

# Helping Humans Drive Better: Building Provably-Correct Driver Assistance Systems using Cognitive Modelling



Francisco Girbal Eiras  
Linacre College, University of Oxford

Supervised by:  
Dr. Morteza Lahijanian and Prof. Marta Kwiatkowska

A thesis submitted for the degree of  
*Master of Science in Computer Science*

Trinity 2018



## Acknowledgements

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Pellentesque accumsan magna eu metus congue, vitae tristique enim commodo. Aliquam consectetur porttitor nisl, eget pulvinar quam venenatis eget. Aliquam erat volutpat. Nullam at ipsum at felis tincidunt consequat non sed mi. Morbi placerat iaculis dapibus. Aliquam non risus ac urna interdum pretium. Donec maximus, massa vitae suscipit congue, ligula purus sagittis magna, vitae interdum enim dui sed felis. Curabitur mollis eleifend diam a varius. Vivamus eu porttitor lorem. Mauris sagittis ultrices ligula, a euismod mi efficitur a. Aenean molestie sapien a nibh suscipit, nec venenatis urna varius. Sed eu ex id lacus tincidunt ultricies. Aenean quam massa, dignissim ac suscipit posuere, varius sit amet turpis. Nunc semper turpis ullamcorper elit venenatis, in molestie nulla fermentum.

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Literature Review and Definitions</b>	<b>2</b>
2.1	Cognitive Architectures . . . . .	2
2.1.1	Adaptive Control of Thought - Rational (ACT-R) . . . . .	3
2.2	Integrated Driver Modelling in ACT-R . . . . .	4
2.3	Probabilistic Model Checking . . . . .	7
2.3.1	Markov Chains and Decision Processes . . . . .	8
2.3.2	Temporal Logics . . . . .	11
2.3.3	PRISM Language . . . . .	15
2.3.4	Model Checking in DTMCs and MDPs . . . . .	17
<b>3</b>	<b>Human Driver Modelling</b>	<b>24</b>
3.1	Continuous Driver Model using ACT-R . . . . .	24
3.2	Model Abstraction . . . . .	30
3.2.1	Non-Probabilistic Control Module . . . . .	30
3.2.2	Decision Making and Monitoring Module . . . . .	30
3.2.3	Probabilistic Control Module . . . . .	31
3.3	Model Evaluation Metrics . . . . .	31
3.3.1	Completeness Property . . . . .	32
3.3.2	Safety Property . . . . .	32
3.3.3	Liveness Properties . . . . .	33
3.4	Simulation of Paths in the Model . . . . .	33
<b>4</b>	<b>Advanced Driver Assistance Systems</b>	<b>34</b>
4.1	Driver Assistance System Design . . . . .	34
4.1.1	Decision Making based ADAS . . . . .	34

4.1.1.1	Decision Making with Fully Compliant Drivers . . .	35
4.1.1.2	Decision Making with Partially Compliant Drivers .	35
4.1.2	Control based ADAS . . . . .	36
4.1.2.1	Active Linear Acceleration Control . . . . .	36
4.1.2.2	Active Steering Control . . . . .	36
4.2	Evaluation . . . . .	37
4.2.1	Multi-Objective Metrics . . . . .	37
4.2.2	Test Cases . . . . .	38
4.2.2.1	$v = 21$ m/s, $v_1 = 19$ m/s, $x_{1,0} = 70$ m . . . . .	38
4.2.2.2	$v = 30$ m/s, $v_1 = 22$ m/s, $x_{1,0} = 50$ m . . . . .	38
4.2.3	Overall Comparison . . . . .	38
<b>5</b>	<b>Experimental Results</b>	<b>40</b>
5.1	Safety Metrics . . . . .	40
5.2	Liveness Metrics . . . . .	41
5.3	Discussion . . . . .	41
5.4	Additional Experiments . . . . .	42
<b>6</b>	<b>Conclusions and Future Work</b>	<b>43</b>
<b>A</b>	<b>Code for Human Driver Modelling</b>	<b>44</b>
A.1	ACT-R Implementation (Matlab) . . . . .	44
A.1.1	Main Loop . . . . .	44
A.1.2	Control Module . . . . .	44
A.1.3	Decision Making Module . . . . .	45
A.1.4	Monitoring Module . . . . .	45
A.2	Control Abstraction (Matlab) . . . . .	46
A.3	Decision Making and Monitoring Abstraction (Matlab) . . . . .	46
A.4	Probabilistic Two Module Model Generator (Python) . . . . .	47
A.5	Prism Automatic Model Checker (Python) . . . . .	47
A.6	Storm Automatic Model Checker (Python) . . . . .	47
A.7	Simulator (Python) . . . . .	48
<b>B</b>	<b>Code for the Advanced Driver Assistance Systems</b>	<b>49</b>
B.1	MDP Model Generators for Different ADAS (Python) . . . . .	49
B.1.1	Perfect Decision Making . . . . .	49

B.1.2	Imperfect Decision Making . . . . .	50
B.1.3	Additive Linear Control . . . . .	50
B.1.4	Additive Steering Control . . . . .	50
B.2	Pareto Curve Generator (Python) . . . . .	51
B.3	Strategy Synthesis and Simulator (Python) . . . . .	51
B.4	Prism Automatic Model Checker (Python) . . . . .	52
<b>C</b>	<b>Code for the Experimental Results and Evaluation</b>	<b>53</b>
C.1	Plot Generator for Human Driver Model (Python) . . . . .	53
C.2	Plot Generator for the Decision Making and Full Control ADAS (Python)	53
	<b>Bibliography</b>	<b>55</b>

# List of Figures

2.1	Overview of the ACT-R high level architecture (from [5]). . . . .	3
2.2	High level representation of the architecture presented in [17]. . . . .	4
2.3	Flowchart of part of the decision making process of the model. . . . .	7
2.4	Example of a Pareto curve represented as a 2D plot. . . . .	22
3.1	Continuous Driver Model overview. . . . .	25
3.2	Flowchart of the Monitoring module. . . . .	27
3.3	Flowchart of the Decision Making module. . . . .	28
3.4	Flowchart of the Control module. . . . .	29

# Chapter 1

## Introduction

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Pellentesque accumsan magna eu metus congue, vitae tristique enim commodo. Aliquam consectetur porttitor nisl, eget pulvinar quam venenatis eget. Aliquam erat volutpat. Nullam at ipsum at felis tincidunt consequat non sed mi. Morbi placerat iaculis dapibus. Aliquam non risus ac urna interdum pretium. Donec maximus, massa vitae suscipit congue, ligula purus sagittis magna, vitae interdum enim dui sed felis. Curabitur mollis eleifend diam a varius. Vivamus eu porttitor lorem. Mauris sagittis ultrices ligula, a euismod mi efficitur a. Aenean molestie sapien a nibh suscipit, nec venenatis urna varius. Sed eu ex id lacus tincidunt ultricies. Aenean quam massa, dignissim ac suscipit posuere, varius sit amet turpis. Nunc semper turpis ullamcorper elit venenatis, in molestie nulla fermentum.



# Chapter 2

## Literature Review and Definitions

This chapter presents the concepts and theory required to understand the methodology, rationale and relevance of this dissertation. It starts by defining a Cognitive Architecture, and goes into more detail about Adaptive Control of Thought - Rational, which is the underlying framework used in the Integrated Driver Modelling described in Section 2.2 and the basis for the system developed in Chapter 3: *Human Driver Modelling*. The chapter concludes with the introduction of several important concepts in the area of Probabilistic Model Checking, which are essential for modelling, verification and synthesis in this context.

### 2.1 Cognitive Architectures

Different definitions of a cognitive architecture can be found in the literature [10]. According to Sun, a cognitive architecture is a *"broadly-scoped, domain-generic computational cognitive model, capturing the essential structure and process of the mind, to be used for a broad, multiple-level, multiple-domain analysis of behavior"* [19]. In that sense, a cognitive architecture is no more than a formalised framework of the perception, memorisation, decision making, reasoning and execution of the human mind. It is important to note that this architecture should capture both the abilities (e.g. memory storage and recall, perception or motor action) and constraints (e.g. memory decay and limited motor performance) of the human system [17].

The idea of a cognitive architecture originated in the early 1970s. A survey by Koseruba and Tsotsos [10] revealed that, since then, an estimated 195 different cognitive

architectures with different assumptions, structural organisations and implementations have been developed. Applications of these range from the fields of Robotics to Natural Language Processing [10]. One of the most well known of those architectures is Adaptive Control of Thought - Rational.

### 2.1.1 Adaptive Control of Thought - Rational (ACT-R)

Adaptive Control of Thought - Rational (or ACT-R as it is commonly known) is a particular cognitive architecture first presented by John R. Anderson in 1983 [2] and further developed by Anderson *et al.* in [3].

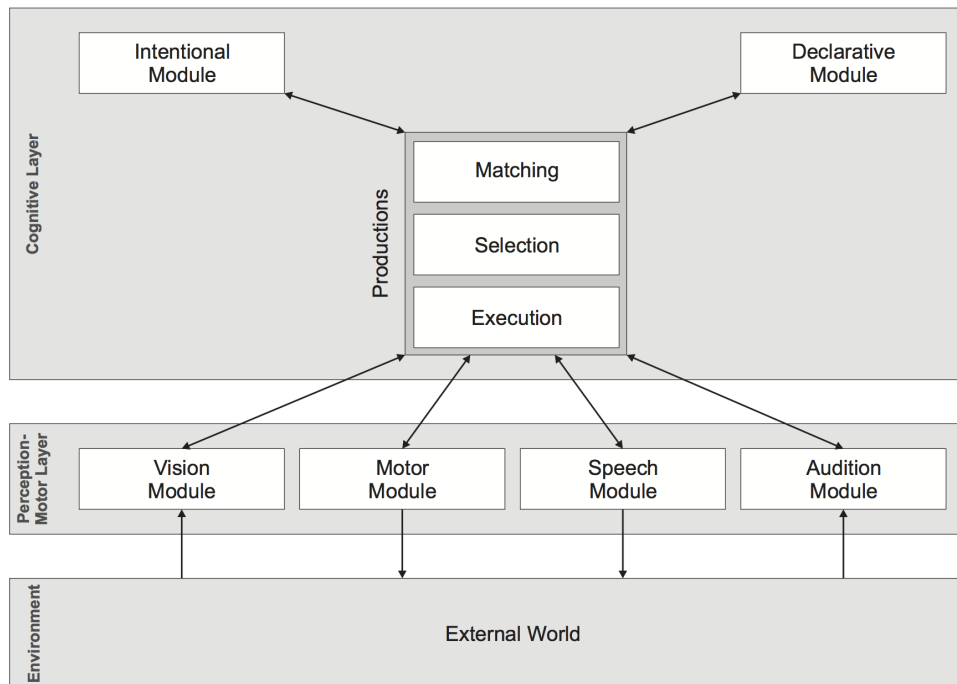


Figure 2.1: Overview of the ACT-R high level architecture (from [5]).

The architecture can be generally described as two distinct modules: a *perceptual-motor* layer and a *cognitive* layer, as presented in Figure 2.1. The perceptual-motor module corresponds to the interface of the cognition with the environment (which plays a key role in ACT-R), being comprised of sub-modules such as vision and motor actions. The cognitive layer is focused on memory, which can be divided into two different categories: *declarative* (consisting of factual knowledge and goals - e.g. "The maximum driving speed in a typical UK motorway is 70 mph" or "Try get to point B") and *procedural* (consisting of rules/procedures - e.g. "If the lead vehicle is going

*slowly, attempt an overtake*") [5]. Furthermore, declarative memory is composed of *chunks*, which are smaller pieces of information divided into categories or attributes that form more complex memories. In the example "*The maximum driving speed in a typical UK motorway is 70 mph*", **maximum driving speed in a typical UK motorway** is a category, whereas 70 mph is an attribute. Another possible attribute for this category could be 112 km/h (since  $70\text{mph} = 112\text{km/h}$ ).

## 2.2 Integrated Driver Modelling in ACT-R

The first integrated approaches to this problem, proposed in [15, 6, 18], rely on modelling human behaviour through continuous controllers and specifically engineered frameworks. However, these methods lack the human behaviour variability attributed to the discrete nature of the control actions performed by the drivers. With this in mind, in [16] Salvucci *et al.* proposed a proof-of-concept of the introduction of human constraints in the driver model through the use of the cognitive architecture ACT-R. In [17], Salvucci proposed an updated version of the human driver model initially introduced in [16], which was improved using the advances in ACT-R (the model uses version 5.0 of the architecture) and re-designed elements (e.g. lane changing behaviour or distraction) resulting from validation performed by empirical studies.

The model Salvucci introduced in [17] was developed with a specific scenario in mind: "*(...) driving a standard midsize vehicle on a multilane highway with moderate traffic*" [17]. It consists of three distinct modules interacting in a sequential way: **control**, **monitoring**, and **decision making**. A high level schematic of the model can be found in Figure 2.2.

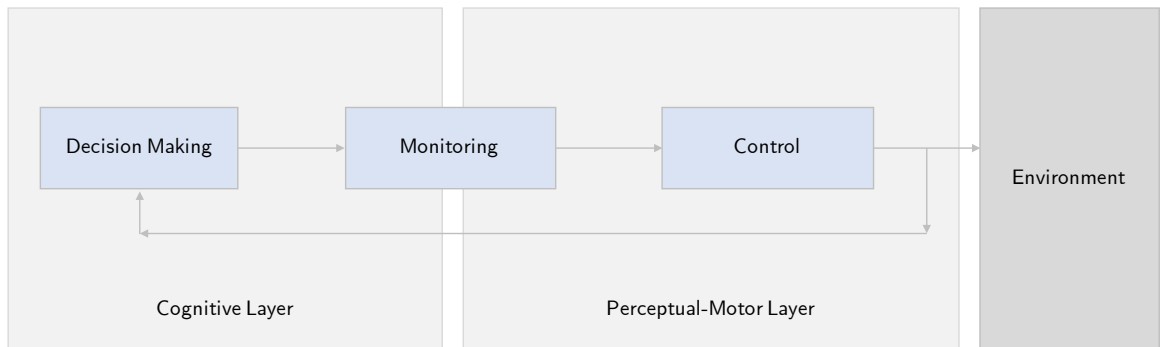


Figure 2.2: High level representation of the architecture presented in [17].

The **control** component manages both the lower level perception cues and the physical manipulation of the vehicle (e.g. steering, accelerating or breaking). The module can be divided into lateral (i.e. steering) and longitudinal (i.e. speed) control, each modelled by a separate control law.

The lateral control is determined by the existence of two artefacts that the driver obtains using lower level perception cues: the *near point* and the *far point*. In this model, *"the near point represents the vehicle's current lane position, used to judge how close the vehicle is to the center of the roadway (...) [and] is characterized as a point in the center of the near lane visible in front of the vehicle, set at a distance of 10 m from the vehicle's center"* while *"the far point indicates the curvature of the upcoming roadway, used to judge what the driver should execute to anticipate the upcoming curvature and remain in the current lane [17].* At any cycle  $i > 0$ , the model works by using perception to determine the visual angles  $\theta_{near}^{(i)}$  and  $\theta_{far}^{(i)}$  and calculates:

$$\begin{aligned}\Delta\theta_{near} &= \theta_{near}^{(i)} - \theta_{near}^{(i-1)} \\ \Delta\theta_{far} &= \theta_{far}^{(i)} - \theta_{far}^{(i-1)}\end{aligned}\tag{2.1}$$

The control law for the steering angle  $\varphi$  can be defined as:

$$\Delta\varphi = k_{far}\Delta\theta_{far} + k_{near}\Delta\theta_{near} + k_I \min(\theta_{near}^{(i)}, \theta_{max})\Delta t\tag{2.2}$$

with  $k_{far}$ ,  $k_{near}$  and  $k_I$  defined as in [17].

The process for the longitudinal control is very similar. At any cycle  $i > 0$ , the model starts by encoding the position of the lead vehicle and calculating the time headway  $thw_{car}^{(i)}$  to it. It then computes:

$$\Delta thw_{car} = thw_{car}^{(i)} - thw_{car}^{(i-1)}\tag{2.3}$$

The control law for the linear acceleration  $\psi$  can be written as:

$$\Delta\psi = k_{car}\Delta thw_{car} + k_{follow}(thw_{car} - thw_{follow})\Delta t\tag{2.4}$$

with  $k_{car}$ ,  $k_{follow}$  and  $thw_{car}$  defined as in [17].

These control laws can be applied in order to model the behaviour of the driver in terms of lane keeping and curve negotiation, and generalise to lane changing actions in a straightforward manner. In order to initiate a lane change, the driver just starts following the near and far points of the destination lane instead of the current lane, as Salvucci and Liu presented in [18].

The **monitoring** component maintains situational awareness through the awareness of the position of other vehicles around the driver's vehicle. It accomplishes this through a random-sampling, with probability  $p_{monitor}$ , of one of four areas: either the left or right lane, and either forward or backward (with equal probability). After deciding which area to sample (if any), the model uses visual perception to determine whether a vehicle is present or not. If it is, the distance, lane and direction are saved in ACT-R's declarative memory. The value of  $p_{monitor}$  is defined in [17].

Finally, the **decision making** module uses the information gathered in the monitoring and control stage (if any is available) to determine which tactical decision should be taken. In this model, the decision corresponds to determining when and where a lane change should occur. Salvucci describes this process in [17] as: *"If the driver's vehicle is in the right lane, the model checks the current time headway to the lead vehicle (if any); if the lead car time headway drops below a desired time headway  $thw_{pass}$ , the model decides to change lanes to pass the vehicle. If the driver vehicle is in the left lane, the model checks instead simply for the presence of a lead vehicle. If there is a lead vehicle, the model remains in the left lane (because this vehicle is also passing other vehicles); otherwise it decides to [try to] change lanes to return to the right lane"*. This process is summarised in the flowchart presented in Figure 2.3, and the value of  $thw_{pass}$  is defined in [17].

Even after this decision to change lanes is taken, the model must then determine whether it is actually safe to do so. In order to verify that it is, the model initially attempts to recall, from the declarative memory, the nearest vehicle to it in the destination lane. If one is recalled at a distance closer than  $d_{safe}$ , then the change is aborted. If not, then the vehicle performs a safety monitoring of the destination lane. If this monitoring does not observe a vehicle at a distance closer than  $d_{safe}$ , then the vehicle initiates the execution of the lane change. The parameter  $d_{safe}$  is as defined in [17].

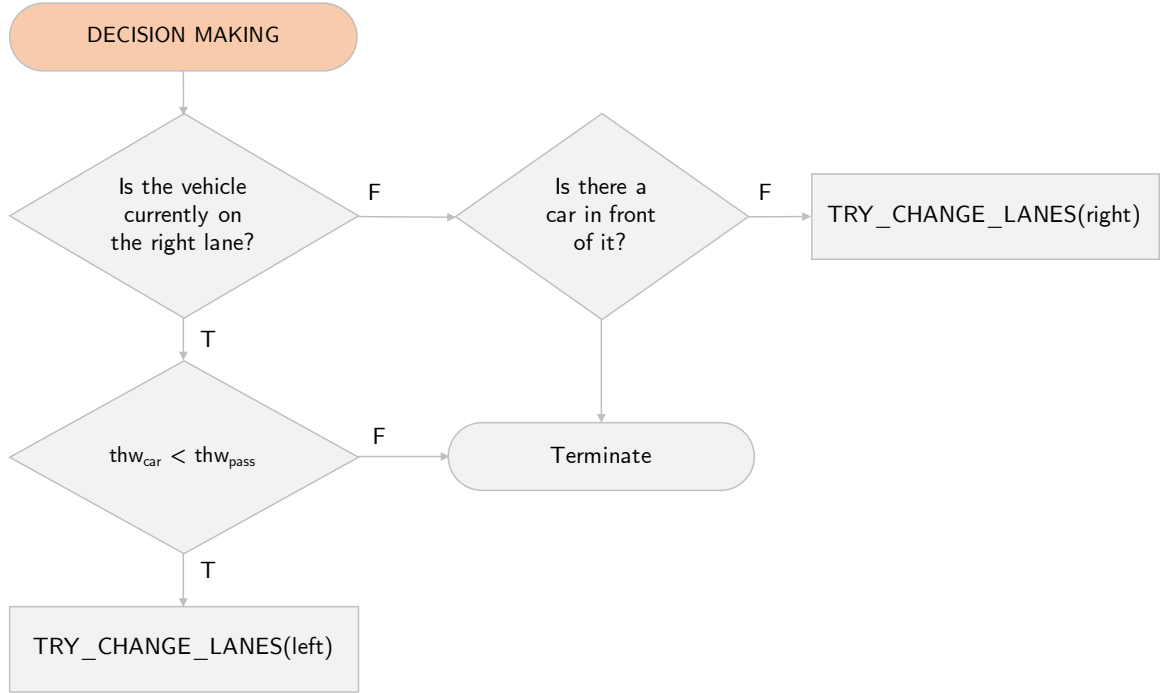


Figure 2.3: Flowchart of part of the decision making process of the model.

## 2.3 Probabilistic Model Checking

Model checking is a formal verification technique consisting of the process of automatically verifying, through exhaustive search, whether a model meets a given specification [4]. A model in this context is considered to be a labelled finite state-transition system, with states corresponding to configurations of the system and transitions between states representing the evolution between the said configurations.

Probabilistic model checking is a generalisation of model checking for the automated verification of systems that exhibit probabilistic behaviour [9, 4]. It should be noted that this is not to be confused with *probabilistic verification*, which amounts to partial state-space exploration [4]. Within this section, important concepts of probabilistic model checking related to modelling of systems, and verification and synthesis of models will be explained in detail.

### 2.3.1 Markov Chains and Decision Processes

Probabilistic model checking begins with the appropriate modelling of the system in a stochastic finite state-transition representation. There are several ways to do this according to the type of system to be modelled.

In *discrete-time Markov Chains* (DTMCs), all choices (i.e. transitions) are probabilistic [4]. While this is useful in some cases, others require nondeterminism, which is where *Markov Decision Processes* (MDPs) come in. In MDPs, nondeterministic probabilistic choices coexist. Essentially, a DTMC is an MDP with a uniquely determined probabilistic distribution [4]. Both DTMCs and MDPs can be augmented with reward structures, which correspond to real positive values that can be interpreted as bonuses or costs [4].

The formal definitions of DTMCs, MDPs and reward structures follow (adapted from [4, 9, 12]).

#### Definition 1. Discrete-time Markov Chain (DTMC)

A *discrete-time Markov chain* is a tuple  $\mathcal{M} = (S, \mathbf{P}, p_{init}, AP, L)$  such that:

- $S$  is a countable, non-empty set of states
- $\mathbf{P} : S \times S \rightarrow [0, 1]$  is a transition probability function such that for all states  $s \in S$ :

$$\sum_{s' \in S} \mathbf{P}(s, s') = 1 \quad (2.5)$$

- $p_{init} : S \rightarrow [0, 1]$  is the initial state distribution, such that:

$$\sum_{s \in S} p_{init}(s) = 1 \quad (2.6)$$

- $AP$  is a set of atomic propositions and  $L : S \rightarrow 2^{AP}$  is a labelling function.

$\mathcal{M}$  is denoted *finite* if  $S$  and  $AP$  are finite sets. ■

**Definition 2. Markov Decision Process (MDP)**

A *Markov decision process* is a tuple  $\mathcal{M} = (S, Act, \mathbf{P}, p_{init}, AP, L)$  such that:

- $S$  is a countable, non-empty set of states
- $Act$  is a set of actions
- $\mathbf{P} : S \times Act \times S \rightarrow [0, 1]$  is a transition probability function such that for all states  $s \in S$  and actions  $\alpha \in Act$ :

$$\sum_{s' \in S} \mathbf{P}(s, \alpha, s') \in \{0, 1\} \quad (2.7)$$

- $p_{init} : S \rightarrow [0, 1]$  is the initial state distribution, such that:

$$\sum_{s \in S} p_{init}(s) = 1 \quad (2.8)$$

- $AP$  is a set of atomic propositions and  $L : S \rightarrow 2^{AP}$  is a labelling function.

An action  $\alpha \in Act$  is enabled in a state  $s$  if, and only if,  $\sum_{s' \in S} \mathbf{P}(s, \alpha, s') = 1$ . Let  $Act(s)$  denote the set of actions enabled in a state  $s$ . It must be that for all states  $s \in S$ ,  $Act(s) \neq \emptyset$ . ■

**Definition 3. Reward Structures for DTMCs and MDPs**

A *reward structure* for a discrete-time Markov chain  $\mathcal{M} = (S, \mathbf{P}, p_{init}, AP, L)$  is a tuple  $\mathcal{C} = (\rho, \iota)$  such that:

- $\rho : S \rightarrow \mathbb{R}_{\geq 0}$  is a state reward function
- $\iota : S \times S \rightarrow \mathbb{R}_{\geq 0}$  is a transition reward function

A *reward structure* for a Markov decision process  $\mathcal{M} = (S, Act, \mathbf{P}, p_{init}, AP, L)$  is a tuple  $\mathcal{C} = (\rho, \iota)$  such that:

- $\rho : S \rightarrow \mathbb{R}_{\geq 0}$  is a state reward function



- $\iota : S \times Act \rightarrow \mathbb{R}_{\geq 0}$  is an action reward function

■

There are some additional concepts related to DTMCs and MDPs which are relevant to the understanding of temporal logics and model checking. These are the concepts of paths in Markov chains or decision processes, and adversary in a Markov decision process (adapted from [4, 9]).

**Definition 4. Paths in DTMCs and MDPs**

An (*infinite*) *path*  $\pi$  in a discrete-time Markov chain  $\mathcal{M} = (S, \mathbf{P}, p_{init}, AP, L)$  is defined as an infinite sequence of states  $\pi = s_0 \rightarrow s_1 \rightarrow s_2 \dots$  (also written as  $\pi = s_0 s_1 s_2 \dots$ ) such that  $\forall i \geq 0 : \mathbf{P}(s_i, s_{i+1}) > 0$ .

A *finite path*  $\rho$  in the discrete-time Markov chain  $\mathcal{M}$  is defined as the prefix of an infinite path  $\pi$ ,  $\rho = s_0 \rightarrow s_1 \rightarrow \dots \rightarrow s_n$  for  $n > 0$ .

An (*infinite*) *path*  $\pi$  in a Markov decision process  $\mathcal{M} = (S, Act, \mathbf{P}, p_{init}, AP, L)$  is defined as an infinite sequence of states and actions  $\pi = s_0 \xrightarrow{a_0} s_1 \xrightarrow{a_1} s_2 \dots$  (also written as  $\pi = s_0 a_0 s_1 a_1 s_2 \dots$ ) such that  $\forall i \geq 0 : \mathbf{P}(s_i, a_i, s_{i+1}) > 0$ .

A *finite path*  $\rho$  in the Markov decision process  $\mathcal{M}$  is defined as the prefix of an infinite path  $\pi$ ,  $\rho = s_0 \xrightarrow{a_0} s_1 \xrightarrow{a_1} \dots \xrightarrow{a_{n-1}} s_n$  for  $n > 0$ .

In both Markov chains and decision processes, the set of all infinite paths is denoted by  $Paths(\mathcal{M})$ , while the set of finite paths is denoted by  $Paths_{fin}(\mathcal{M})$  ■

**Definition 5. Adversary (alternatively Strategy or Scheduler)**

An *adversary* for an MDP  $\mathcal{M} = (S, Act, \mathbf{P}, p_{init}, AP, L)$  is a function  $\sigma : S^+ \rightarrow Act$  such that  $\sigma(s_0 s_1 \dots s_n) \in Act(s_n)$  for all  $s_0 s_1 \dots s_n \in S^+$ .

An adversary  $\sigma$  is called *memoryless* if, and only if, for any  $\pi_1 = s_0 s_1 \dots s_n$  and  $\pi_2 = s'_0 s'_1 \dots s_n$  it is true that  $\sigma(\pi_1) = \sigma(\pi_2) = \sigma(s_n)$ . ■

### 2.3.2 Temporal Logics

In order to verify properties in models, there needs to be a way of expressing them formally and unequivocally. While some reachability and reward-based properties come directly from the definition of DTMCs, MDPs and reward structures [9], more interesting and complex properties require the use of a temporal logic. A propositional temporal logic is essentially an extension of propositional logic by temporal operators [4]. In this context, one can consider time as being *linear* or *branching*. In linear time, at every moment in time there must only exist a single successor moment, whereas from the branching perspective, there may exist multiple ones, forming a tree-like structure of alternatives. Linear Temporal Logic (LTL) is a temporal logic which assumes the first, while Computation Tree Logic (CTL) is based on the branching perspective. Probabilistic Computational Tree Logic (PCTL) is an extension of CTL that takes into account a probabilistic operator.

Given the definition of a path in an MC and MDP, it is possible to define the syntax for LTL, CTL and PCTL (adapted from [4, 9]).

#### Definition 6. Syntax and Semantics of LTL

An *LTL formulae*  $\varphi$  over a set of atomic propositions  $AP$  can be formed according to the following grammar:

$$\varphi ::= \mathbf{true} \mid a \mid \varphi_1 \wedge \varphi_2 \mid \neg\varphi \mid X\varphi \mid \varphi_1 \mathbf{U} \varphi_2 \quad (2.9)$$

where  $a \in AP$ ,  $X$  is the next operator and  $\mathbf{U}$  is the until operator as defined below for Markov chains and decision processes.

For a given path  $\pi = s_0s_1s_2\dots$  in a discrete-time Markov chain  $\mathcal{M} = (S, \mathbf{P}, p_{init}, AP, L)$  (generalising trivially to a Markov decision process):

- $\pi \models \mathbf{true}$  always.
- $\pi \models a$  if, and only if,  $a \in L(s_0)$ .
- $\pi \models \varphi_1 \wedge \varphi_2$  if, and only if,  $\pi \models \varphi_1 \wedge \pi \models \varphi_2$ .
- $\pi \models \neg\varphi$  if, and only if,  $\pi \not\models \varphi$ .

- $\pi \models X\varphi$  if, and only if,  $s_1s_2\ldots \models \varphi$ .
- $\pi \models \varphi_1 \text{ U } \varphi_2$  if, and only if:

$$\exists i \geq 0 \text{ s.t. } (s_i s_{i+1} \ldots \models \varphi_2) \wedge (\forall j < i : s_j s_{j+1} \ldots \models \varphi_1) \quad (2.10)$$

The operator F (eventually) can also be defined as:

$$F\varphi = \text{true} \text{ U } \varphi \quad (2.11)$$

■

It can be noticed that, as a result of the linear time consideration in LTL, all the formulas are taken over paths. This is not the case in CTL which can have both path and state formulae, as shown below.

### Definition 7. Syntax and Semantics of CTL

A CTL state formulae  $\Phi$  over a set of atomic propositions  $AP$  can be formed according to the following grammar:

$$\Phi ::= \text{true} \mid a \mid \Phi_1 \wedge \Phi_2 \mid \neg\Phi \mid E\varphi \mid A\varphi \quad (2.12)$$

where  $a \in AP$  and  $\varphi$  is a path formula.

For a given state  $s \in S$  in a discrete-time Markov chain  $\mathcal{M} = (S, \mathbf{P}, p_{init}, AP, L)$  (generalising trivially to a Markov decision process):

- $s \models \text{true}$  always.
- $s \models a$  if, and only if,  $a \in L(s)$ .
- $s \models \Phi_1 \wedge \Phi_2$  if, and only if,  $s \models \Phi_1 \wedge s \models \Phi_2$ .
- $s \models \neg\Phi$  if, and only if,  $s \not\models \Phi$ .
- $s \models E\varphi$  if, and only if,  $\exists \pi \in Paths(s) : \pi \models \varphi$ .

- $s \models A\varphi$  if, and only if,  $\forall \pi \in Paths(s) : \pi \models \varphi$

A *CTL path formulae*  $\varphi$  over a set of atomic propositions  $AP$  can formed according to the following grammar:

$$\varphi ::= X\Phi \mid \Phi_1 \cup \Phi_2 \quad (2.13)$$

where  $a \in AP$  and  $\Phi$  is a state formula.

For a given path  $\pi = s_0s_1s_2\dots$  in a Markov chain  $\mathcal{M} = (S, \mathbf{P}, p_{init}, AP, L)$  (generalising trivially to a Markov decision process):

- $\pi \models X\Phi$  if, and only if,  $s_1 \models \Phi$ .
- $\pi \models \Phi_1 \cup \Phi_2$  if, and only if:

$$\exists i \geq 0 \text{ s.t. } (s_i \models \Phi_2) \wedge (\forall j < i : s_j \models \Phi_1) \quad (2.14)$$

It is also possible to define the operators F (eventually) and G (globally) as:

$$\begin{aligned} F\Phi &= \text{true} \cup \Phi \\ G\Phi &= \neg E(\neg\Phi) \end{aligned} \quad (2.15)$$

■

### Definition 8. Syntax and Semantics of PCTL

A *PCTL state formulae*  $\Phi$  over a set of atomic propositions  $AP$  can formed according to the following grammar:

$$\Phi ::= \text{true} \mid a \mid \Phi_1 \wedge \Phi_2 \mid \neg\Phi \mid \mathbb{P}_{\triangleright p}(\varphi) \quad (2.16)$$

where  $a \in AP$ ,  $p \in [0, 1]$ ,  $\triangleright \in \{>, <, \geq, \leq\}$  is a probability bound and  $\varphi$  is a path formula.

For a given state  $s \in S$  in a discrete-time Markov chain  $\mathcal{M} = (S, \mathbf{P}, p_{init}, AP, L)$  (generalising trivially to a Markov decision process):

- $s \models \mathbf{true}$  always.
- $s \models a$  if, and only if,  $a \in L(s)$ .
- $s \models \Phi_1 \wedge \Phi_2$  if, and only if,  $s \models \Phi_1 \wedge s \models \Phi_2$ .
- $s \models \neg\Phi$  if, and only if,  $s \not\models \Phi$ .
- $s \models \mathbb{P}_{\triangleright p}(\varphi)$  if, and only if,  $P_s(\pi \in Paths(s) : \pi \models \varphi) \triangleright p$  (that is, the probability, from state  $s$ , that  $\varphi$  is true for an outgoing path satisfies  $\triangleright p$ ).

A *PCTL path formulae*  $\varphi$  over a set of atomic propositions  $AP$  can be formed according to the following grammar:

$$\varphi ::= X\Phi \mid \Phi_1 \text{ U}^{\leq k} \Phi_2 \mid \Phi_1 \text{ U } \Phi_2 \quad (2.17)$$

where  $a \in AP$ ,  $k \in \mathbb{N}$  and  $\Phi$  is a state formula.

For a given path  $\pi = s_0 s_1 s_2 \dots$  in a discrete-time Markov chain  $\mathcal{M} = (S, \mathbf{P}, p_{init}, AP, L)$  (generalising trivially to a Markov decision process):

- $\pi \models X\Phi$  if, and only if,  $s_1 \models \Phi$ .
- $\pi \models \Phi_1 \text{ U}^{\leq k} \Phi_2$  if, and only if:

$$\exists i \leq k \text{ s.t. } (s_i \models \Phi_2) \wedge (\forall j < i : s_j \models \Phi_1) \quad (2.18)$$

- $\pi \models \Phi_1 \text{ U } \Phi_2$  if, and only if:

$$\exists i \geq 0 \text{ s.t. } (s_i \models \Phi_2) \wedge (\forall j < i : s_j \models \Phi_1) \quad (2.19)$$

Similarly to CTL, it is possible to define the operators F (eventually) and G (globally) as:

$$\begin{aligned} F\Phi &= \mathbf{true} \text{ U } \Phi \\ G\Phi &= \neg E(\neg\Phi) \end{aligned} \quad (2.20)$$

■

While PCTL path formulas are defined above, it should be mentioned that all PCTL formulas are state formulas, and that path formulas should only appear inside the probabilistic operator.

In relation to the probabilistic operator in PCTL, the generalisation from the Markov chain to the Markov decision process can be done using the concept of adversary (also denoted strategy or scheduler), initially introduced in Section 2.3.1. In the case of an MDP, the operator  $\mathbb{P}_{\triangleright p}$  should be interpreted under the following semantics:

$$s \models \mathbb{P}_{\triangleright p}(\varphi) \text{ if, and only if, } \forall \sigma \in Adv : Pr_s^\sigma(\pi \in Paths^\sigma(s) : \pi \models \varphi) \triangleright p \quad (2.21)$$

that is, the probability, from state  $s$ , that  $\varphi$  is true for an outgoing path satisfies  $\triangleright p$  for all adversaries  $\sigma \in Adv$  (with  $Adv$  constituting the set of all adversaries that the MDP admits).

### 2.3.3 PRISM Language

PRISM is a probabilistic model checking tool developed in cooperation between the Universities of Oxford and Birmingham [13]. It was one of the first tools within stochastic model checking to be released and it is currently maintained by David Parker. In order to use PRISM (and other alternative model checking techniques), it is essential to be able to specify the model. To achieve this, the team behind PRISM developed the PRISM Language, initially described in [11].

A model described in the PRISM Language (typically a `.pm`, `.nm` or `.prism` text file) should contain a keyword that indicates the kind of model to expect. For example, the keyword `dtmc` indicates the model is a DTMC (discrete-time Markov chain) and `mdp` indicates a MDP (Markov decision process).

The PRISM language uses two different basic components to define a model *modules* and *variables*. A module is composed of a set of local variables whose behaviour is described by *commands*. A general command is of the form:

```
[ ] guard -> prob_1: update_1 + ... + prob_n: update_n;
```

where *guard* is a boolean expression over the variables, *prob<sub>i</sub>* is the probability that *update<sub>i</sub>* will be taken (with  $\sum_i \text{prob}_i = 1$ ) and *update<sub>i</sub>* is the the new value of local and/or global variables if the guard is true and a transition is taken to this configuration.

A state of the module is defined as a configuration of its local variables, and a state of the model is a configuration of the states of all modules which compose it and the global variables.

In Listing 2.1, an example of a simple MDP model described in PRISM Language is shown. This example is comprised of two modules, **M1** and **M2**, each with their own local variables, and no global variables are declared. It should be noted that the variables in PRISM Language should either be integers (**int**) or booleans (**bool**) and can be initialised by specifying **init** after the initial declaration. Due to the fact that we are dealing with finite models, the integers must be in a finite range as well.

```
// Example 1 (modified)
// Two process mutual exclusion

mdp

module M1

    x : [0..2] init 0;

    [] x=0 -> 0.8:(x'=0) + 0.2:(x'=1);
    [] x=1 & y!=2 -> (x'=2);
    [] x=2 -> 1:(x'=2);
    [] x=2 -> 1:(x'=0);

endmodule

module M2

    y : [0..2] init 0;

    [] y=0 -> 0.8:(y'=0) + 0.2:(y'=1);
    [] y=1 & x!=2 -> (y'=2);
    [] y=2 -> 1:(y'=2);
    [] y=2 -> 1:(y'=0);

endmodule
```

Listing 2.1: Example of a simple MDP model from [1]

Models in the PRISM Language can also include *formulas*, which are expressions that avoid repetition of code, and *labels*, which are boolean expressions that identify sets of states of interest.

It is also possible to define rewards structures using the following convention:

```
rewards "name"
  guard: state_reward;
  ...
  [action] guard: transition_reward;
  ...
endrewards
```

where the *guards* are similarly defined as before, but the distinction between state and transition rewards is defined by the *action* (which may be empty).

### 2.3.4 Model Checking in DTMCs and MDPs

After the description of the model using a language such as the PRISM Language, it is possible to verify properties on the said model. These properties can be of different types according to the model and the model checking tool chosen.

In this dissertation, the models built will be represented through discrete-time Markov chains and Markov decision processes, so the properties of interest tend to be best expressed in the temporal logics LTL or PCTL. The PRISM model checker is one of the tools which allows a user to import a text file with the probabilistic model, build it and verify LTL and PCTL properties on it [13]. Another tool used for this purpose is the Storm model checker [8]. Each of these tools has its strengths and weaknesses. PRISM is an extremely robust tool, with lots of options in terms of engines to perform the required calculations, and a graphical interface. Storm is a relatively recent tool which builds on top of some of PRISM's features and introduces new ones, such as an optimised Sparse engine (a great advantage when dealing with models with a small state space) and conditional properties in DTMCs [8]. However, it does not provide a graphical interface and is limited in terms of capabilities and I/O. Due to the fact that Storm builds on top of PRISM, it uses the PRISM Language for model input and the properties are specified using the same convention.



In order to illustrate properties, consider the example of Listing 2.1. A PCTL property in this case could be defined as:

```
P>=0.5 [F x = 2]
```

This property can be interpreted as *"with probability greater or equal to 0.5, eventually the model reaches a state where  $x = 2$ "*. Building the model and verifying the property yields the value **false**, which means that, according to this model, there exists at least one adversary under which this property is not satisfied. Both PRISM and Storm allow for quantitative probabilistic properties as well. These properties take different formats in DTMCs and MDPs, as presented below.

```
// DTMC
P=? [F x = 2]

// MDP
Pmin=? [F x = 2]
Pmax=? [F x = 2]
```

Instead of returning a boolean value of satisfaction, these return the actual values of the probabilities of the properties being satisfied. They are, therefore, extremely useful metrics for model comparison. For the case of the DTMC, due to the fact that there is no nondeterminism, the probability of a path is uniquely defined (as seen in Section 2.3.2), and thus can be expressed simply as  $P=?$ . For MDPs, this is not the case. In order to verify the probabilistic operator  $\mathbb{P}_{\triangleright p}$  on the path formula  $\varphi$ , a model checker needs to calculate the following quantities:

$$\begin{aligned} p_{\max,s}(\varphi) &= \sup_{\sigma \in Adv} P_s^\sigma(\varphi) \\ p_{\min,s}(\varphi) &= \inf_{\sigma \in Adv} P_s^\sigma(\varphi) \end{aligned} \tag{2.22}$$

where  $P_s^\sigma(\varphi) = Pr_s^\sigma(\pi \in Paths^\sigma(s) : \pi \models \varphi)$ .

Therefore, PRISM and Storm allow users to access these values by writing properties using **Pmax** and **Pmin** (boolean or quantitative).

Storm allows for conditional probability properties of the type:

$P=? \ [ \ F \ x=2 \ || \ F \ y=2 \ ]$ 

This property can be interpreted as "*what is the probability that the model eventually reaches the state  $x = 2$ , given that it eventually reaches the state  $y = 2$* ". These properties, in DTMCs, correspond simply to the application of Bayes' theorem of conditional probability, which states that:

$$P_s(\varphi_1 \mid \varphi_2) = \frac{P_s(\varphi_1 \wedge \varphi_2)}{P_s(\varphi_2)} \quad (2.23)$$

For  $P_s(\varphi_2) \neq 0$ . It must be emphasised that this is **not** the case for MDPs, where the existence of multiple adversaries complicates the calculations of conditional probabilities.

This does not mean that it is not possible to reason over conditional probabilities in MDPs. For some specific scenarios, it is possible to obtain meaningful bounds which allow comparisons between the ones obtained in DTMCs and MDPs. One of those scenarios is defined in the proposition below, with the proof following it.

**Proposition 2.3.1.** Consider a Markov decision process  $\mathcal{M} = (S, Act, \mathbf{P}, p_{init}, AP, L)$  which admits adversaries  $\sigma \in Adv$ . Let  $\varphi_1, \varphi_2$  be two PCTL path formulas, such that, for any path  $\pi \in Paths(s)$ :  $\pi \models \varphi_1 \Rightarrow \pi \models \varphi_2$ . Assume as well that:

$$\sigma' \in \arg \sup_{\sigma \in Adv} P_s^\sigma(\varphi_1 \mid \varphi_2) \quad (2.24)$$

And that:

$$P_s^{\sigma'}(\varphi_1) = p_{\max,s}(\varphi_1) \quad (2.25)$$

That is, the adversary which maximises the probability of  $\varphi_1 \mid \varphi_2$  also maximises the probability  $\varphi_1$ . In this case it holds that:

$$p_{\max,s}(\varphi_1 \mid \varphi_2) \geq \frac{p_{\max,s}(\varphi_1)}{p_{\max,s}(\varphi_2)} \quad (2.26)$$

with:

$$p_{\max,s}(\varphi_1 \mid \varphi_2) = \sup_{\sigma \in Adv} P_s^\sigma(\varphi_1 \mid \varphi_2) \quad (2.27)$$

*Proof.* Consider  $\sigma_2 \in Adv$  such that:

$$\sigma_2 \in \arg \sup_{\sigma \in Adv} P_s^\sigma(\varphi_2) \quad (2.28)$$

Assume, for the sake of contradiction, that:

$$p_{\max,s}(\varphi_1 \mid \varphi_2) < \frac{p_{\max,s}(\varphi_1)}{p_{\max,s}(\varphi_2)} \quad (2.29)$$

Under adversary  $\sigma'$ , it can be written that, through Bayes' theorem of conditional probability:

$$p_{\max,s}(\varphi_1 \mid \varphi_2) = P_s^{\sigma'}(\varphi_1 \mid \varphi_2) = \frac{P_s^{\sigma'}(\varphi_1 \wedge \varphi_2)}{P_s^{\sigma'}(\varphi_2)} = \frac{P_s^{\sigma'}(\varphi_1)}{P_s^{\sigma'}(\varphi_2)} \quad (2.30)$$

Therefore, it follows that:

$$\frac{P_s^{\sigma'}(\varphi_1)}{P_s^{\sigma'}(\varphi_2)} < \frac{p_{\max,s}(\varphi_1)}{p_{\max,s}(\varphi_2)} \quad (2.31)$$

For this to be true, it must be that  $P_s^{\sigma'}(\varphi_2) > p_{\max,s}(\varphi_2)$ , since  $P_s^{\sigma'}(\varphi_1) = p_{\max,s}(\varphi_1)$ . This constitutes a contradiction, since in that case  $\sigma_2$  would not maximise the probability of  $\varphi_2$  ( $p_{\max,s}(\varphi_2) = P_s^{\sigma_2}(\varphi_2)$ , by definition), a violation of the assumption described in Equation 2.28. Thus, the proposition is proven. ■

In MDPs, however, it is also possible to perform multi-objective verification, whereas multiple properties will be tested simultaneously (both in PRISM and Storm). One example of this would be:

```
multi(P>=0 [F x=2], P>=0.2 [F y=2])
```

The value of the verification of this property corresponds to the boolean conjunction of the individual properties. It is also possible to optimise over a single property while constraining other properties, restricting the possibilities in terms of outcomes. An example of this would be:

```
multi(Pmax=? [F x=2], P>=0.2 [F y=2])
```

Finally, it is also possible to perform multi-objective optimisation over the properties by defining them as (for example):

```
multi(Pmax=? [F x=2], Pmax=? [F y=2])
```

This leads to a situation where compromises between the optimisation of either one of the properties might have to be reached. Pareto curves are the result of the verification of such properties [7].

**Definition 9. Multi-objective Query, Pareto Vector and Pareto Curve (alternatively Set)**

A *multi-objective query* (MQ)  $\phi$  of  $n$  objectives is a positive boolean combination of  $n$  predicates of the form  $r_i \geq v_i$ , where  $r_i$  is a reward function,  $v_i \in \mathbb{Q}$  is a bound. The notation  $\phi[\mathbf{x}]$ ,  $\mathbf{x} \in \mathbb{R}^n$  is used to denote  $\phi$  in which each  $r_i \geq v_i$  is replaced by  $r_i \geq x_i$ .

Consider a multi-objective query MQ  $\phi$  of  $n$  objectives. The vector  $\mathbf{x} \in \mathbb{R}^n$  is a *Pareto vector* of  $\phi$  if, and only if:

1.  $\phi[\mathbf{x} - \varepsilon]$  is achievable for all  $\varepsilon > 0$ , and
2.  $\phi[\mathbf{x} + \varepsilon]$  is not achievable for any  $\varepsilon > 0$

A *Pareto curve* of  $\phi$  is the set of all Pareto vectors that  $\phi$  admits.

■

Thus, the result of the verification of a multi-objective optimisation property corresponds to a Pareto curve (this curve might contain a single vector). Both PRISM

and Storm allow for the verification of properties containing up to 2 objectives each. As a result, 2D plots are the best way to represent these solutions. In Figure 2.4, the result of the verification of the property `multi(Pmax=? [F G x=2], Pmax=? [F G y=2])` is shown. As it can be observed, either property can be satisfied with certainty, but not both simultaneously (mutual exclusion). The green area under the curve corresponds to the achievable set.

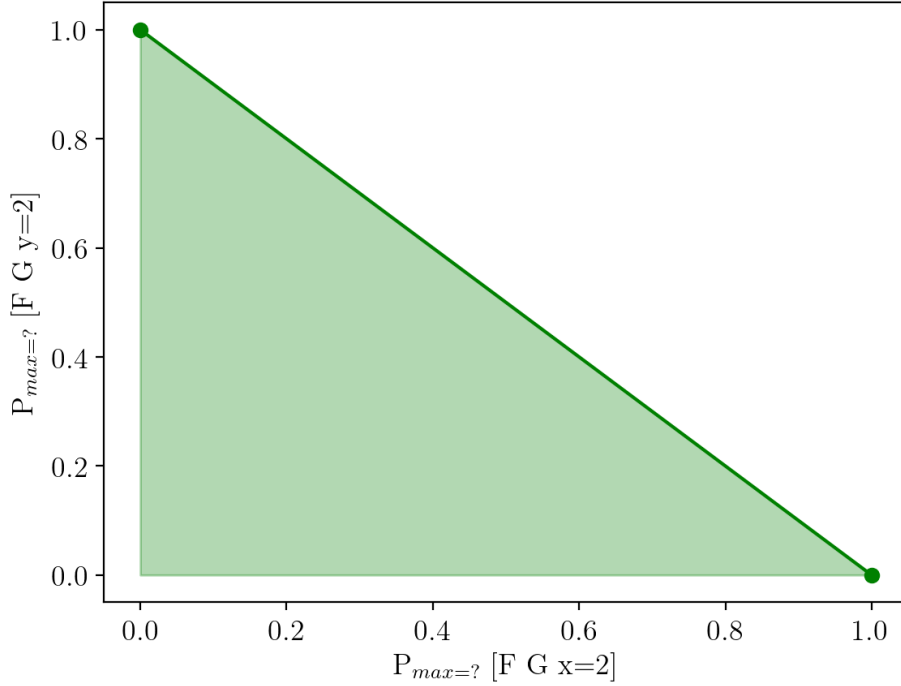


Figure 2.4: Example of a Pareto curve represented as a 2D plot.

Each of the values resulting of the verification of quantitative properties in MDPs (whether they be a single value from a single optimisation, or a Pareto vector) is the result of an adversary which solves the nondeterminism and converts the MDP into a DTMC [4]. The process of obtaining the adversary which satisfies certain properties is denoted by **strategy synthesis** or **adversary generation**, and it is formally defined below.

**Definition 10. Strategy Synthesis (alternatively Adversary Generation)**

Consider a Markov decision process  $\mathcal{M} = (S, Act, \mathbf{P}, p_{init}, AP, L)$ . The process of maximal (similarly minimal) *strategy synthesis* of the path formula  $\varphi$  is defined as a process that yields an adversary  $\sigma \in Adv$  such that:

$$\sigma \in \arg \sup_{\sigma' \in Adv} P_s^{\sigma'}(\varphi) \quad (2.32)$$

(respectively  $\arg \inf$  for minimal), where  $Adv$  is the set of all adversaries that  $\mathcal{M}$  admits.

■

## Chapter 3

# Human Driver Modelling

In this chapter, the human driver model that Salvucci presents in [17] is implemented and the metrics used to verify the model are established. Initially, the model is implemented in *Matlab* using the continuous driver model. To be able to verify it using the probabilistic model checking techniques presented in Chapter 2, the model undergoes an abstraction process, out of which a discrete-time Markov chain is obtained using several different approaches and assumptions. Metrics are then established to be able to evaluate whether the model meets the requirements and how well the human performs in a given situation. Finally, a simulator is presented in Section 3.4 in order to allow visualisation of paths in the model.

### 3.1 Continuous Driver Model using ACT-R

In order to obtain a continuous driver model, we interpret Salvucci’s integrated driver model as presented in [17]. The scenario considered is the one Salvucci envisioned: a multilane highway with moderate (or low) traffic [17]. The model uses the three modules presented in Figure 2.2, that is, monitoring, decision making and control. All these modules rely on the constant update of the environment, as some are based on visual perception or low-level perception cues. While the human model is presented clearly in [17], the dynamics of the car (which is part of the environment) are left to the reader, as it is outside the scope of the paper. In that sense, some assumptions had to be made about the environment.

The main initial assumption is that the driver is perfectly aware of its surroundings,

and thus it is able to obtain the positions of other vehicles (within a certain distance) and the near and far points perfectly (this assumption is challenged later in the abstraction process). It is also assumed that the vehicle the driver is using is directly controlled by the updating of any pertinent variables within the control module of the architecture (i.e. there is no intermediate system or separation). The environment is assumed to change at the same rate as a cycle of the ACT-R model.

The overall view of the system is presented in Figure 3.1. The information flow is marked using the dotted arrows, while the sequential flow of the program is marked in the filled, lighter gray ones. Both the monitoring and the decision making make use of the perception which corresponds to querying the environment for the information (e.g. near and far point or another vehicle's position). The control both queries and sends updated information (e.g. vehicle position) to the environment. Modules are updated sequentially, following the order: control, monitoring, decision making and environment (e.g. position, velocity and acceleration of other vehicles).

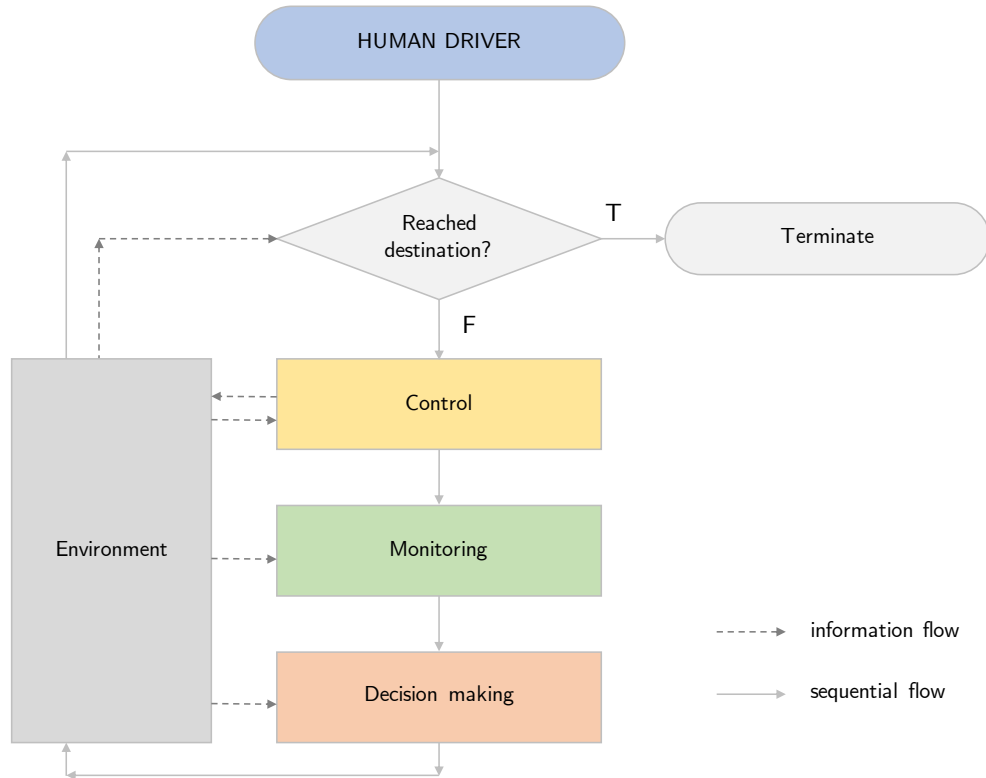


Figure 3.1: Continuous Driver Model overview.

The monitoring module is implemented according to the flowchart presented in Figure 3.2, from the description given in [17]. The threshold for monitoring is set at 0.2,



as this is the value suggested in Salvucci in [17]. The output of this process corresponds to altering the global variable  $a$  corresponding to the declarative memory cell composed of  $k$  chunks. No specific value for  $k$  is given by Salvucci in [17], but it is taken to be 8 from [14]. The function `GET_DISTANCE` is not described in depth due to the fact that it is trivial under the assumption stated in the previous paragraph.

The overall flowchart of the decision making module is shown in Figure 3.3. The high level part is similar to the flowchart presented in Figure 2.3, with small differences related to the loading of variables and information from the different memories available. It is worth noticing that the control makes use of the function `LOOK_VEHICLE` initially presented in the monitoring module to verify the presence of a vehicle in a relative position of a lane. The function `SET_LANE_FOLLOWING` is also omitted due to the trivial nature of this function under the assumptions made.

Finally, a flowchart for the control module is presented in Figure 3.4. This module is responsible for the calculation of  $\Delta\varphi(t)$  and  $\Delta\psi(t)$  and the update of the position. Given these two values, the position is updated following discrete laws of motion applied to rigid objects:

$$\begin{aligned} \mathbf{v}(t) &= \mathbf{v}(t-1) + \mathbf{a}(t)\Delta t \\ \mathbf{x}(t) &= \mathbf{x}(t-1) + \mathbf{v}(t-1)\Delta t + \frac{1}{2}\mathbf{a}(t)\Delta t^2 \\ t &= t + \Delta t \end{aligned} \tag{3.1}$$

where  $\mathbf{x}(t) = (x(t), y(t))$ ,  $\mathbf{v}(t) = (v_x(t), v_y(t))$ , and  $\mathbf{a}(t) = (a_x(t), a_y(t))$ , with:

$$\begin{aligned} a_x(t) &= \Delta\psi(t) \\ a_y(t) &= \sin(\Delta\varphi(t)) \end{aligned} \tag{3.2}$$

In this case, as Salvucci mentions in [17],  $\Delta t = 0.5s$ . It is also assumed that the vehicle does not speed, and therefore,  $v_x \in [15, 34] \text{ m/s}$  (between  $50\text{km/h}$  and  $120\text{km/h}$  - typical limits in highway speeds).

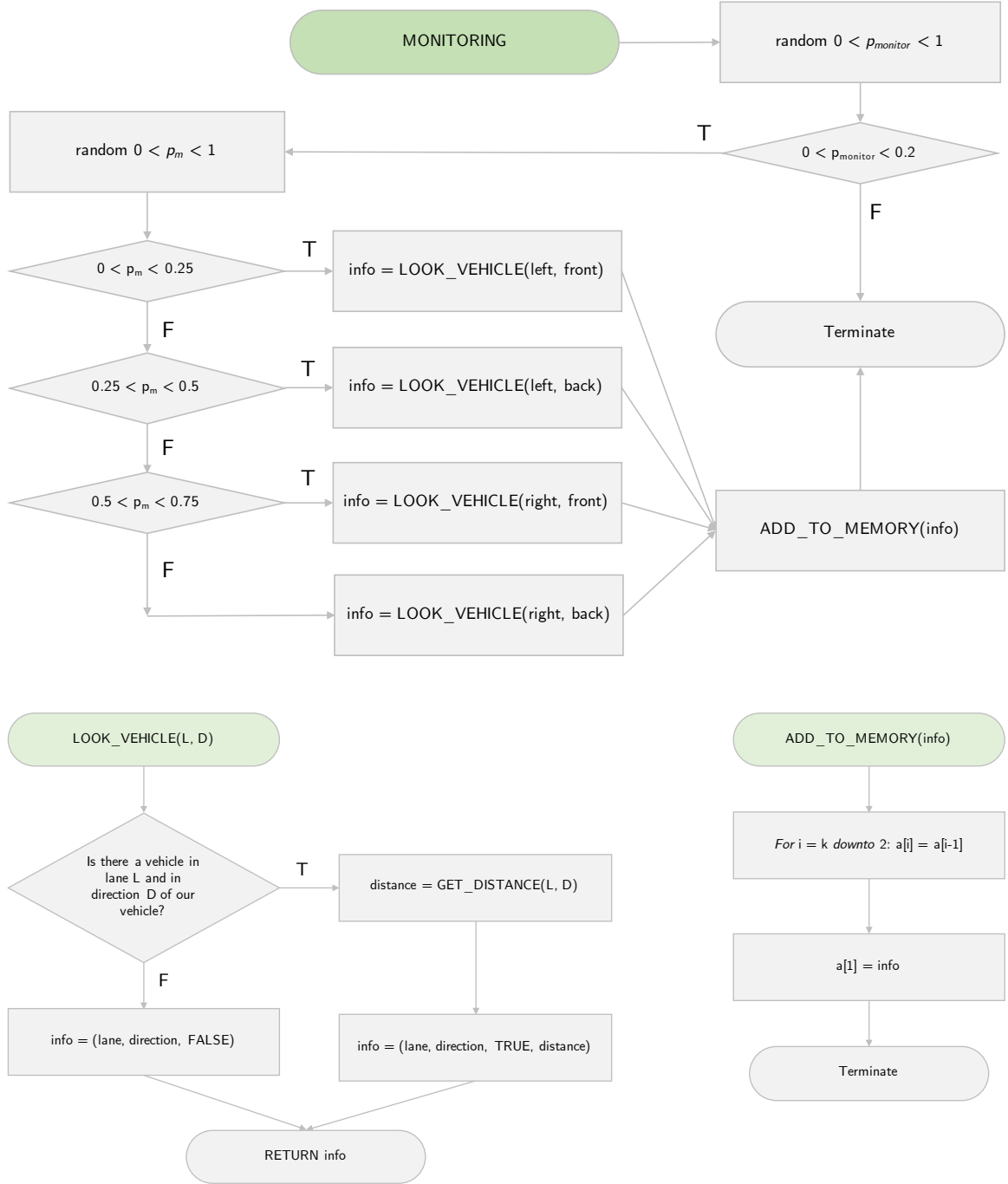


Figure 3.2: Flowchart of the Monitoring module.

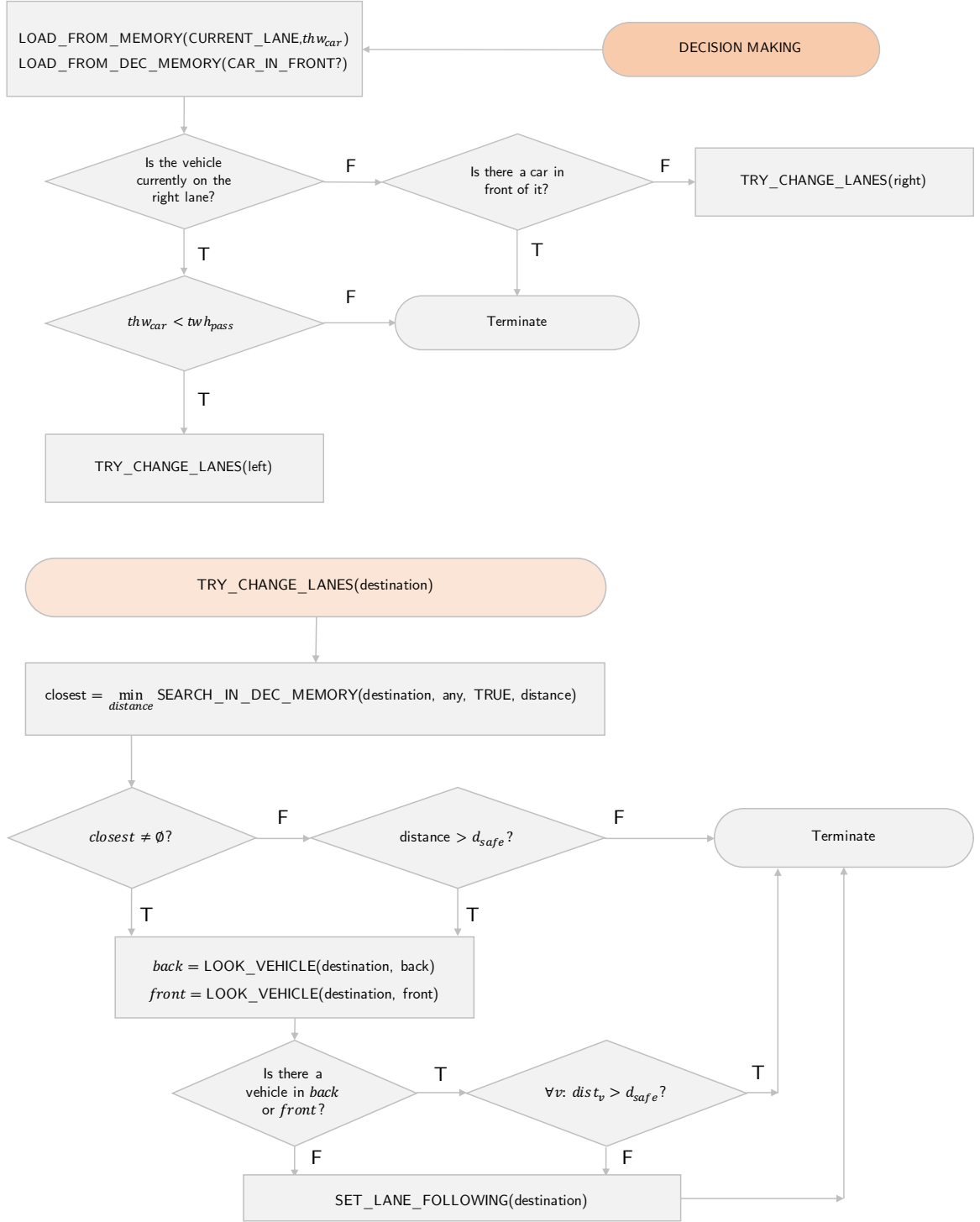


Figure 3.3: Flowchart of the Decision Making module.

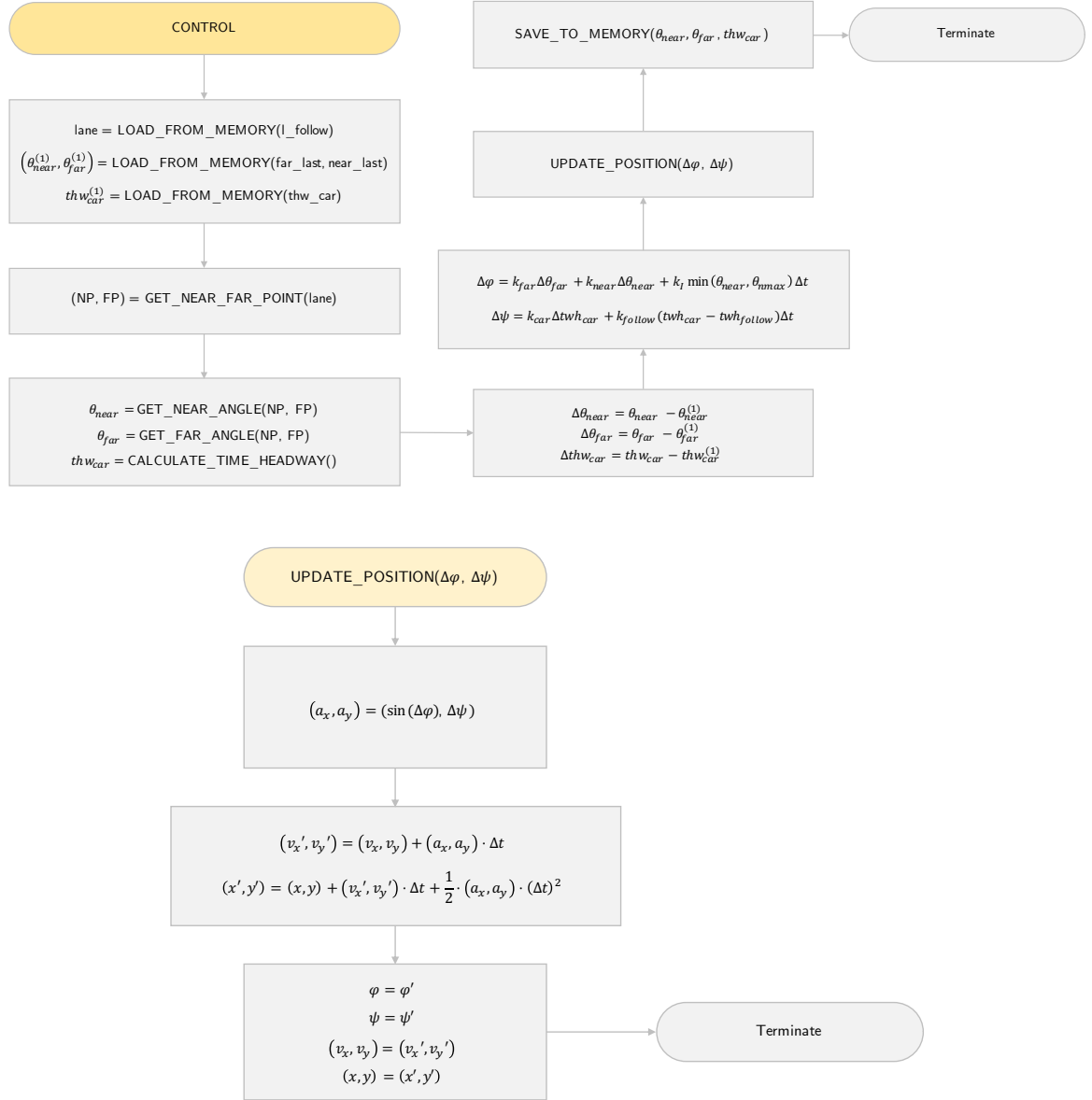


Figure 3.4: Flowchart of the Control module.

## 3.2 Model Abstraction

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Pellentesque accumsan magna eu metus congue, vitae tristique enim commodo. Aliquam consectetur porttitor nisl, eget pulvinar quam venenatis eget. Aliquam erat volutpat. Nullam at ipsum at felis tincidunt consequat non sed mi. Morbi placerat iaculis dapibus. Aliquam non risus ac urna interdum pretium. Donec maximus, massa vitae suscipit congue, ligula purus sagittis magna, vitae interdum enim dui sed felis. Curabitur mollis eleifend diam a varius. Vivamus eu porttitor lorem. Mauris sagittis ultrices ligula, a euismod mi efficitur a. Aenean molestie sapien a nibh suscipit, nec venenatis urna varius. Sed eu ex id lacus tincidunt ultricies. Aenean quam massa, dignissim ac suscipit posuere, varius sit amet turpis. Nunc semper turpis ullamcorper elit venenatis, in molestie nulla fermentum.

### 3.2.1 Non-Probabilistic Control Module

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Pellentesque accumsan magna eu metus congue, vitae tristique enim commodo. Aliquam consectetur porttitor nisl, eget pulvinar quam venenatis eget. Aliquam erat volutpat. Nullam at ipsum at felis tincidunt consequat non sed mi. Morbi placerat iaculis dapibus. Aliquam non risus ac urna interdum pretium. Donec maximus, massa vitae suscipit congue, ligula purus sagittis magna, vitae interdum enim dui sed felis. Curabitur mollis eleifend diam a varius. Vivamus eu porttitor lorem. Mauris sagittis ultrices ligula, a euismod mi efficitur a. Aenean molestie sapien a nibh suscipit, nec venenatis urna varius. Sed eu ex id lacus tincidunt ultricies. Aenean quam massa, dignissim ac suscipit posuere, varius sit amet turpis. Nunc semper turpis ullamcorper elit venenatis, in molestie nulla fermentum.

### 3.2.2 Decision Making and Monitoring Module

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Pellentesque accumsan magna eu metus congue, vitae tristique enim commodo. Aliquam consectetur porttitor nisl, eget pulvinar quam venenatis eget. Aliquam erat volutpat. Nullam at ipsum at felis tincidunt consequat non sed mi. Morbi placerat iaculis dapibus. Aliquam non

risus ac urna interdum pretium. Donec maximus, massa vitae suscipit congue, ligula purus sagittis magna, vitae interdum enim dui sed felis. Curabitur mollis eleifend diam a varius. Vivamus eu porttitor lorem. Mauris sagittis ultrices ligula, a euismod mi efficitur a. Aenean molestie sapien a nibh suscipit, nec venenatis urna varius. Sed eu ex id lacus tincidunt ultricies. Aenean quam massa, dignissim ac suscipit posuere, varius sit amet turpis. Nunc semper turpis ullamcorper elit venenatis, in molestie nulla fermentum.

### 3.2.3 Probabilistic Control Module

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Pellentesque accumsan magna eu metus congue, vitae tristique enim commodo. Aliquam consectetur porttitor nisl, eget pulvinar quam venenatis eget. Aliquam erat volutpat. Nullam at ipsum at felis tincidunt consequat non sed mi. Morbi placerat iaculis dapibus. Aliquam non risus ac urna interdum pretium. Donec maximus, massa vitae suscipit congue, ligula purus sagittis magna, vitae interdum enim dui sed felis. Curabitur mollis eleifend diam a varius. Vivamus eu porttitor lorem. Mauris sagittis ultrices ligula, a euismod mi efficitur a. Aenean molestie sapien a nibh suscipit, nec venenatis urna varius. Sed eu ex id lacus tincidunt ultricies. Aenean quam massa, dignissim ac suscipit posuere, varius sit amet turpis. Nunc semper turpis ullamcorper elit venenatis, in molestie nulla fermentum.

## 3.3 Model Evaluation Metrics

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Pellentesque accumsan magna eu metus congue, vitae tristique enim commodo. Aliquam consectetur porttitor nisl, eget pulvinar quam venenatis eget. Aliquam erat volutpat. Nullam at ipsum at felis tincidunt consequat non sed mi. Morbi placerat iaculis dapibus. Aliquam non risus ac urna interdum pretium. Donec maximus, massa vitae suscipit congue, ligula purus sagittis magna, vitae interdum enim dui sed felis. Curabitur mollis eleifend diam a varius. Vivamus eu porttitor lorem. Mauris sagittis ultrices ligula, a euismod mi efficitur a. Aenean molestie sapien a nibh suscipit, nec venenatis urna varius. Sed eu ex id lacus tincidunt ultricies. Aenean quam massa, dignissim ac suscipit posuere,

varius sit amet turpis. Nunc semper turpis ullamcorper elit venenatis, in molestie nulla fermentum.

### 3.3.1 Completeness Property

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Pellentesque accumsan magna eu metus congue, vitae tristique enim commodo. Aliquam consectetur porttitor nisl, eget pulvinar quam venenatis eget. Aliquam erat volutpat. Nullam at ipsum at felis tincidunt consequat non sed mi. Morbi placerat iaculis dapibus. Aliquam non risus ac urna interdum pretium. Donec maximus, massa vitae suscipit congue, ligula purus sagittis magna, vitae interdum enim dui sed felis. Curabitur mollis eleifend diam a varius. Vivamus eu porttitor lorem. Mauris sagittis ultrices ligula, a euismod mi efficitur a. Aenean molestie sapien a nibh suscipit, nec venenatis urna varius. Sed eu ex id lacus tincidunt ultricies. Aenean quam massa, dignissim ac suscipit posuere, varius sit amet turpis. Nunc semper turpis ullamcorper elit venenatis, in molestie nulla fermentum.

### 3.3.2 Safety Property

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Pellentesque accumsan magna eu metus congue, vitae tristique enim commodo. Aliquam consectetur porttitor nisl, eget pulvinar quam venenatis eget. Aliquam erat volutpat. Nullam at ipsum at felis tincidunt consequat non sed mi. Morbi placerat iaculis dapibus. Aliquam non risus ac urna interdum pretium. Donec maximus, massa vitae suscipit congue, ligula purus sagittis magna, vitae interdum enim dui sed felis. Curabitur mollis eleifend diam a varius. Vivamus eu porttitor lorem. Mauris sagittis ultrices ligula, a euismod mi efficitur a. Aenean molestie sapien a nibh suscipit, nec venenatis urna varius. Sed eu ex id lacus tincidunt ultricies. Aenean quam massa, dignissim ac suscipit posuere, varius sit amet turpis. Nunc semper turpis ullamcorper elit venenatis, in molestie nulla fermentum.

### 3.3.3 Liveness Properties

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Pellentesque accumsan magna eu metus congue, vitae tristique enim commodo. Aliquam consectetur porttitor nisl, eget pulvinar quam venenatis eget. Aliquam erat volutpat. Nullam at ipsum at felis tincidunt consequat non sed mi. Morbi placerat iaculis dapibus. Aliquam non risus ac urna interdum pretium. Donec maximus, massa vitae suscipit congue, ligula purus sagittis magna, vitae interdum enim dui sed felis. Curabitur mollis eleifend diam a varius. Vivamus eu porttitor lorem. Mauris sagittis ultrices ligula, a euismod mi efficitur a. Aenean molestie sapien a nibh suscipit, nec venenatis urna varius. Sed eu ex id lacus tincidunt ultricies. Aenean quam massa, dignissim ac suscipit posuere, varius sit amet turpis. Nunc semper turpis ullamcorper elit venenatis, in molestie nulla fermentum.

## 3.4 Simulation of Paths in the Model

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Pellentesque accumsan magna eu metus congue, vitae tristique enim commodo. Aliquam consectetur porttitor nisl, eget pulvinar quam venenatis eget. Aliquam erat volutpat. Nullam at ipsum at felis tincidunt consequat non sed mi. Morbi placerat iaculis dapibus. Aliquam non risus ac urna interdum pretium. Donec maximus, massa vitae suscipit congue, ligula purus sagittis magna, vitae interdum enim dui sed felis. Curabitur mollis eleifend diam a varius. Vivamus eu porttitor lorem. Mauris sagittis ultrices ligula, a euismod mi efficitur a. Aenean molestie sapien a nibh suscipit, nec venenatis urna varius. Sed eu ex id lacus tincidunt ultricies. Aenean quam massa, dignissim ac suscipit posuere, varius sit amet turpis. Nunc semper turpis ullamcorper elit venenatis, in molestie nulla fermentum.



## Chapter 4

# Advanced Driver Assistance Systems

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Pellentesque accumsan magna eu metus congue, vitae tristique enim commodo. Aliquam consectetur porttitor nisl, eget pulvinar quam venenatis eget. Aliquam erat volutpat. Nullam at ipsum at felis tincidunt consequat non sed mi. Morbi placerat iaculis dapibus. Aliquam non risus ac urna interdum pretium. Donec maximus, massa vitae suscipit congue, ligula purus sagittis magna, vitae interdum enim dui sed felis. Curabitur mollis eleifend diam a varius. Vivamus eu porttitor lorem. Mauris sagittis ultrices ligula, a euismod mi efficitur a. Aenean molestie sapien a nibh suscipit, nec venenatis urna varius. Sed eu ex id lacus tincidunt ultricies. Aenean quam massa, dignissim ac suscipit posuere, varius sit amet turpis. Nunc semper turpis ullamcorper elit venenatis, in molestie nulla fermentum.

### 4.1 Driver Assistance System Design

#### 4.1.1 Decision Making based ADAS

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Pellentesque accumsan magna eu metus congue, vitae tristique enim commodo. Aliquam consectetur porttitor nisl, eget pulvinar quam venenatis eget. Aliquam erat volutpat. Nullam at ipsum at felis tincidunt consequat non sed mi. Morbi placerat iaculis dapibus. Aliquam non risus ac urna interdum pretium. Donec maximus, massa vitae suscipit congue, ligula

purus sagittis magna, vitae interdum enim dui sed felis. Curabitur mollis eleifend diam a varius. Vivamus eu porttitor lorem. Mauris sagittis ultrices ligula, a euismod mi efficitur a. Aenean molestie sapien a nibh suscipit, nec venenatis urna varius. Sed eu ex id lacus tincidunt ultricies. Aenean quam massa, dignissim ac suscipit posuere, varius sit amet turpis. Nunc semper turpis ullamcorper elit venenatis, in molestie nulla fermentum.

#### **4.1.1.1 Decision Making with Fully Compliant Drivers**

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Pellentesque accumsan magna eu metus congue, vitae tristique enim commodo. Aliquam consectetur porttitor nisl, eget pulvinar quam venenatis eget. Aliquam erat volutpat. Nullam at ipsum at felis tincidunt consequat non sed mi. Morbi placerat iaculis dapibus. Aliquam non risus ac urna interdum pretium. Donec maximus, massa vitae suscipit congue, ligula purus sagittis magna, vitae interdum enim dui sed felis. Curabitur mollis eleifend diam a varius. Vivamus eu porttitor lorem. Mauris sagittis ultrices ligula, a euismod mi efficitur a. Aenean molestie sapien a nibh suscipit, nec venenatis urna varius. Sed eu ex id lacus tincidunt ultricies. Aenean quam massa, dignissim ac suscipit posuere, varius sit amet turpis. Nunc semper turpis ullamcorper elit venenatis, in molestie nulla fermentum.

#### **4.1.1.2 Decision Making with Partially Compliant Drivers**

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Pellentesque accumsan magna eu metus congue, vitae tristique enim commodo. Aliquam consectetur porttitor nisl, eget pulvinar quam venenatis eget. Aliquam erat volutpat. Nullam at ipsum at felis tincidunt consequat non sed mi. Morbi placerat iaculis dapibus. Aliquam non risus ac urna interdum pretium. Donec maximus, massa vitae suscipit congue, ligula purus sagittis magna, vitae interdum enim dui sed felis. Curabitur mollis eleifend diam a varius. Vivamus eu porttitor lorem. Mauris sagittis ultrices ligula, a euismod mi efficitur a. Aenean molestie sapien a nibh suscipit, nec venenatis urna varius. Sed eu ex id lacus tincidunt ultricies. Aenean quam massa, dignissim ac suscipit posuere, varius sit amet turpis. Nunc semper turpis ullamcorper elit venenatis, in molestie nulla fermentum.

### 4.1.2 Control based ADAS

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Pellentesque accumsan magna eu metus congue, vitae tristique enim commodo. Aliquam consectetur porttitor nisl, eget pulvinar quam venenatis eget. Aliquam erat volutpat. Nullam at ipsum at felis tincidunt consequat non sed mi. Morbi placerat iaculis dapibus. Aliquam non risus ac urna interdum pretium. Donec maximus, massa vitae suscipit congue, ligula purus sagittis magna, vitae interdum enim dui sed felis. Curabitur mollis eleifend diam a varius. Vivamus eu porttitor lorem. Mauris sagittis ultrices ligula, a euismod mi efficitur a. Aenean molestie sapien a nibh suscipit, nec venenatis urna varius. Sed eu ex id lacus tincidunt ultricies. Aenean quam massa, dignissim ac suscipit posuere, varius sit amet turpis. Nunc semper turpis ullamcorper elit venenatis, in molestie nulla fermentum.

#### 4.1.2.1 Active Linear Acceleration Control

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Pellentesque accumsan magna eu metus congue, vitae tristique enim commodo. Aliquam consectetur porttitor nisl, eget pulvinar quam venenatis eget. Aliquam erat volutpat. Nullam at ipsum at felis tincidunt consequat non sed mi. Morbi placerat iaculis dapibus. Aliquam non risus ac urna interdum pretium. Donec maximus, massa vitae suscipit congue, ligula purus sagittis magna, vitae interdum enim dui sed felis. Curabitur mollis eleifend diam a varius. Vivamus eu porttitor lorem. Mauris sagittis ultrices ligula, a euismod mi efficitur a. Aenean molestie sapien a nibh suscipit, nec venenatis urna varius. Sed eu ex id lacus tincidunt ultricies. Aenean quam massa, dignissim ac suscipit posuere, varius sit amet turpis. Nunc semper turpis ullamcorper elit venenatis, in molestie nulla fermentum.

#### 4.1.2.2 Active Steering Control

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Pellentesque accumsan magna eu metus congue, vitae tristique enim commodo. Aliquam consectetur porttitor nisl, eget pulvinar quam venenatis eget. Aliquam erat volutpat. Nullam at ipsum at felis tincidunt consequat non sed mi. Morbi placerat iaculis dapibus. Aliquam non risus ac urna interdum pretium. Donec maximus, massa vitae suscipit congue, ligula

purus sagittis magna, vitae interdum enim dui sed felis. Curabitur mollis eleifend diam a varius. Vivamus eu porttitor lorem. Mauris sagittis ultrices ligula, a euismod mi efficitur a. Aenean molestie sapien a nibh suscipit, nec venenatis urna varius. Sed eu ex id lacus tincidunt ultricies. Aenean quam massa, dignissim ac suscipit posuere, varius sit amet turpis. Nunc semper turpis ullamcorper elit venenatis, in molestie nulla fermentum.

## 4.2 Evaluation

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Pellentesque accumsan magna eu metus congue, vitae tristique enim commodo. Aliquam consectetur porttitor nisl, eget pulvinar quam venenatis eget. Aliquam erat volutpat. Nullam at ipsum at felis tincidunt consequat non sed mi. Morbi placerat iaculis dapibus. Aliquam non risus ac urna interdum pretium. Donec maximus, massa vitae suscipit congue, ligula purus sagittis magna, vitae interdum enim dui sed felis. Curabitur mollis eleifend diam a varius. Vivamus eu porttitor lorem. Mauris sagittis ultrices ligula, a euismod mi efficitur a. Aenean molestie sapien a nibh suscipit, nec venenatis urna varius. Sed eu ex id lacus tincidunt ultricies. Aenean quam massa, dignissim ac suscipit posuere, varius sit amet turpis. Nunc semper turpis ullamcorper elit venenatis, in molestie nulla fermentum.

### 4.2.1 Multi-Objective Metrics

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Pellentesque accumsan magna eu metus congue, vitae tristique enim commodo. Aliquam consectetur porttitor nisl, eget pulvinar quam venenatis eget. Aliquam erat volutpat. Nullam at ipsum at felis tincidunt consequat non sed mi. Morbi placerat iaculis dapibus. Aliquam non risus ac urna interdum pretium. Donec maximus, massa vitae suscipit congue, ligula purus sagittis magna, vitae interdum enim dui sed felis. Curabitur mollis eleifend diam a varius. Vivamus eu porttitor lorem. Mauris sagittis ultrices ligula, a euismod mi efficitur a. Aenean molestie sapien a nibh suscipit, nec venenatis urna varius. Sed eu ex id lacus tincidunt ultricies. Aenean quam massa, dignissim ac suscipit posuere, varius sit amet turpis. Nunc semper turpis ullamcorper elit venenatis, in molestie nulla fermentum.

## 4.2.2 Test Cases

### 4.2.2.1 $v = 21 \text{ m/s}$ , $v_1 = 19 \text{ m/s}$ , $x_{1,0} = 70 \text{ m}$

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Pellentesque accumsan magna eu metus congue, vitae tristique enim commodo. Aliquam consectetur porttitor nisl, eget pulvinar quam venenatis eget. Aliquam erat volutpat. Nullam at ipsum at felis tincidunt consequat non sed mi. Morbi placerat iaculis dapibus. Aliquam non risus ac urna interdum pretium. Donec maximus, massa vitae suscipit congue, ligula purus sagittis magna, vitae interdum enim dui sed felis. Curabitur mollis eleifend diam a varius. Vivamus eu porttitor lorem. Mauris sagittis ultrices ligula, a euismod mi efficitur a. Aenean molestie sapien a nibh suscipit, nec venenatis urna varius. Sed eu ex id lacus tincidunt ultricies. Aenean quam massa, dignissim ac suscipit posuere, varius sit amet turpis. Nunc semper turpis ullamcorper elit venenatis, in molestie nulla fermentum.

### 4.2.2.2 $v = 30 \text{ m/s}$ , $v_1 = 22 \text{ m/s}$ , $x_{1,0} = 50 \text{ m}$

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Pellentesque accumsan magna eu metus congue, vitae tristique enim commodo. Aliquam consectetur porttitor nisl, eget pulvinar quam venenatis eget. Aliquam erat volutpat. Nullam at ipsum at felis tincidunt consequat non sed mi. Morbi placerat iaculis dapibus. Aliquam non risus ac urna interdum pretium. Donec maximus, massa vitae suscipit congue, ligula purus sagittis magna, vitae interdum enim dui sed felis. Curabitur mollis eleifend diam a varius. Vivamus eu porttitor lorem. Mauris sagittis ultrices ligula, a euismod mi efficitur a. Aenean molestie sapien a nibh suscipit, nec venenatis urna varius. Sed eu ex id lacus tincidunt ultricies. Aenean quam massa, dignissim ac suscipit posuere, varius sit amet turpis. Nunc semper turpis ullamcorper elit venenatis, in molestie nulla fermentum.

## 4.2.3 Overall Comparison

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Pellentesque accumsan magna eu metus congue, vitae tristique enim commodo. Aliquam consectetur porttitor nisl, eget pulvinar quam venenatis eget. Aliquam erat volutpat. Nullam at ipsum

at felis tincidunt consequat non sed mi. Morbi placerat iaculis dapibus. Aliquam non risus ac urna interdum pretium. Donec maximus, massa vitae suscipit congue, ligula purus sagittis magna, vitae interdum enim dui sed felis. Curabitur mollis eleifend diam a varius. Vivamus eu porttitor lorem. Mauris sagittis ultrices ligula, a euismod mi efficitur a. Aenean molestie sapien a nibh suscipit, nec venenatis urna varius. Sed eu ex id lacus tincidunt ultricies. Aenean quam massa, dignissim ac suscipit posuere, varius sit amet turpis. Nunc semper turpis ullamcorper elit venenatis, in molestie nulla fermentum.

# Chapter 5

## Experimental Results

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Pellentesque accumsan magna eu metus congue, vitae tristique enim commodo. Aliquam consectetur porttitor nisl, eget pulvinar quam venenatis eget. Aliquam erat volutpat. Nullam at ipsum at felis tincidunt consequat non sed mi. Morbi placerat iaculis dapibus. Aliquam non risus ac urna interdum pretium. Donec maximus, massa vitae suscipit congue, ligula purus sagittis magna, vitae interdum enim dui sed felis. Curabitur mollis eleifend diam a varius. Vivamus eu porttitor lorem. Mauris sagittis ultrices ligula, a euismod mi efficitur a. Aenean molestie sapien a nibh suscipit, nec venenatis urna varius. Sed eu ex id lacus tincidunt ultricies. Aenean quam massa, dignissim ac suscipit posuere, varius sit amet turpis. Nunc semper turpis ullamcorper elit venenatis, in molestie nulla fermentum.

### 5.1 Safety Metrics

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Pellentesque accumsan magna eu metus congue, vitae tristique enim commodo. Aliquam consectetur porttitor nisl, eget pulvinar quam venenatis eget. Aliquam erat volutpat. Nullam at ipsum at felis tincidunt consequat non sed mi. Morbi placerat iaculis dapibus. Aliquam non risus ac urna interdum pretium. Donec maximus, massa vitae suscipit congue, ligula purus sagittis magna, vitae interdum enim dui sed felis. Curabitur mollis eleifend diam a varius. Vivamus eu porttitor lorem. Mauris sagittis ultrices ligula, a euismod mi efficitur a. Aenean molestie sapien a nibh suscipit, nec venenatis urna varius. Sed

eu ex id lacus tincidunt ultricies. Aenean quam massa, dignissim ac suscipit posuere, varius sit amet turpis. Nunc semper turpis ullamcorper elit venenatis, in molestie nulla fermentum.

## 5.2 Liveness Metrics

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Pellentesque accumsan magna eu metus congue, vitae tristique enim commodo. Aliquam consectetur porttitor nisl, eget pulvinar quam venenatis eget. Aliquam erat volutpat. Nullam at ipsum at felis tincidunt consequat non sed mi. Morbi placerat iaculis dapibus. Aliquam non risus ac urna interdum pretium. Donec maximus, massa vitae suscipit congue, ligula purus sagittis magna, vitae interdum enim dui sed felis. Curabitur mollis eleifend diam a varius. Vivamus eu porttitor lorem. Mauris sagittis ultrices ligula, a euismod mi efficitur a. Aenean molestie sapien a nibh suscipit, nec venenatis urna varius. Sed eu ex id lacus tincidunt ultricies. Aenean quam massa, dignissim ac suscipit posuere, varius sit amet turpis. Nunc semper turpis ullamcorper elit venenatis, in molestie nulla fermentum.

## 5.3 Discussion

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Pellentesque accumsan magna eu metus congue, vitae tristique enim commodo. Aliquam consectetur porttitor nisl, eget pulvinar quam venenatis eget. Aliquam erat volutpat. Nullam at ipsum at felis tincidunt consequat non sed mi. Morbi placerat iaculis dapibus. Aliquam non risus ac urna interdum pretium. Donec maximus, massa vitae suscipit congue, ligula purus sagittis magna, vitae interdum enim dui sed felis. Curabitur mollis eleifend diam a varius. Vivamus eu porttitor lorem. Mauris sagittis ultrices ligula, a euismod mi efficitur a. Aenean molestie sapien a nibh suscipit, nec venenatis urna varius. Sed eu ex id lacus tincidunt ultricies. Aenean quam massa, dignissim ac suscipit posuere, varius sit amet turpis. Nunc semper turpis ullamcorper elit venenatis, in molestie nulla fermentum.



## 5.4 Additional Experiments

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Pellentesque accumsan magna eu metus congue, vitae tristique enim commodo. Aliquam consectetur porttitor nisl, eget pulvinar quam venenatis eget. Aliquam erat volutpat. Nullam at ipsum at felis tincidunt consequat non sed mi. Morbi placerat iaculis dapibus. Aliquam non risus ac urna interdum pretium. Donec maximus, massa vitae suscipit congue, ligula purus sagittis magna, vitae interdum enim dui sed felis. Curabitur mollis eleifend diam a varius. Vivamus eu porttitor lorem. Mauris sagittis ultrices ligula, a euismod mi efficitur a. Aenean molestie sapien a nibh suscipit, nec venenatis urna varius. Sed eu ex id lacus tincidunt ultricies. Aenean quam massa, dignissim ac suscipit posuere, varius sit amet turpis. Nunc semper turpis ullamcorper elit venenatis, in molestie nulla fermentum.

## Chapter 6

# Conclusions and Future Work

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Pellentesque accumsan magna eu metus congue, vitae tristique enim commodo. Aliquam consectetur porttitor nisl, eget pulvinar quam venenatis eget. Aliquam erat volutpat. Nullam at ipsum at felis tincidunt consequat non sed mi. Morbi placerat iaculis dapibus. Aliquam non risus ac urna interdum pretium. Donec maximus, massa vitae suscipit congue, ligula purus sagittis magna, vitae interdum enim dui sed felis. Curabitur mollis eleifend diam a varius. Vivamus eu porttitor lorem. Mauris sagittis ultrices ligula, a euismod mi efficitur a. Aenean molestie sapien a nibh suscipit, nec venenatis urna varius. Sed eu ex id lacus tincidunt ultricies. Aenean quam massa, dignissim ac suscipit posuere, varius sit amet turpis. Nunc semper turpis ullamcorper elit venenatis, in molestie nulla fermentum.

# Appendix A

## Code for Human Driver Modelling

### A.1 ACT-R Implementation (Matlab)

#### A.1.1 Main Loop

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Pellentesque accumsan magna eu metus congue, vitae tristique enim commodo. Aliquam consectetur porttitor nisl, eget pulvinar quam venenatis eget. Aliquam erat volutpat. Nullam at ipsum at felis tincidunt consequat non sed mi. Morbi placerat iaculis dapibus. Aliquam non risus ac urna interdum pretium. Donec maximus, massa vitae suscipit congue, ligula purus sagittis magna, vitae interdum enim dui sed felis. Curabitur mollis eleifend diam a varius. Vivamus eu porttitor lorem. Mauris sagittis ultrices ligula, a euismod mi efficitur a. Aenean molestie sapien a nibh suscipit, nec venenatis urna varius. Sed eu ex id lacus tincidunt ultricies. Aenean quam massa, dignissim ac suscipit posuere, varius sit amet turpis. Nunc semper turpis ullamcorper elit venenatis, in molestie nulla fermentum.

#### A.1.2 Control Module

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Pellentesque accumsan magna eu metus congue, vitae tristique enim commodo. Aliquam consectetur porttitor nisl, eget pulvinar quam venenatis eget. Aliquam erat volutpat. Nullam at ipsum at felis tincidunt consequat non sed mi. Morbi placerat iaculis dapibus. Aliquam non

risus ac urna interdum pretium. Donec maximus, massa vitae suscipit congue, ligula purus sagittis magna, vitae interdum enim dui sed felis. Curabitur mollis eleifend diam a varius. Vivamus eu porttitor lorem. Mauris sagittis ultrices ligula, a euismod mi efficitur a. Aenean molestie sapien a nibh suscipit, nec venenatis urna varius. Sed eu ex id lacus tincidunt ultricies. Aenean quam massa, dignissim ac suscipit posuere, varius sit amet turpis. Nunc semper turpis ullamcorper elit venenatis, in molestie nulla fermentum.

### **A.1.3 Decision Making Module**

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Pellentesque accumsan magna eu metus congue, vitae tristique enim commodo. Aliquam consectetur porttitor nisl, eget pulvinar quam venenatis eget. Aliquam erat volutpat. Nullam at ipsum at felis tincidunt consequat non sed mi. Morbi placerat iaculis dapibus. Aliquam non risus ac urna interdum pretium. Donec maximus, massa vitae suscipit congue, ligula purus sagittis magna, vitae interdum enim dui sed felis. Curabitur mollis eleifend diam a varius. Vivamus eu porttitor lorem. Mauris sagittis ultrices ligula, a euismod mi efficitur a. Aenean molestie sapien a nibh suscipit, nec venenatis urna varius. Sed eu ex id lacus tincidunt ultricies. Aenean quam massa, dignissim ac suscipit posuere, varius sit amet turpis. Nunc semper turpis ullamcorper elit venenatis, in molestie nulla fermentum.

### **A.1.4 Monitoring Module**

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Pellentesque accumsan magna eu metus congue, vitae tristique enim commodo. Aliquam consectetur porttitor nisl, eget pulvinar quam venenatis eget. Aliquam erat volutpat. Nullam at ipsum at felis tincidunt consequat non sed mi. Morbi placerat iaculis dapibus. Aliquam non risus ac urna interdum pretium. Donec maximus, massa vitae suscipit congue, ligula purus sagittis magna, vitae interdum enim dui sed felis. Curabitur mollis eleifend diam a varius. Vivamus eu porttitor lorem. Mauris sagittis ultrices ligula, a euismod mi efficitur a. Aenean molestie sapien a nibh suscipit, nec venenatis urna varius. Sed eu ex id lacus tincidunt ultricies. Aenean quam massa, dignissim ac suscipit posuere, varius sit amet turpis. Nunc semper turpis ullamcorper elit venenatis, in molestie nulla fermentum.

## A.2 Control Abstraction (Matlab)

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Pellentesque accumsan magna eu metus congue, vitae tristique enim commodo. Aliquam consectetur porttitor nisl, eget pulvinar quam venenatis eget. Aliquam erat volutpat. Nullam at ipsum at felis tincidunt consequat non sed mi. Morbi placerat iaculis dapibus. Aliquam non risus ac urna interdum pretium. Donec maximus, massa vitae suscipit congue, ligula purus sagittis magna, vitae interdum enim dui sed felis. Curabitur mollis eleifend diam a varius. Vivamus eu porttitor lorem. Mauris sagittis ultrices ligula, a euismod mi efficitur a. Aenean molestie sapien a nibh suscipit, nec venenatis urna varius. Sed eu ex id lacus tincidunt ultricies. Aenean quam massa, dignissim ac suscipit posuere, varius sit amet turpis. Nunc semper turpis ullamcorper elit venenatis, in molestie nulla fermentum.

## A.3 Decision Making and Monitoring Abstraction (Matlab)

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Pellentesque accumsan magna eu metus congue, vitae tristique enim commodo. Aliquam consectetur porttitor nisl, eget pulvinar quam venenatis eget. Aliquam erat volutpat. Nullam at ipsum at felis tincidunt consequat non sed mi. Morbi placerat iaculis dapibus. Aliquam non risus ac urna interdum pretium. Donec maximus, massa vitae suscipit congue, ligula purus sagittis magna, vitae interdum enim dui sed felis. Curabitur mollis eleifend diam a varius. Vivamus eu porttitor lorem. Mauris sagittis ultrices ligula, a euismod mi efficitur a. Aenean molestie sapien a nibh suscipit, nec venenatis urna varius. Sed eu ex id lacus tincidunt ultricies. Aenean quam massa, dignissim ac suscipit posuere, varius sit amet turpis. Nunc semper turpis ullamcorper elit venenatis, in molestie nulla fermentum.

## A.4 Probabilistic Two Module Model Generator (Python)

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Pellentesque accumsan magna eu metus congue, vitae tristique enim commodo. Aliquam consectetur porttitor nisl, eget pulvinar quam venenatis eget. Aliquam erat volutpat. Nullam at ipsum at felis tincidunt consequat non sed mi. Morbi placerat iaculis dapibus. Aliquam non risus ac urna interdum pretium. Donec maximus, massa vitae suscipit congue, ligula purus sagittis magna, vitae interdum enim dui sed felis. Curabitur mollis eleifend diam a varius. Vivamus eu porttitor lorem. Mauris sagittis ultrices ligula, a euismod mi efficitur a. Aenean molestie sapien a nibh suscipit, nec venenatis urna varius. Sed eu ex id lacus tincidunt ultricies. Aenean quam massa, dignissim ac suscipit posuere, varius sit amet turpis. Nunc semper turpis ullamcorper elit venenatis, in molestie nulla fermentum.

## A.5 Prism Automatic Model Checker (Python)

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Pellentesque accumsan magna eu metus congue, vitae tristique enim commodo. Aliquam consectetur porttitor nisl, eget pulvinar quam venenatis eget. Aliquam erat volutpat. Nullam at ipsum at felis tincidunt consequat non sed mi. Morbi placerat iaculis dapibus. Aliquam non risus ac urna interdum pretium. Donec maximus, massa vitae suscipit congue, ligula purus sagittis magna, vitae interdum enim dui sed felis. Curabitur mollis eleifend diam a varius. Vivamus eu porttitor lorem. Mauris sagittis ultrices ligula, a euismod mi efficitur a. Aenean molestie sapien a nibh suscipit, nec venenatis urna varius. Sed eu ex id lacus tincidunt ultricies. Aenean quam massa, dignissim ac suscipit posuere, varius sit amet turpis. Nunc semper turpis ullamcorper elit venenatis, in molestie nulla fermentum.

## A.6 Storm Automatic Model Checker (Python)

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Pellentesque accumsan magna eu metus congue, vitae tristique enim commodo. Aliquam consectetur portti-

tor nisl, eget pulvinar quam venenatis eget. Aliquam erat volutpat. Nullam at ipsum at felis tincidunt consequat non sed mi. Morbi placerat iaculis dapibus. Aliquam non risus ac urna interdum pretium. Donec maximus, massa vitae suscipit congue, ligula purus sagittis magna, vitae interdum enim dui sed felis. Curabitur mollis eleifend diam a varius. Vivamus eu porttitor lorem. Mauris sagittis ultrices ligula, a euismod mi efficitur a. Aenean molestie sapien a nibh suscipit, nec venenatis urna varius. Sed eu ex id lacus tincidunt ultricies. Aenean quam massa, dignissim ac suscipit posuere, varius sit amet turpis. Nunc semper turpis ullamcorper elit venenatis, in molestie nulla fermentum.

## A.7 Simulator (Python)

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Pellentesque accumsan magna eu metus congue, vitae tristique enim commodo. Aliquam consectetur porttitor nisl, eget pulvinar quam venenatis eget. Aliquam erat volutpat. Nullam at ipsum at felis tincidunt consequat non sed mi. Morbi placerat iaculis dapibus. Aliquam non risus ac urna interdum pretium. Donec maximus, massa vitae suscipit congue, ligula purus sagittis magna, vitae interdum enim dui sed felis. Curabitur mollis eleifend diam a varius. Vivamus eu porttitor lorem. Mauris sagittis ultrices ligula, a euismod mi efficitur a. Aenean molestie sapien a nibh suscipit, nec venenatis urna varius. Sed eu ex id lacus tincidunt ultricies. Aenean quam massa, dignissim ac suscipit posuere, varius sit amet turpis. Nunc semper turpis ullamcorper elit venenatis, in molestie nulla fermentum.

# Appendix B

## Code for the Advanced Driver Assistance Systems

### B.1 MDP Model Generators for Different ADAS (Python)

#### B.1.1 Perfect Decision Making

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Pellentesque accumsan magna eu metus congue, vitae tristique enim commodo. Aliquam consectetur porttitor nisl, eget pulvinar quam venenatis eget. Aliquam erat volutpat. Nullam at ipsum at felis tincidunt consequat non sed mi. Morbi placerat iaculis dapibus. Aliquam non risus ac urna interdum pretium. Donec maximus, massa vitae suscipit congue, ligula purus sagittis magna, vitae interdum enim dui sed felis. Curabitur mollis eleifend diam a varius. Vivamus eu porttitor lorem. Mauris sagittis ultrices ligula, a euismod mi efficitur a. Aenean molestie sapien a nibh suscipit, nec venenatis urna varius. Sed eu ex id lacus tincidunt ultricies. Aenean quam massa, dignissim ac suscipit posuere, varius sit amet turpis. Nunc semper turpis ullamcorper elit venenatis, in molestie nulla fermentum.



### **B.1.2 Imperfect Decision Making**

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Pellentesque accumsan magna eu metus congue, vitae tristique enim commodo. Aliquam consectetur porttitor nisl, eget pulvinar quam venenatis eget. Aliquam erat volutpat. Nullam at ipsum at felis tincidunt consequat non sed mi. Morbi placerat iaculis dapibus. Aliquam non risus ac urna interdum pretium. Donec maximus, massa vitae suscipit congue, ligula purus sagittis magna, vitae interdum enim dui sed felis. Curabitur mollis eleifend diam a varius. Vivamus eu porttitor lorem. Mauris sagittis ultrices ligula, a euismod mi efficitur a. Aenean molestie sapien a nibh suscipit, nec venenatis urna varius. Sed eu ex id lacus tincidunt ultricies. Aenean quam massa, dignissim ac suscipit posuere, varius sit amet turpis. Nunc semper turpis ullamcorper elit venenatis, in molestie nulla fermentum.

### **B.1.3 Additive Linear Control**

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Pellentesque accumsan magna eu metus congue, vitae tristique enim commodo. Aliquam consectetur porttitor nisl, eget pulvinar quam venenatis eget. Aliquam erat volutpat. Nullam at ipsum at felis tincidunt consequat non sed mi. Morbi placerat iaculis dapibus. Aliquam non risus ac urna interdum pretium. Donec maximus, massa vitae suscipit congue, ligula purus sagittis magna, vitae interdum enim dui sed felis. Curabitur mollis eleifend diam a varius. Vivamus eu porttitor lorem. Mauris sagittis ultrices ligula, a euismod mi efficitur a. Aenean molestie sapien a nibh suscipit, nec venenatis urna varius. Sed eu ex id lacus tincidunt ultricies. Aenean quam massa, dignissim ac suscipit posuere, varius sit amet turpis. Nunc semper turpis ullamcorper elit venenatis, in molestie nulla fermentum.

### **B.1.4 Additive Steering Control**

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Pellentesque accumsan magna eu metus congue, vitae tristique enim commodo. Aliquam consectetur porttitor nisl, eget pulvinar quam venenatis eget. Aliquam erat volutpat. Nullam at ipsum at felis tincidunt consequat non sed mi. Morbi placerat iaculis dapibus. Aliquam non risus ac urna interdum pretium. Donec maximus, massa vitae suscipit congue, ligula

purus sagittis magna, vitae interdum enim dui sed felis. Curabitur mollis eleifend diam a varius. Vivamus eu porttitor lorem. Mauris sagittis ultrices ligula, a euismod mi efficitur a. Aenean molestie sapien a nibh suscipit, nec venenatis urna varius. Sed eu ex id lacus tincidunt ultricies. Aenean quam massa, dignissim ac suscipit posuere, varius sit amet turpis. Nunc semper turpis ullamcorper elit venenatis, in molestie nulla fermentum.

## **B.2 Pareto Curve Generator (Python)**

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Pellentesque accumsan magna eu metus congue, vitae tristique enim commodo. Aliquam consectetur porttitor nisl, eget pulvinar quam venenatis eget. Aliquam erat volutpat. Nullam at ipsum at felis tincidunt consequat non sed mi. Morbi placerat iaculis dapibus. Aliquam non risus ac urna interdum pretium. Donec maximus, massa vitae suscipit congue, ligula purus sagittis magna, vitae interdum enim dui sed felis. Curabitur mollis eleifend diam a varius. Vivamus eu porttitor lorem. Mauris sagittis ultrices ligula, a euismod mi efficitur a. Aenean molestie sapien a nibh suscipit, nec venenatis urna varius. Sed eu ex id lacus tincidunt ultricies. Aenean quam massa, dignissim ac suscipit posuere, varius sit amet turpis. Nunc semper turpis ullamcorper elit venenatis, in molestie nulla fermentum.

## **B.3 Strategy Synthesis and Simulator (Python)**

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Pellentesque accumsan magna eu metus congue, vitae tristique enim commodo. Aliquam consectetur porttitor nisl, eget pulvinar quam venenatis eget. Aliquam erat volutpat. Nullam at ipsum at felis tincidunt consequat non sed mi. Morbi placerat iaculis dapibus. Aliquam non risus ac urna interdum pretium. Donec maximus, massa vitae suscipit congue, ligula purus sagittis magna, vitae interdum enim dui sed felis. Curabitur mollis eleifend diam a varius. Vivamus eu porttitor lorem. Mauris sagittis ultrices ligula, a euismod mi efficitur a. Aenean molestie sapien a nibh suscipit, nec venenatis urna varius. Sed eu ex id lacus tincidunt ultricies. Aenean quam massa, dignissim ac suscipit posuere, varius sit amet turpis. Nunc semper turpis ullamcorper elit venenatis, in molestie nulla fermentum.

## B.4 Prism Automatic Model Checker (Python)

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Pellentesque accumsan magna eu metus congue, vitae tristique enim commodo. Aliquam consectetur porttitor nisl, eget pulvinar quam venenatis eget. Aliquam erat volutpat. Nullam at ipsum at felis tincidunt consequat non sed mi. Morbi placerat iaculis dapibus. Aliquam non risus ac urna interdum pretium. Donec maximus, massa vitae suscipit congue, ligula purus sagittis magna, vitae interdum enim dui sed felis. Curabitur mollis eleifend diam a varius. Vivamus eu porttitor lorem. Mauris sagittis ultrices ligula, a euismod mi efficitur a. Aenean molestie sapien a nibh suscipit, nec venenatis urna varius. Sed eu ex id lacus tincidunt ultricies. Aenean quam massa, dignissim ac suscipit posuere, varius sit amet turpis. Nunc semper turpis ullamcorper elit venenatis, in molestie nulla fermentum.

# Appendix C

## Code for the Experimental Results and Evaluation

### C.1 Plot Generator for Human Driver Model (Python)

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Pellentesque accumsan magna eu metus congue, vitae tristique enim commodo. Aliquam consectetur porttitor nisl, eget pulvinar quam venenatis eget. Aliquam erat volutpat. Nullam at ipsum at felis tincidunt consequat non sed mi. Morbi placerat iaculis dapibus. Aliquam non risus ac urna interdum pretium. Donec maximus, massa vitae suscipit congue, ligula purus sagittis magna, vitae interdum enim dui sed felis. Curabitur mollis eleifend diam a varius. Vivamus eu porttitor lorem. Mauris sagittis ultrices ligula, a euismod mi efficitur a. Aenean molestie sapien a nibh suscipit, nec venenatis urna varius. Sed eu ex id lacus tincidunt ultricies. Aenean quam massa, dignissim ac suscipit posuere, varius sit amet turpis. Nunc semper turpis ullamcorper elit venenatis, in molestie nulla fermentum.

### C.2 Plot Generator for the Decision Making and Full Control ADAS (Python)

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Pellentesque accumsan magna eu metus congue, vitae tristique enim commodo. Aliquam consectetur porttitor nisl, eget pulvinar quam venenatis eget. Aliquam erat volutpat. Nullam at ipsum

at felis tincidunt consequat non sed mi. Morbi placerat iaculis dapibus. Aliquam non risus ac urna interdum pretium. Donec maximus, massa vitae suscipit congue, ligula purus sagittis magna, vitae interdum enim dui sed felis. Curabitur mollis eleifend diam a varius. Vivamus eu porttitor lorem. Mauris sagittis ultrices ligula, a euismod mi efficitur a. Aenean molestie sapien a nibh suscipit, nec venenatis urna varius. Sed eu ex id lacus tincidunt ultricies. Aenean quam massa, dignissim ac suscipit posuere, varius sit amet turpis. Nunc semper turpis ullamcorper elit venenatis, in molestie nulla fermentum.

# Bibliography

- [1] PRISM Manual | The PRISM Language | Example 1, Dec 2010.
- [2] John R Anderson. *The Architecture of Cognition*. Psychology Press, 2013.
- [3] John R Anderson, Michael Matessa, and Christian Lebiere. ACT-R: A theory of higher level cognition and its relation to visual attention. *Human-Computer Interaction*, 12(4):439–462, 1997.
- [4] Christel Baier and Joost-Pieter Katoen. *Principles of model checking*. MIT press, 2008.
- [5] Tina Balke and Nigel Gilbert. How do agents make decisions? A survey. *Journal of Artificial Societies and Social Simulation*, 17(4):13, 2014.
- [6] Ewin R Boer and Marika Hoedemaeker. Modeling driver behavior with different degrees of automation: A hierarchical decision framework of interacting mental models. In *Proceedings of the 17th European annual conference on human decision making and manual control*, pages 63–72, 1998.
- [7] Taolue Chen, Vojtěch Forejt, Marta Kwiatkowska, Aistis Simaitis, and Clemens Wiltsche. On stochastic games with multiple objectives. In *International Symposium on Mathematical Foundations of Computer Science*, pages 266–277. Springer, 2013.
- [8] Christian Dehnert, Sebastian Junges, Joost-Pieter Katoen, and Matthias Volk. A storm is coming: A modern probabilistic model checker. In *International Conference on Computer Aided Verification*, pages 592–600. Springer, 2017.
- [9] Vojtěch Forejt, Marta Kwiatkowska, Gethin Norman, and David Parker. Automated verification techniques for probabilistic systems. In *International School on Formal Methods for the Design of Computer, Communication and Software Systems*, pages 53–113. Springer, 2011.

- [10] Iuliia Kotseruba and John K Tsotsos. A review of 40 years of cognitive architecture research: Core cognitive abilities and practical applications. *arXiv preprint arXiv:1610.08602*, 2016.
- [11] Marta Kwiatkowska, Gethin Norman, and David Parker. Prism: Probabilistic symbolic model checker. In *International Conference on Modelling Techniques and Tools for Computer Performance Evaluation*, pages 200–204. Springer, 2002.
- [12] Marta Kwiatkowska, Gethin Norman, and David Parker. Stochastic model checking. In *International School on Formal Methods for the Design of Computer, Communication and Software Systems*, pages 220–270. Springer, 2007.
- [13] Marta Kwiatkowska, Gethin Norman, and David Parker. Prism 4.0: Verification of probabilistic real-time systems. In *International conference on computer aided verification*, pages 585–591. Springer, 2011.
- [14] Chak Lam. Driver assistance using cognitive modelling and strategy synthesis.
- [15] Andrew Liu and Dario Salvucci. Modeling and prediction of human driver behavior. In *Intl. Conference on HCI*, 2001.
- [16] Dario Salvucci, Erwin Boer, and Andrew Liu. Toward an integrated model of driver behavior in cognitive architecture. *Transportation Research Record: Journal of the Transportation Research Board*, (1779):9–16, 2001.
- [17] Dario D Salvucci. Modeling driver behavior in a cognitive architecture. *Human factors*, 48(2):362–380, 2006.
- [18] Dario D Salvucci and Andrew Liu. The time course of a lane change: Driver control and eye-movement behavior. *Transportation research part F: traffic psychology and behaviour*, 5(2):123–132, 2002.
- [19] Ron Sun. The importance of cognitive architectures: An analysis based on CLARION. *Journal of Experimental & Theoretical Artificial Intelligence*, 19(2):159–193, 2007.