypoints with low and high 5-HT levels. In Aldrich park, the navigation experiments were carried out with road following activated.

A. Waypoint Navigation in Encinitas Community park

We ran 6 trials for low 5-HT and 6 trials for high 5-HT in the Encinitas Community park (see Figure 7). The waypoints were roughly 50-60 meters apart. In Figure 7, each marker denotes the GPS location from the smartphone when the robot was within 20 meters of a waypoint (different colors denote different trials). Note that this reading could vary dramatically due to GPS inaccuracies.

The level of 5-HT affected the robot's patience in finding a waypoint. Over the 6 trials, 9 waypoints were skipped when 5-HT was low, but only 2 waypoints were skipped when 5-HT was high. The average time before skipping a waypoint was 68 seconds for low 5-HT and 97 seconds for high 5-HT (see Table I). These experiments demonstrated how this model could change route planning behaviors.

Figure 8 shows all the GPS readings from two representative trials, one with high 5-HT and the other with low 5-HT. In the high 5-HT trial, the robot reached every waypoint. In the low 5-HT trial, the probability to wait was exceeded for reaching Waypoint 6 after 69 seconds and the robot skipped to Waypoint 9. A video of the robot performing waypoint navigation with low 5-HT can be found at: https://youtu.be/6EcNchTGLKw, and a video of the robot performing waypoint navigation with high 5-HT can be found at: https://youtu.be/q_m0gbVN6UE.

B. Waypoint Navigation in Aldrich park

We ran 5 high 5-HT trials and 5 low 5-HT trials in Aldrich park (see Figure 9). Since the area is sunken in a bowl

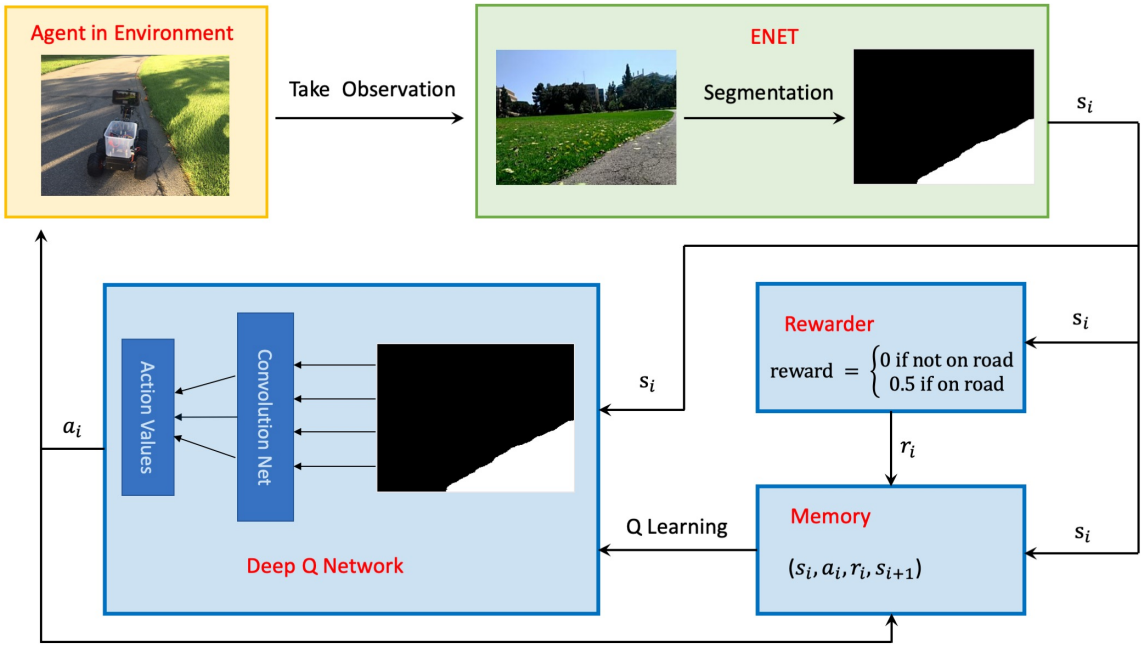

Fig. 6: Detailed illustration of the data pipeline for the road following algorithm. ENet was used to segment road from non-road [11]. The network gave a positive reward for actions that kept the robot on the road and a penalty for actions that caused the robot to go off road. Training was based on a DQN reinforcement learning paradigm [8]. Training and testing were carried out online in Aldrich park.

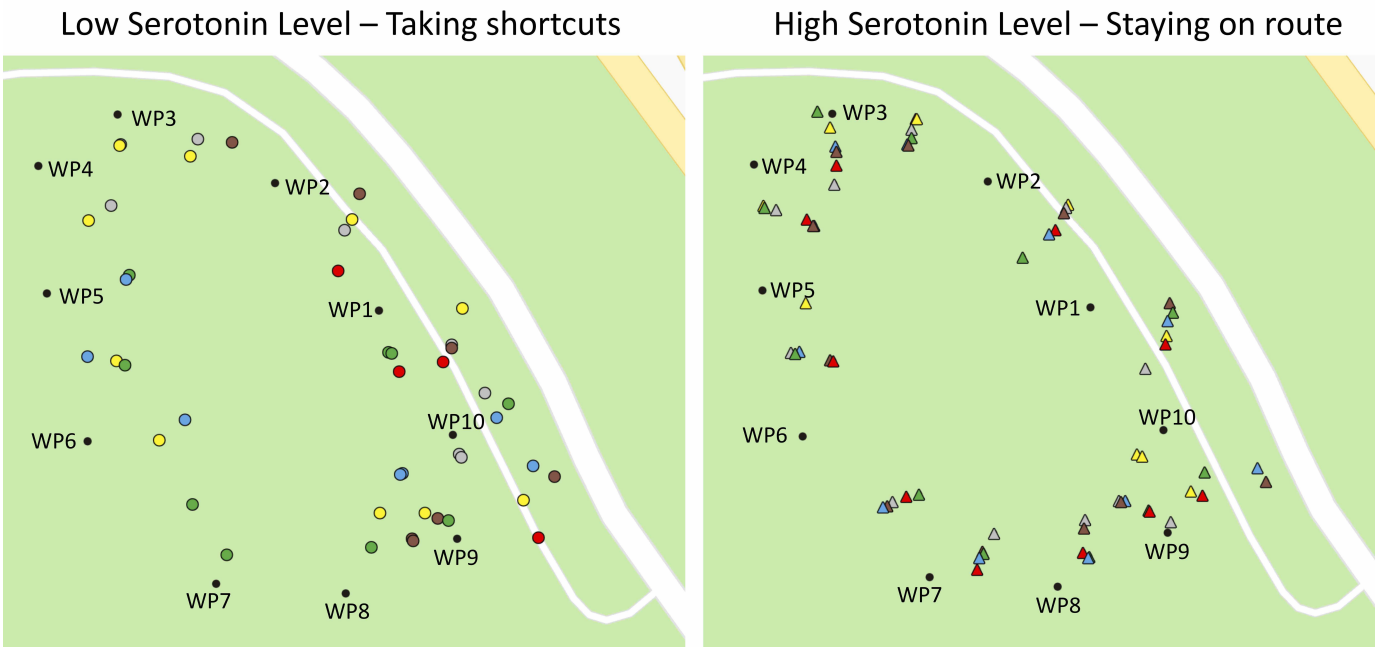

Fig. 7: Robot navigation trials in the Encinitas Community park. The black dots are the waypoint destinations (WP1 – WP10). There were 6 trials. Each color represents an individual trial. Each colored marker denotes the robot reaching a waypoint.

TABLE I: Results of High/Low 5-HT Modulating Navigation

	Encinitas Park		Aldrich Park	
	High 5-HT	Low 5-HT	High 5-HT	Low 5-HT
Navigation Time (s)	525.521	413.549	414.905	389.625
Shortcuts	0.3	1.5	0.0	1.4
Waypoints Reached	9.67	6.5	8.0	6.0

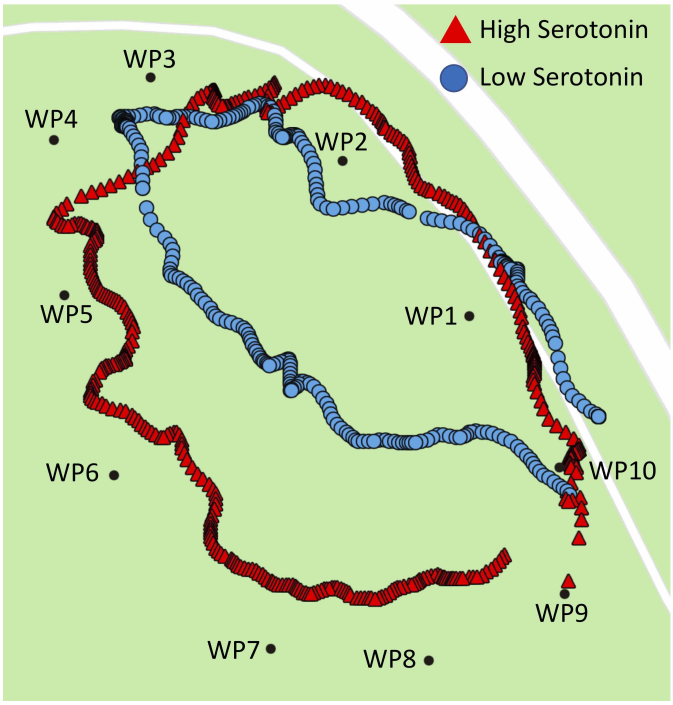

Fig. 8: Two representative navigation trials in the Encinitas Community park. All the GPS points are shown. The markers in the figure correspond to the grey markers in Figure 7.

surrounded by tall buildings and trees, the GPS readings were highly inaccurate. In particular, the robot had difficulty finding Waypoints 2 and 3 due to the poor GPS signal. As a result, we introduced the road following algorithm described in Section II-D, which helped the robot stay on the sidewalk and increased the likelihood of finding a waypoint within the probability of wait constraint. During road following based navigation, the robot moved towards the waypoint by following the road. Since the waypoints were placed along the outer ring of the test area, the robot tended to move closer to the next waypoint by following the road. When the robot decided to take a shortcut because of being impatient, the movement of the robot was based on GPS because the shortcuts took the robot off the road and over the grassy interior of the test area. Once the shortcut waypoint was reached, the road following algorithm took over again.

It should be noted that during road following, the robot took longer to complete the course with high 5-HT (i.e., 420 seconds on average) than with low 5-HT (i.e., 390 seconds on average) in which shortcuts were taken. However, since the robot was traveling over smoother terrain with high 5-HT, it reached more waypoints and took less energy than when it took shortcuts with low 5-HT (see Table I). A video of the robot navigating using road following can be found at: https://youtu.be/DixOxO2UafQ.

These results show the benefits and the tradeoffs associated with being patient versus being impulsive during navigation. In both two parks, when 5-HT was high, the robot was more patient when navigating towards waypoints, which meant it took less shortcuts and reached more waypoints, but at the cost of taking longer to complete a trial (see Table I).

IV. Discussion

In the present paper, we showed how a concept from behavioral neuroscience could be applied to robot navigation and possibly self-driving vehicles. It has been shown that 5-HT in the brain affects impulsiveness in an animal's behavior [2]. The present model applied this idea to waypoint navigation in autonomous robots. Specifically, we showed that simulating high 5-HT led to increased search time for a desired location and that simulating low 5-HT led to an increase in calling off the search for some waypoints. Even under high 5-HT conditions, if a waypoint was particularly difficult to find or there were environmental challenges, there was a limit to how long the robot would try to reach a desired location (see Figure 9). Our results showed that neuromodulated patience led to flexible behaviors, which are not typically found in traditional navigation solutions [3], [5].

The goal of the present algorithm and demonstrations was not to achieve some benchmark, but rather to suggest a neurobiologically inspired strategy that could complement other navigation systems. The present approach could be applied to biomimetic navigation systems [15], [16], as well as engineering approaches to navigation [17], [18]. In general, the probability to wait suggests a level of urgency in the overall system. We imagine this could be applied to a number of tasks where resource allocation is time critical.

Furthermore, the probability of waiting could be associated to some internal parameter in the system (e.g., battery level or prioritizing goals). Presumably, the impulsiveness signal in the rodent is closely tied to its natural foraging behavior. The animal will search for food, but the time it will search depends on the food value and on the uncertainty of the food resource. Such considerations could be beneficial for a robot navigation system or for a self-driving vehicle.

The patience-based neuromodulated navigation algorithm adds another dimension to the present navigation system. By giving the robot an alternative to point-to-point navigation, the robot now must weigh the cost of staying on a smooth and

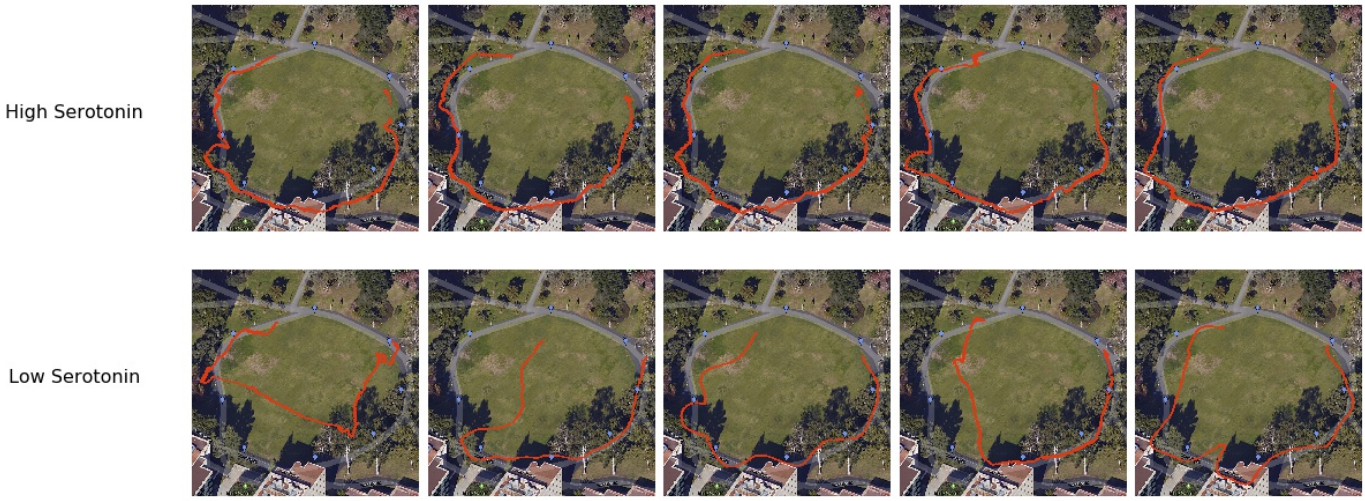

Fig. 9: Five trials in Aldrich park with high 5-HT (upper figure) and five trials in Aldrich park with low 5-HT (lower figure). Red traces are drawn from GPS readings from the phone mounted on the robot. The robot went across all waypoints one by one in trials with high 5-HT and took multiple shortcuts when 5-HT was low.

reliable road that may take longer to travel versus traversing over rough terrain that may be shorter but takes more energy and could be potentially harmful to the robot. Since the deep reinforcement learning introduced here is designed for online learning, these costs could be learned along with the rewards for staying on the road. Ideally, the deep reinforcement learning algorithm could set the 5-HT level dynamically.

The present algorithm is a step towards a complete navigation or self-driving system that takes inspiration from neurobiology and behavioral neuroscience.

V. Acknowledgements

This work was supported by the Defense Advanced Research Projects Agency (DARPA) via Air Force Research Laboratory (AFRL) Contract No. FA8750-18-C-0103 (Lifelong Learning Machines: L2M), and by the Air Force Office of Scientific Research (AFOSR) Contract No. FA9550-19-1-0306. The authors would like to thank Nicholas Ketz, Soheil Kolouri and Andrea Soltoggio for fruitful discussions and suggestions. The authors would also like to thank Katsuhiko Miyazaki for making his code available.

References

[1] M. C. Avery and J. L. Krichmar, "Neuromodulatory systems and their interactions: A review of models, theories, and experiments," *Frontiers in Neural Circuits*, vol. 11, no. 108, 2017.

[2] K. Miyazaki, K. W. Miyazaki, A. Yamanaka, T. Tokuda, K. F. Tanaka, and K. Doya, "Reward probability and timing uncertainty alter the effect of dorsal raphe serotonin neurons on patience," *Nature Communications*, vol. 9, no. 1, p. 2048, 2018.

[3] S. Lavalle, "Motion planning: Part ii: Wild frontiers," *IEEE Robot. Autom. Mag.*, vol. 18, pp. 108–118, 2011.

[4] P. E. Hart, N. J. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE Trans. Syst. Sci. Cybern*, vol. 4, pp. 100–107, 1968.

[5] A. Stentz, "Optimal and efficient path planning for partially-known environments," in *Proceedings of the 1994 IEEE International Conference on Robotics and Automation*, May 1994, pp. 3310–3317 vol.4.

[6] T. Hwu, A. Y. Wang, N. Oros, and J. L. Krichmar, "Adaptive robot path planning using a spiking neuron algorithm with axonal delays," *IEEE Transactions on Cognitive and Developmental Systems*, vol. 10, no. 2, pp. 126–137, 2018.

[7] U. M. Erdem and M. Hasselmo, "A goal-directed spatial navigation model using forward trajectory planning based on grid cells," *Eur. J. Neurosci.*, vol. 35, pp. 916–931, 2012.

[8] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, "Playing atari with deep reinforcement learning," *arXiv preprint arXiv:1312.5602*, 2013.

[9] G. Kahn, A. Villaflor, B. Ding, P. Abbeel, and S. Levine, "Self-supervised deep reinforcement learning with generalized computation graphs for robot navigation," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 1–8.

[10] A. Faust, K. Oslund, O. Ramirez, A. Francis, L. Tapia, M. Fiser, and J. Davidson, "Prm-rl: Long-range robotic navigation tasks by combining reinforcement learning and sampling-based planning," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 5113–5120.

[11] A. Paszke, A. Chaurasia, S. Kim, and E. Culurciello, "Enet: A deep neural network architecture for real-time semantic segmentation," *arXiv:1606.02147 [cs.CV]*, 2016.

[12] M. Menze and A. Geiger, "Object scene flow for autonomous vehicles," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[13] A. Bréhéret, "Pixel annotation tool," https://github.com/abreheret/PixelAnnotationTool, 2017.

[14] Z.-W. Hong, C. Yu-Ming, S.-Y. Su, T.-Y. Shann, Y.-H. Chang, H.-K. Yang, B. H.-L. Ho, C.-C. Tu, Y.-C. Chang, T.-C. Hsiao *et al.*, "Virtual-to-real: Learning to control in visual semantic segmentation," *arXiv preprint arXiv:1802.00285*, 2018.

[15] M. Milford and R. Schulz, "Principles of goal-directed spatial robot navigation in biomimetic models," *Philos Trans R Soc Lond B Biol Sci*, vol. 369, no. 1655, 2014.

[16] P. Gaussier, J. P. Banquet, N. Cuperlier, M. Quoy, L. Aubin, P. Y. Jacob, F. Sargolini, E. Save, J. L. Krichmar, and B. Poucet, "Merging information in the entorhinal cortex: what can we learn from robotics experiments and modeling?" *J Exp Biol*, vol. 222, no. Pt Suppl 1, 2019.

[17] C. Urmson, R. Simmons, and I. Nesnas, "A generic framework for robotic navigation," *2003 Ieee Aerospace Conference Proceedings, Vols 1-8*, pp. 2463–2470, 2003. [Online]. Available: ⟨GotoISI⟩://WOS:000185997700244

[18] Y. Wang, D. Mulvaney, I. Sillitoe, and E. Swere, "Robot navigation by waypoints," *Journal of Intelligent and Robotic Systems*, vol. 52, no. 2, pp. 175–207, 2008.