The Information Complexity of Navigating with Momentum

Bente Riegler, Daniel Polani and Volker Steuber

School of Physics, Engineering and Computer Science, University of Hertfordshire
b.riegler@herts.ac.uk

Abstract

Many models of organism navigation concern themselves in essence just with the sequence of locations visited and how to manage it. However, larger and bulkier organisms have also to deal with managing momentum. We expect that this affects the cognitive management of movement. Here we propose a simple model for the information processing complexity of navigation when velocity and acceleration are considered, moving away from a kinematic perspective to a partially dynamic model, to separate the effects of location and momentum management.

The work is discussed in the context of recent neurobiological research suggesting that biological agents plan around acceleration and deceleration phases, showing high neural activity during their body's velocity changes.

Introduction

Commonly, navigation and movement tasks are modeled by defining movements through a sequence of positions, and eventually to a final position, be it via key poses or through forward and inverse kinematics. In more physically involved scenarios other approaches had great successes modelling the specific dynamics of the agent and their domain. Such models include the presence of momentum and inertia and the use of force to effect changes. The inverse pendulum balancing task is a classical example and benchmark in nonlinear control theory (Boubaker, 2013). It has been solved with multiple algorithms taking the position and angular velocity as well as the mass and force into account (Boubaker, 2013; Furuta et al., 1992) to highlight few. In autonomous vehicle control, speed, angular velocity and vehicle mass constitutes the dynamic of the system. Reinforcement learning with hierarchical temporal abstraction has achieved safe control in merging traffic lanes (Shalev-Shwartz et al., 2016). In helicopter control, differential dynamic programming (Abbeel et al., 2006) allows for difficult maneuvers such as tail-in funnel or flip. Similarly, in legged robots approaches such as Zero-Moment-Point walking (Kajita et al., 2003; Mitobe et al., 2000) that generate walking motion while balancing the dynamical center of the particular agent. In all these approaches, the use and limits of force are largely given by real world physical models and the dynamics are centered around the specific use case.

Evidence from neurobiological research suggests that organisms whose brains include a cerebellum do in fact model and simulate their world in a dynamic instead of a kinematic way when controlling movement. Recent research by Becker and Person focusing on the cerebellar control activity of a mouse in regards to precisely reaching a goal position showed the importance of controlling velocities (Becker and Person, 2019). While the task is still defined by kinematics, the control very much requires velocity management. In their experiment, they measured both the velocity and the neural activity in the cerebellum of the mouse. While reaching for the goal, the mouse shows moderately high neural activity at the start of the reaching motion, during acceleration, and high activity during the deceleration phase, towards the end of the movement.

Having to consider velocity creates a timing component and thus requires some temporal planning. When operating with velocities, one has to manage momentum which requires the ability to integrate the application of forces. In fact, specific neural circuits in the cerebellum have the ability to integrate (Maex and Steuber, 2013). This suggests that an agent with a cerebellum is endowed with the capacity to predict or simulate possible future positions through forward integration which permits it to plan when the agent needs to manage velocity.

While the various scenarios of momentum/velocity management are specific to particular agents or organisms, we wish to extract several general insights which emerge from the information processing cost that momentum/velocity management requires as compared to purely kinematic/location-based navigation models. Current models that allow for velocity management be it in autonomous vehicle control, legged robot control or the inverse pendulum — though achieving balanced and precise movements within their problem domain — are too concerned with the specific agent, its physics and the task at hand to allow general insight. These models, furthermore, are not concerned with the pressure towards parsimonious information

processing which would not be necessary for a merely optimally performing solution in the given problem domain, but becomes highly relevant once it comes to biological agents (Polani, 2009). Therefore, we will introduce a minimalistic model to study aforementioned phenomena.

The paper is organized as follows: in section 2 we will define our models and in section 3 we will define how we measure cognitive cost. In section 4 we will present our experiments and results which we will discuss in section 5.

Perception-Action Models

The perception-action loop setup throughout this paper is in line with the general Reinforcement Learning framework modeled as a Markov Decision Process (MDP) (Sutton and Barto, 2018). States are given by $s \in S$ and actions are given by $a \in A$. We assume the individual transitions, given by $p(s_{t_2}|s_{t_1}, a)$, with s_{t_1} being the state at time t_1, a the action taken in s_{t_1} and s_{t_2} being the resulting state after the action is performed at t_2, to be deterministic. A policy is denoted as $\pi(a|s)$, which denotes the probability that action a is selected in state s; such transitions incur rewards r that the agent aims to maximize over the run. The achieved rewards are summarized by the Q-function $Q^\pi(s, a)$ which expresses the reward that is accrued when, starting in state s and selecting first action a, the agent proceeds to follow policy π. In traditional MDPs, one seeks to find $Q^*(s, a)$ which maximises this value over all possible policies. Given a state s, an optimal action in the state can be directly read off this quantity, by selecting the action a that achieves the highest $Q^*(s, a)$ for the state s. This forms the basis of the reward structures we consider in the following.

In the following we will introduce two models: The standard kinematic/location-based (K/L) model which we will use as a baseline and our new proposed acceleration/velocity (A/V) model which includes velocity and acceleration. This allows us to model and handle inertia of the agent though we will not be specifically modeling mass or other physical implications. We will limit ourselves to look at abstract simple one-dimensional grid-like models to make the salient differences between the two setups as apparent as possible.

Kinematic/Location-based Model without Velocity

A typical way to represent navigation or movement tasks in reinforcement learning models uses actions that comprise the agent taking a single step from one location to a neighbouring one. Technically, this means accelerating the agent, moving it one step and immediately stopping it. Interpreted physically, this can be considered a model for high-friction where a movement stops immediately when the applied action a ceases.

As defined by the MDP framework there exists a state space (S) and an action set (A). The state space (S) consists of a set of discrete positions (P) aligned in one dimension. The action set (A) consists of actions that move the agent to an adjacent state, here *move left* $(m = -1)$ and *move right* $(m = +1)$. Every action incurs a cost of 1. We apply no discount over time, but consider episodic tasks only. Specifically, we assume there exists a single goal state which can be any state of S or a set of such states. Any goal state is modeled as a trapping state, i.e. once reached, it does not allow further action and does not incur further costs. In RL terminology, an episode ends once the agent reaches a goal state. Note that the trapping property is important for an appropriate calculation of the informational costs. The grid world is finite and limited by walls. An action that pushes the agent into a wall leaves the agent in the same state but still incurs the usual cost of 1. Since in the present paper we only consider optimal policies, no agent will walk into walls.

Acceleration/Velocity Model

In the following we describe the extended model. In this model, the states are not only defined by their position on the grid world but also the agent's velocity. Thus, each state of the MDP is now a tuple of position and velocity. The states are now tuples of positions (P) and velocities (V). For simplicity, we only consider integer velocities[1]. The state space is now $S = P \times V$. In our model, during each time step, the dynamics of the world moves the agent to a new position based on its current position and velocity:

$$p_{t+1} = p_t + v_t \tag{1}$$

In our simplistic one-dimensional model, the agent can chose between three actions: *positive acceleration* to the right $(a = +1)$ and *negative acceleration* to the left $(a = -1)$, and *no change* $(a = 0)$ which are added to the velocity:

$$v_{t+1} = v_t + a_t \tag{2}$$

Note the agent can not directly affect its position. It can only influence its velocity which then affects the position change mediated through the velocity. The agent's change of velocity will happen at the same time as the change in position which means any position change due to the choice of an action will only be observable at the subsequent time step.

The cost function incurs a cost of 1 for each time step outside of the goal regardless of the action taken. This grid world is limited by walls keeping the agent inside the world. Just like in the K/L model the agent will on its own try to avoid hitting the wall when it would not be optimal otherwise (note, though, that there are starting position-velocity pairs in which the agent cannot avoid hitting a wall).

Reaching the goal We will use two types of goal sets. Both are specified by a goal position.

[1] A velocity specification includes directionality

The first type allows for any velocity: $\{p_g\} \times V$ (i.e. a single given position, but with arbitrary velocity). When reaching/passing p_g, the agent will automatically be stopped, effectively reducing its velocity to 0 instantaneously, even if the agent would overshoot the goal otherwise. In this scenario the responsibility to decelerate the agent is placed on the environment (we interpret this as offloading the informational cost of an instantaneous stop to the embodiment of the agent). As a real-life analogue, one can compare this to the arresting gear for airplanes landing on an aircraft carrier.

The second type of goal set, on the other hand, requires the velocity to reach precisely zero for the goal to be considered satisfactorily achieved: $(p_g, 0)$. This goal type requires the agent to explicitly decelerate/break before reaching the goal position. In particular, overshooting will be considered a miss.

Cognitive Cost

Cognitive processing in natural agents requires neural activity which is energetically expensive yet crucial for the survival of the agent in question (Laughlin, 2001; Polani, 2009). As such, keeping the processing cost as minimal as possible without losing optimality with regards to some value function becomes an important secondary objective. In vivo, measurements of the cognitive load or neural activity of a living being can be measured using EEG (Niedermeyer and da Silva, 2005), fMRi (Huettel et al., 2004) or intrusive methods like implanted optical fibres (Becker and Person, 2019).

However, our minimal and theoretical model employs a different method of determining the cognitive cost. Neural computation which processes sensory inputs to make a decision which then in turn is communicated to the actuators of the agent can be directly translated to a message sent from the sensors to the actuators (Tanaka and Sandberg, 2015). This opens the way to use information theory as the basis to measure the information flow through the agent's perception-action loop which can be interpreted as the cognitive cost of any agent — theoretical or not (Polani, 2009). The main objective in our work is the consideration of utility-optimizing behaviours in the MDP while respecting the secondary objective of minimizing this cognitive cost. In general, one can further reduce cognitive cost by trading in some utility (Polani et al., 2006). However, for simplicity, we focus entirely on optimal policies.

Specifically, we will use the formalism of *relevant information* to measure the cognitive cost of the agent to control its movement (Polani et al., 2006). Relevant information for an MDP is defined as the minimal information required about the current state to select an action to achieve a given utility. It represents a lower bound of how much cognitive cost per decision is required to achieve a given utility. Formally, relevant information is defined as

$$\min_{\pi(A|S) s.t. E^{\pi}[Q(s,a)] \stackrel{!}{=} \bar{Q}} I(S;A) \tag{3}$$

i.e. as the minimum amount of information the actuators A use about the state S as to achieve a given utility, with $\bar{Q}$ being the desired utility of the MDP. We will typically choose $\bar{Q}$ as Q^*, the optimal utility. By introducing a Lagrangian factor β, this constrained minimization can be converted into the unconstrained minimization:

$$\min_{\pi} \big(I(S;A) - \beta E[Q(S,A)] \big) \tag{4}$$

In this paper we will focus on achieving the optimal values only, e.g. the Lagrangian 4 will be considered for β tending towards the infinite limit. We further exclude the goal states from the calculation of the mutual information $I(S;A)$, as these are at the end of an episode and contribute no relevant decision in the policy. Since, $I(S;A)$ is a concave function of $p(s)$ for a fixed $p(a|s)$ and a convex function of $p(a|s)$ for a fixed $p(s)$, the relevant information minimization is formally equivalent to the standard rate-distortion problem known from information theory (Cover and Thomas, 1991) with a different fixed point. The rate-distortion problem is solved with the Blahut-Arimoto fixed-point iteration algorithm (which implements in essence a sequence of mutual projections between two convex sets, see (Cover and Thomas, 1991). We use practically the same algorithm here, replacing the information-theoretical distortion of a signal by the optimal utility Q^* (Polani et al., 2006). [2]

Note that the naive use of Blahut-Arimoto in the present context is only possible since Q^* does not depend on the policy but only on the MDP. When one considers the general case of suboptimal policies, such a simplification is no longer possible (see Polani et al., 2006; Polani, 2009) for details).

Crucially, this optimization gives us two results: The minimal information cost the agent has to pay per step to ensure it reaches the goal with perfect cost expenditure and the policy with which to achieve this. Thus the informational cost of a particular setup which in turn is used to compare the overall cognitive cost of kinematic/location-based agent movement and the acceleration/velocity-based agent.

Model Comparison

We now compare the two presented models (K/L and V/A) directly and use the two goal types of the velocity/acceleration model(V/A model) using two different goal sets $\{p_g\} \times V$) and $(p_g, 0)$. s We will look at a "border goal"

[2] to see the equivalence with the rate-distortion problem, note that our regret $Q^*(s,a^*) - Q^*(s,a)$ here effectively acts as a distortion, where s is the sent symbol, a^* is the desired transmitted symbol — the correct action — and a the actually received symbol — the actually chosen action.

as one extreme and the goal in the middle as the most general non-border position on the other end of the spectrum. Throughout these experiments we will limit the maximum velocity to one. Further acceleration is possible but does not affect the velocity. Thus, the agent in both models can at most move one position (to the left or right) in each time step.

The agent starts randomly in one of the non-goal states of the MDP. This means in the V/A model the agent can start with a velocity. This of course has an effect on its performance as it may already be on track to the goal or needs to decelerate first. There is no counterpart for this in the Kinematic/location-based (K/L) model.

Since the reward function is entirely based on the amount of time passed, no adjustments are needed. Note that the agent starting with zero velocity will be one time step slower to reach the goal in the V/A case compared to the K/L case. Thus, the comparison is not about directly analyzing the exact performance or specific information cost but rather to investigate the general behavior change of the agent within the proposed V/A model as compared to the K/L model, as well as identifying exactly when behavioral changes take place and where cognitive costs are incurred.

Result 1 — border goal with fully trapping goal position

We observe no significant difference in the behavior of the policy or the information cost between the K/L and the V/A model. As shown in figure 1, both policies only include a single action ($m = +1$ and $a = +1$) in all possible states, resulting in a relevant information of 0 since no states need to be distinguished from others to decide on an action to take.

We observe that none of the agents selects the "no change"-action. In the case of the K/L model this action is suppressed by the optimality requirement because using this action would mean a "lost" time step and thus suboptimal performance. In the acceleration-based model however, the "no change"-action appears in some optimal policies if one purely optimizes in terms of value (e.g. the MDP solution with Q^*). When additionally optimizing under the relevant information constraint, this constraint reduces the set of possible optimal policies; the policy with the "no change"-action is now suppressed in favour of the policy shown in figure 1 since the former would require distinguishing a state on whether to apply $a = +1$ or $a = 0$ whereas the latter does not.

Result 2 — middle-of-the-field goal with fully trapping goal position

We see the same general behaviour as in the first setup but now the goal can be reached from two directions (see Figure 2). Both agents directly move towards the goal from their respective side. The agent now needs to distinguish at every

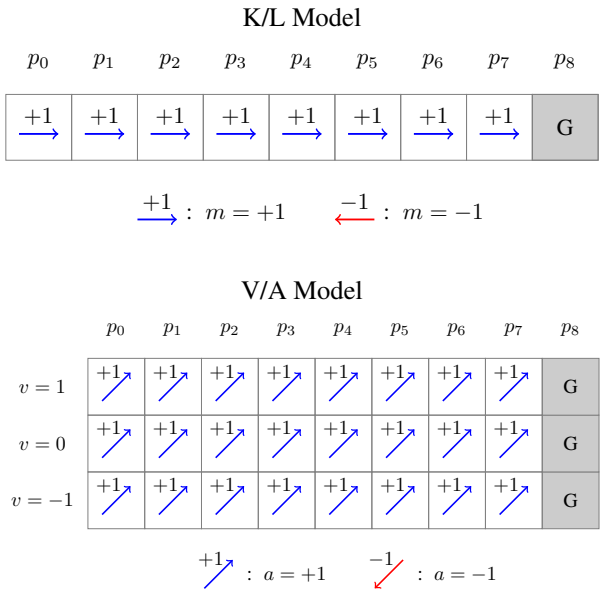

Figure 1: Top: The resulting policies of the K/L model in terms of m in which the arrows symbolize the action to take one step in the marked direction. This policy only contains a single action $m = +1$, with an information of 0 bit per step and the goal marked with the letter "G". Bottom: The resulting policy of the V/A model in terms of a with again an information of 0 bit per step and the goal marked with the letter "G". Here, the arrow indicates the immediate change in velocity and the delayed change in location induced by the action ($a = +1$).

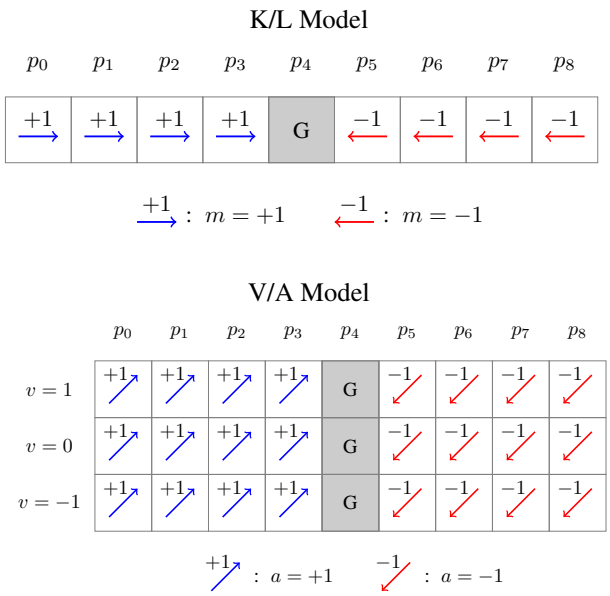

Figure 2: Top: The resulting policy of the K/L model in terms of m in which the arrows symbolize the action to take one step in the marked direction. The policy shows two equally large sets of states (1 bit per step) in which the same action is taken and the goal marked with the letter "G". Bottom: The resulting policy of the velocity/acceleration model in terms of a which also shows two equally large sets of states (1 bit per step) and the goal marked with the letter "G". Here, the arrow indicates the immediate change in velocity and the delayed change in location induced by the action.

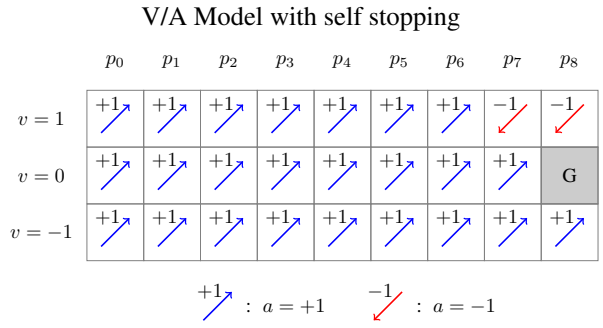

Figure 3: The resulting policy of the acceleration-based model with self stopping and an informational cost of > 0 bit per step and the goal marked with the letter "G". There two states which require a different action than the rest.

time its state amongst two equally large sets of states which results in a relevant information of 1 bit per step. (We note here that the relevant information formalism used here assumes that the agent has no memory, so it has to "look up" its state at each decision point).

Result 3 — border goal with active stopping

Here, we see for the first time a specific behavior of deceleration and its cost near the goal because the latter can only be reached with zero velocity. Far away from the goal the agent behaves the same in all states but, once near the goal, it has to decelerate (see Figure 3). We see now a slight increase in relevant information compared to the 0 in the previous fully trapping border goal. This increase results from the two states $(p_7, +1)$ and $(p_8, +1)$ in the top right corner which require the action $a = -1$ to decelerate the agent while in the rest of the states the action $a = -1$ is taken. The exact value of information depends on the number of states because the relevant information is an average over all states. In this particular example the relevant information is 0.39 bit per step. Importantly, the agent now does not only need to move towards the goal but needs also to plan (slightly) ahead to arrive and stop once reaching the goal position.

Result 4 — middle goal with active stopping

We observe a combination of the behaviors in setups 2 and 3. Again observe that around the goal the state space is partitioned into two behaviors, now not purely based on the position but also on the velocity. In contrast to the K/L model (see Figure 2), however, we do not have the same action overall on one side of the goal (see Figure 4). Instead, we can see that the process of decelerating timely is more difficult in this scenario.

From these experiments we see that velocity-based movements only show differences in strategy if the responsibility to stop is placed on the agent itself.

V/A Model with self stopping

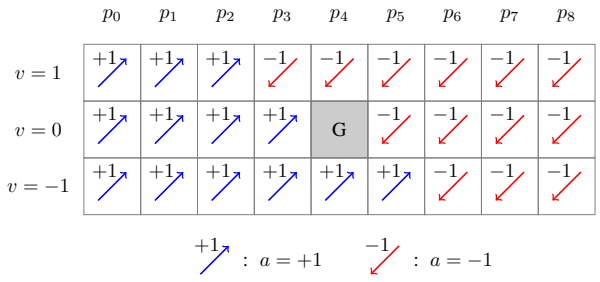

Figure 4: The resulting policy with active deceleration. This policy also has only two equally large sets of states (1 bit per step) and the goal marked with the letter "G". However, we clearly see that the area around the goal is the important part.

Increasing the Maximum Velocity

In our previous experiments we limited the possible velocities in our V/A model to create agent trajectories comparable to the K/L model. In the following experiments we now allow higher velocities and thus faster movements. This has no counterpart in the K/L model without expanding the model significantly.

Again, we investigate the cost of stopping at the right position e.g. $(p_g, 0)$. The agent again starts randomly in one of the non-goal states of the MDP.

In the first experiment we restrict the agent to a maximum velocity of 2 in both directions and in the second we discuss the theoretical case of no restriction to velocity. To avoid dealing with infinite state spaces in our simple framework, we consider various maximum velocities $v_{max} = k$ where $k \in \mathbb{N}$. The agent can still only accelerate or decelerate by 1 at each time step which means it may have to overshoot the goal or hit walls if it starts with a too high velocity at the wrong location.

	Setup: Goal	Setup: vel_{max}	Setup: Self Stopping	Relevant Information	Braking Distance	"No Change" utilised
K/L	Border	-	-	0 bit	0	no
	Middle	-	-	1 bit	0	no
V/A Model	Border	1	No	0 bit	0	no
	Middle	1	No	1 bit	0	no
	Border	1	Yes	0.39 bit	1	no
	Middle	1	Yes	1 bit	1	no
	Middle	2	Yes	1.06 bit	3	yes
	Middle	k	Yes	1-1.5 bit	$\frac{k(k+1)}{2}$	yes

Table 1: Cognitive cost, in relevant information, braking distance, and use of the "No-Change" action for all tested setups.

V/A Model with max velocity 2 and self stopping

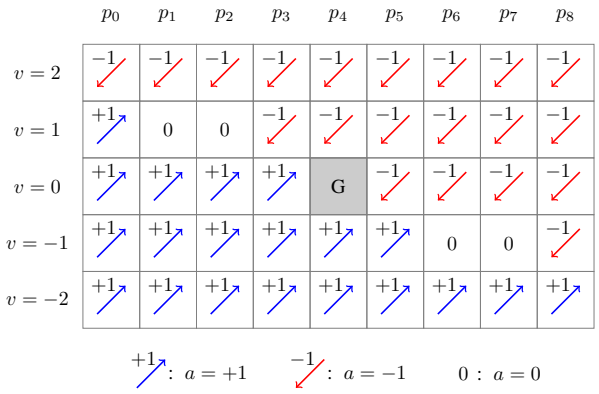

Figure 5: The resulting policy of an agent with a maximum velocity 2 and the goal marked with the letter "G". The cost of this policy is more than 1 bit per step because the "no change"-action (0) is the only optimal action in four states.

Result 5 — setting the maximal velocity v_{max} to 2

We see an extension of the effects in experiment 4. The larger the velocity, the further away from the goal the deceleration process needs to be initiated to avoid overshooting the goal. We observe for the first time the necessity of a "no change"-action $a = 0$ to achieve optimal cost. In the states $(p_2, +1)$, $(p_1, +1)$, $(p_6, -1)$ and $(p_7, -1)$ in which the agent is two positions away and already moves towards the goal, the agent can neither accelerate nor decelerate without wasting time but rather has to keep its velocity steady (shown as 0 in Figure 5). This further increases the cognitive cost of deceleration and managing velocity. As a result, the policy has a relevant information of more than 1 bit per time step, again the exact increase depends on the amount of positions in the world as positions further away from the goal are not affected. For this particular setup the relevant information is 1.06 bit per step.

In summary, this experiment shows that stopping with higher velocity requires measurably more complex decision-making.

Result 6 — larger maximal velocities v_{max}

We again see a continuation of the effect in experiment 5. Higher velocities require even earlier deceleration and planning (see Figure 6 resulting in a relevant information of 1.31 for the shown example. In fact, the agent needs to start decelerating $\sum_{i=1}^{k} i = \frac{k(k+1)}{2}$ positions away from the goal, where k is the current velocity. The agent can only reduce its velocity by 1 each time step but will still be moving towards the goal, in other words the braking distance of the agent is that long. The "no change"-action appears more often and on all levels of velocity in a specific pattern: 2 state for velocity 1, 3 for velocity 2, 4 for velocity 3 and so on

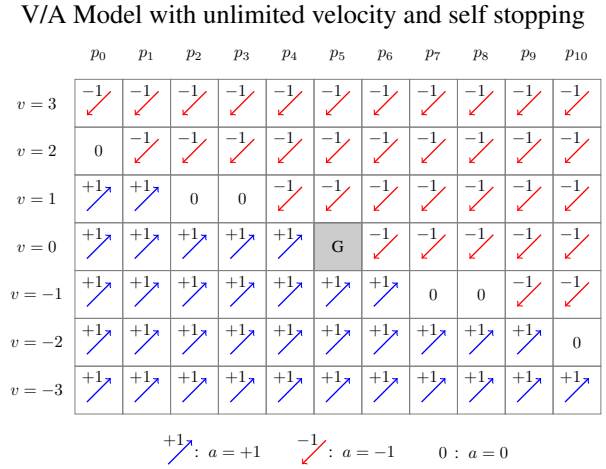

Figure 6: Part of the policy with unlimited velocities showing the longer deceleration phase and the pattern of "no-change"-actions($a = 0$) necessary to reach the goal optimally.

directly before the position in which the deceleration phase starts. For a velocity of 2 there are three states in which the agent has to maintain its velocity in order to neither stop before reaching the goal position nor overshooting it. This increases to four states when the velocity is 3 and continues growing linearly with the velocity. The number of states in which the agent has to maintain its velocity is tied to the displacement the agent will experience before its next decision — e.g. before it can start the actual deceleration phase.

Discussion

We have introduced a model for movement or navigation of an agent that extends the typically studied perspective from a kinematic to a dynamic view. This proposed Velocity/Acceleration model (V/A model) expands the agent's state space to include its current velocity. In the V/A model, the actions are accelerations which directly affect only the velocity which in turn affects the position. We took this as a step to understand the agent's dynamics when it has to contend with momentum and inertia as opposed to the typical high-friction scenarios where this is not necessary.

In the first two experiments (trapping goals) we have seen that both models effectively function in the same fashion when it is the environment that is responsible for deceleration (e.g. $\{p_g\} \times V$) (compare row 1–4 in Table 1). Once we transfer this responsibility to the agent (Experiment 3 to 6, non-trapping), the agent needs to carry out a deceleration behavior. Around the goal position the policy shows a distinctive pattern of actions to generate this deceleration behavior. Our first minimalistic model offers a glimpse into how we can model movement and understand the recent findings by Becker and Person, investigating the neural activity of a mouse reaching for an object (Becker and Person, 2019). Their results showed that mice show an increase in neural activity — which we interpret as investment of cognitive processing cost — in the cerebellum while decelerating and correcting [3].

In experiments 5 and 6 our model predicts that agents with richer velocity spaces require more cognitive cost and planning. Perhaps the introduction of more semantic actions — decelerate to zero — via options (Sutton et al., 1999), scripts (Riegler et al., 2021) or subgoals (van Dijk and Polani, 2011) might be interesting approaches to reduce the cognitive cost at the decision-making level.

Maex and Steuber have suggested that specific neural circuits in the cerebellum are capable of mathematical integration (Maex and Steuber, 2013). This would theoretically provide the computational capabilities which would allow the integration-based forward planning when velocities are involved.

The idea of the present work is to explicitly consider the necessity to manage momentum and inertia and suggest possible consequences for the cognitive processing and possibly the brain structure of the respective biological organisms as compared to organisms that live in high-friction environments where they only manage positioning directly. In particular, we propose that information-processing considerations may directly suggest evolutionary pressures towards brain structures geared towards processing of particular types of movement control information and thus help contributing to the prediction of the presence and functionality of certain components of the brain across organisms.

References

Abbeel, P., Coates, A., Quigley, M., and Ng, A. (2006). An application of reinforcement learning to aerobatic helicopter flight. *Advances in neural information processing systems*, 19.

Becker, M. I. and Person, A. L. (2019). Cerebellar control of reach kinematics for endpoint precision. *Neuron*, 103(2):335–348.

Boubaker, O. (2013). The inverted pendulum benchmark in nonlinear control theory: a survey. *International Journal of Advanced Robotic Systems*, 10(5):233.

Cover, T. M. and Thomas, J. A. (1991). Information theory and statistics. *Elements of information theory*, 1(1):279–335.

Furuta, K., Yamakita, M., and Kobayashi, S. (1992). Swing-up control of inverted pendulum using pseudo-state feedback. *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, 206(4):263–269.

Huettel, S. A., Song, A. W., McCarthy, G., et al. (2004). *Functional magnetic resonance imaging*, volume 1. Sinauer Associates Sunderland, MA.

[3] Specifically, Figure 4 of Becker and Person's paper show the connection of outward velocity and neural activity

Kajita, S., Kanehiro, F., Kaneko, K., Fujiwara, K., Harada, K., Yokoi, K., and Hirukawa, H. (2003). Biped walking pattern generation by using preview control of zero-moment point. In *2003 IEEE International Conference on Robotics and Automation (Cat. No.03CH37422)*, volume 2, pages 1620–1626 vol.2.

Laughlin, S. B. (2001). Energy as a constraint on the coding and processing of sensory information. *Current opinion in neurobiology*, 11(4):475–480.

Maex, R. and Steuber, V. (2013). An integrator circuit in cerebellar cortex. *European Journal of Neuroscience*, 38(6):2917–2932.

Mitobe, K., Capi, G., and Nasu, Y. (2000). Control of walking robots based on manipulation of the zero moment point. *Robotica*, 18(6):651–657.

Niedermeyer, E. and da Silva, F. L. (2005). *Electroencephalography: basic principles, clinical applications, and related fields*. Lippincott Williams & Wilkins.

Polani, D. (2009). Information: currency of life? *HFSP journal*, 3(5):307–316.

Polani, D., Nehaniv, C. L., Martinetz, T., and Kim, J. T. (2006). Relevant information in optimized persistence vs. progeny strategies. In *In: Artificial Life X: Proceedings of the Tenth International Conference on the Simulation and Synthesis of Living Systems*. Mit Press.

Riegler, B., Polani, D., and Steuber, V. (2021). Embodiment and its influence on informational costs of decision density—atomic actions vs. scripted sequences. *Frontiers in Robotics and AI*, 8:24.

Shalev-Shwartz, S., Shammah, S., and Shashua, A. (2016). Safe, multi-agent, reinforcement learning for autonomous driving. *CoRR*, abs/1610.03295.

Sutton, R. S. and Barto, A. G. (2018). *Reinforcement learning: An introduction*. MIT press.

Sutton, R. S., Precup, D., and Singh, S. (1999). Between mdps and semi-mdps: A framework for temporal abstraction in reinforcement learning. *Artificial intelligence*, 112(1-2):181–211.

Tanaka, T. and Sandberg, H. (2015). Sdp-based joint sensor and controller design for information-regularized optimal lqg control. In *2015 54th IEEE Conference on Decision and Control (CDC)*, pages 4486–4491. IEEE.

van Dijk, S. G. and Polani, D. (2011). Grounding subgoals in information transitions. In *2011 IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning (ADPRL)*, pages 105–111. IEEE.