POLITECNICO DI TORINO

Master of science
in Ingegneria Matematica

Master's Thesis

Infer causality relationships in the motion of manned and unmanned vehicles using transfer entropy

Supervisors
Prof. Alessandro Rizzo
Dr. Stefano Primatesta

Candidate
Denise Tumiotto

Academic Year 2018-2019

To anyone who believed in me.

Summary

Recently, the interest in autonomous vehicles is steadily increasing. Many studies are being conducted, and researchers are investigating the relationship between automatic vehicles and human beings. Those studies focus on the relationship between autonomous vehicles and pedestrians, or between the drivers of vehicles and their autonomous counterpart. In fact, the presence of autonomous vehicles on the road is already a reality in some countries and studying how we can improve their efficiency is key for future developments in the automotive and robotic fields. In this thesis, we study the relationship between autonomous and manned vehicles, since it is unlikely that since the inception of this technology autonomous vehicles will be the only ones moving on our street grids.
Here, we aim at understanding the causal relationship between unmanned and manned vehicles. In order to characterize this relationship, we first apply a symbolic analysis to vehicle trajectories and, subsequently, we carry out a transfer entropy analysis. We define and use many types of symbols to code vehicle trajectories, and we test them, assessing their performance in four different cases: when the two vehicles proceed along parallel routes, two cases in which two vehicles move in opposite directions, facing each other, and the case in which the two vehicles intersect their routes. Then, we make a parametric study using one of the symbols that better performs in the first study. In particular, we analyze those parameters that characterize each type of symbols and we study how the transfer entropy varies with the parameters. We obtain interesting results. In our results, it is clear that approximately two meters before the encounter there is a flow of information from manned to unmanned vehicle that causes the variation in the trajectory of the latter. The manned vehicle entails the movement of the unmanned one, and during the encounter the unmanned vehicle moves away before the manned one. Indeed, the transfer entropy method is able to detect not only the instant, but also the direction in the flow of information between the vehicles. This information allows us to infer the causal relationship between the vehicles.

Acknowledgements

There are many people I would like to thank for my academic path.

First I thank my supervisors. Professor Alessandro Rizzo gave me a big opportunity in writing this thesis under his supervision and he gave me wise pieces of advice. I always appreciate his corrections and suggestions. Doctor Stefano Primatesta helped me through the entire project with patience and positiveness, even when I wasn't sure of my work.

Then, I thank my parents. Obviously, without them I would never accomplish this goal. They gave me not only economic support, but most of all they gave me their trust. They listened to me, even if they didn't understand completely. They gave me support, when I wasn't confident. They taught me valuable lessons on face problems with strength, and never give up. They also taught me the value of hard work, which was essential during the entire journey.

Finally, I want to thank my friends: from high school, from scouts, from Polytechnic. They all shared with me unforgettable moments. However, I want to cite Rosita, my best friend. I think of her as a sister, she is always prepared to listen to my complain and my rants.

At least, the most valuable person in my life. I thank with all my heart my boyfriend Umberto. Since we met, he become my best friend, my lover, and first supporter. He helps me more than anyone to overcome moments of discouragement.

Contents

List of Figures

Chapter 1

Introduction

1.1 Motivation and related works

Recently, the presence of robots in our daily life is increasing. From the simpler ones, such as the robotic vacuum cleaner, to the more complex, such as autonomous cars, see Figure 1.1. This sector of research is called Service Robotics. A service

(a) Robotic vacuum cleaner

(b) Autonomous vehicle prototype

Figure 1.1: Examples of different autonomous robots.

robot is a robot that directly interact with humans, offering them services. Usually, service robots do not perform repetitive tasks as industrial robots. On the contrary, they operate in a human environment and cooperate with humans. The dynamic and unstructured environments in which these types of robots work is a key challenge in research.
Another type of robots are the industrial ones. Many robotic arms help human work and, in some cases, they totally replace humans. For example, in the automobile assembly line many tasks, like the welding of the body, the assembly of a windscreen and the production of many pieces is completely autonomous, Figure 1.2. Nevertheless, there are many issues that remain unknown, and many studies

Figure 1.2: Robotic arms in the automotive industry.

are performed to improve robot interaction with humans and their productivity. Not only the presence of robots can help humans with their work, but it can also provide a safer environment. In fact, it is likely to think that the presence of autonomous vehicles on the streets will reduce the number of accidents, even if the majority of people don't trust completely this type of transportation.

The importance of trust in human-robot interaction is studied in depth in many researches. In particular, [1, 2] analyze in different ways the trust issue. The former approaches the problem from the human point of view, in particular on how to repair the trust between humans and robots after a major violation by the machine. The authors propose a study on how the trust can be violated, and a model of the expected trust repair rate. The second article focuses more on quantify the trust. First, the authors conduct a meta-analysis of the empirical studies. The results reveal that human-robot trust is important and it depends on many factors. In particular, it varies with the robot characteristics, as, for example, performance, appearance, and proximity. In a second moment, the authors try to measure the trust. They focus on what they measure, because in the literature there are multiples type of trust. The authors also ask themselves how to measure the trust

(subjectively, objectively, etc) and when to measure it (prior to the interaction, during the interaction, after the interaction, multiple times).
Moreover, autonomous cars are useful for many other reasons: they facilitate the transportation of those people which cannot drive, they decrease congestion and the number of parking lots, as the vehicle can return autonomously to its starting location and it will reduce the overbuilding of the areas and, generally, it improves the quality of life. On the other hand, there are not only positive consequences, as we well know it is very important to understand if an automatic car can take the right decision and if there exists always the right decision [3].
The process of decision making is deeply studied and it starts from the understanding of human decisions. In [4], the authors try to generate a human-like motion planning. To reach this goal they use a game theory approach and, in particular, they refer to the problem as a game, in which solutions decided by humans are the Nash equilibria. On the other hand, in [5] they focus on how the automatic machines should communicate their decisions to humans. In fact, they underline the importance for humans to understand why a certain decision is taken. Not only it has to explain why the agents do a certain action, but also it has to explain what is different from the previous plan, it is called the explainable planning.

1.1.1 Detecting Causality

What we want to add to the existent literature is the answer to the question "Is it possible to understand the causal relationship between manned and unmanned vehicles?". The problem is to know who causes the movement of whom in the encounter of two vehicles. The study of causality is a problem of central importance, and understanding why something happens or acts in a certain way is crucial in many fields of science. For what concern the probability, it is fundamental the distinction between causality and correlation. In fact, two variables can be correlated, but this does not implicate that they are caused by each other. The causality relationship implies that a certain variable changes its value *because* of another one. Correlation does not entail causality: two series can be correlated, that means they do have a certain relationship, but it is possible that they do not cause each other, [6]. We will concentrate on causality because we want to detect it.
In [7], the authors explain in details how the symbolic representation works and how to apply the causation entropy to it. The evaluation of causality is exemplified using the tent map. In [8], Rounge et al. present different types of causal inference methods. They start showing the Granger Causality method, which can detect the causal relationship between stochastic time series only if the relationship is linear. Its nonlinear counterpart is the transfer entropy method [9]. Another method is the convergent cross-mapping [10]. In this case, it is necessary to find

the deterministic nonlinear attractor from the data, and demonstrate that the two dynamical variables belong to the same dynamical system. Finally, they present the causal network learning algorithms, as Peter and Clark Algorithm [11], and the structural causal models, as in [12, 13, 14]. The former are based on the reconstruction of large-scale causal graphical models, with two assumptions: the Markov condition and the Faithfulness assumption, [15]. The latter methods are able to discover causality even when the causation occurs instantaneously, because assumptions on the class of the model are made.

1.1.2 Using Transfer Entropy

The transfer entropy method is one of the most accredited and used in the literature. This method was never used for the purpose we aim to achieve, but it is well tested in many other fields.
In [16], transfer entropy is used to determine in an experiment with a zebrafish and a robotic replica which is the leader. In the experiment the two fish, one real and one robotic, move in two different portions of the same aquarium. Their trajectories are coded with a symbolic analysis [17] and, then, the transfer entropy is evaluated to determine which fish follows the other and understand who is the leader.
The same method is applied to a different field in [18]. Here, the authors focus their work on the policy diffusion in USA, represented as a network in which each node is a State and the links between nodes represent information flows. This is a problem of network reconstruction. They demonstrate that transfer entropy quantifies the information transferred between nodes, and it is possible to predict the causal relationship in policymaking.
Another field in which transfer entropy reveals efficient is neuroscience. For example, in [19] transfer entropy is used in contrast with Granger Causality and the dynamic causal modeling. Both are alternative methods to evaluate causality, but they both require previous knowledge of the network and input. Moreover, Granger Causality is not able to detect a nonlinear causal relationship. The authors prove that transfer entropy is able to overtake those *a priori* problems.
Moreover, in [20] the authors actually demonstrate that Granger Causality and transfer entropy are equivalent measures for Gaussian variables.
In conclusion, given the good results in different fields, we decide to use transfer entropy for this study. Here, we want to understand the causal relationship between manned and unmanned vehicles. Using transfer entropy we are able to tell, during the encounter of the vehicles, which one causes the maneuver of the other. We analyze four different scenarios: two scenarios in which the trajectories have opposite direction; one scenario with two parallel trajectories; and, finally, a scenario in which the two trajectories intersect.

1.2 Thesis outline

In this section we describe the organization of this thesis. It is mainly divided into two bigger chapters and a last one dedicated to the conclusions.

- In **chapter 2** we describe the mathematical tool in details, such as what symbols do we use and we give the definition of transfer entropy. Moreover, we explain how transfer entropy is evaluated and used to find causal relationship. We present symbol coding systems that use up to six different symbols. For each one we give an example to understand how we used it;
- In **chapter 3** we introduce **ROS**, Robot Operating System, used to execute the experimental simulation. We adopt it to perform our simulations and to extract the data. We present the simulations we made, studying first which symbol system performs better and, then, we propose a parametric study using the selected symbols in the previous part. Therein, we explain and interpret the obtained results;
- Finally, in **chapter 4** we discuss a possible future development using the study done in this thesis. We discuss about a possible development of the work, and possible future applications.

Chapter 2

Methodology

The mathematical tools for analyzing our problem are most of all two: the symbolic analysis and the use of transfer entropy. The first one is very valuable to analyze a dynamical system simplifying the expression of it. The second one is one of the main tools to understand the causality between coupled dynamical systems.
We now describe in details them, focusing on how they will be used in the following chapter.

2.1 Symbolic analysis

The symbolic analysis aims to simplify the given data. Using a symbol instead of a pair of coordinates improves the way in which the data can be used. In fact, some properties are highlighted by the frequency of the symbols, and that is why it is simpler to analyze them. As defined in [7], the symbolization discretizes a continuous variable in a finite partition of the space. That means that if we indicate the space with M a partition is

$$M = \bigcup_{i=0}^{m} \overline{A}_i$$

where $\mathcal{A} = \{A_0, \ldots, A_m\}$ is a finite set of disjoint open sets.
In our case we collect the trajectory as $x-$ and $y-$ coordinates at discrete time steps of 4 Hz each, then we elaborate this data to work easily with them. In the following, we will identify the partition and then we propose an explanatory example.

2.1.1 Symbol 1

The first used approach is based on the symbolic trajectories presented by [16]. We consider a time series of coordinates, where a coordinate (x_t, y_t) is the position

of the robot at time t. First of all, we compute the ideal trajectory having as starting point (x_1, y_1) and a fixed goal (x_n, y_n), where n is the number of total time steps. Then, we choose a positive integer $m \ll n$, and we select m time steps from (x_k, y_k) to (x_{k+m}, y_{k+m}). For each position we determine the distance d from the ideal trajectory

$$d_k = \frac{|ax_k + by_k + c|}{\sqrt{a^2 + b^2}} \tag{2.1}$$

where $a,\ b,\ c$ are the coefficients of the line $ax + by + c = 0$ that determines the ideal trajectory. After that, we associate a symbol i with the following conditions:

1. $d_{t+i_1} \leq d_{t+i_2} \leq \cdots \leq d_{t+i_m}$
2. $i_{s-1} < i_s \quad \text{if } d_{t+i_{s-1}} = d_{t+i_s}$

The partition is the collection of $m!$ tuples originated from the permutation of the digits $\{0, \ldots, m\}$. In Figure 2.1 a graphical representation of the reworking of the data is depicted.

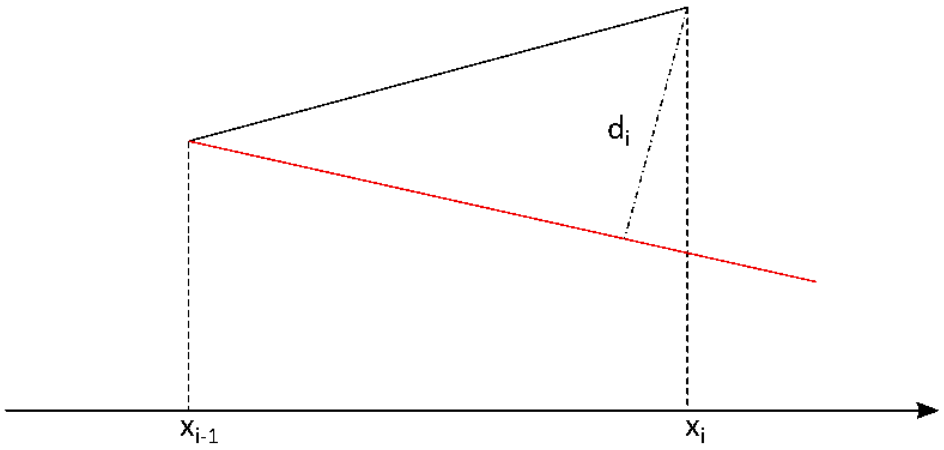


Figure 2.1: A portion of trajectory to identify the distance used for determine the *Symbol 1*. In red the ideal trajectory.

Example 1

Suppose to have the following positions: $X = (1,\ 3,\ 4,\ 6,\ 7,\ 8,\ 9)$ and $Y = (2,\ 4,\ 3,\ 5,\ 3.5,\ 1.5,\ 2)$. In Figure 2.2 is represented the trajectory. The ideal trajectory in this case is the segment from $x = 1$ to $x = 9$ in the line $y = 2$. Let's fix $m = 3$, so the total symbol are $m! = 3! = 6$. In this example the

distance is $d = |y_k - 2|$. So we first focus on the coordinates $(1;2)$, $(3;4)$, $(4;3)$, the respective distances from the ideal trajectory are $d_1 = 0$, $d_2 = 2$, $d_3 = 1$, and the symbol that follows the previous rules is (0,2,1). The second coordinates in exam are $(3;4)$, $(4;3)$, $(6;5)$, with distances $d_2 = 2$, $d_3 = 1$, $d_4 = 3$, so the symbol will be (1,0,2). The following symbols are evaluate in a similar way and they are $s = ((0\ 2\ 1), (1\ 0\ 2), (0\ 2\ 1), (2\ 1\ 0), (2\ 1\ 0))$, as shown in Figure 2.3.

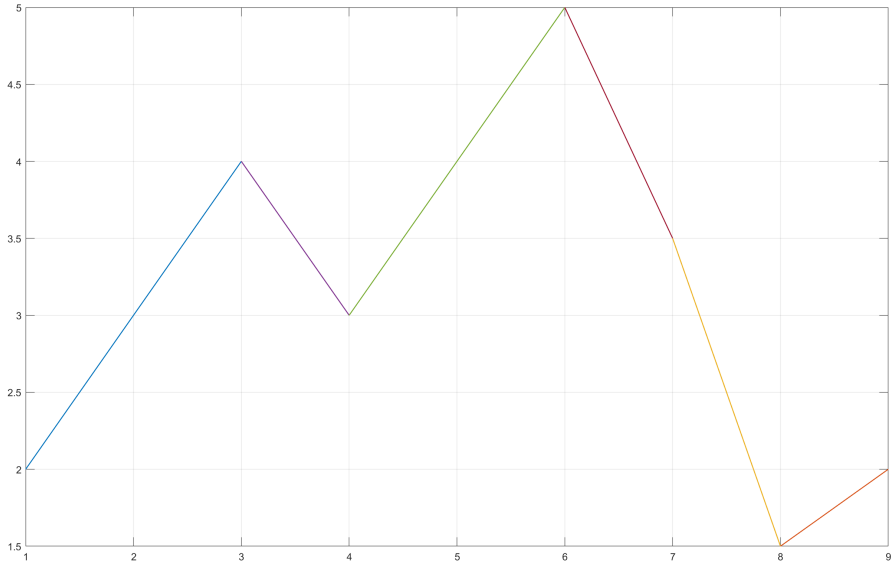


Figure 2.2: Example of a trajectory.

2.1.2 Symbol 2

The second approach is based on the work proposed in [21]. We evaluate the *relative distances* (σ_i) as

$$\sigma_i = \frac{\sqrt{(y_{i+1} - y_i)^2 + (x_{i+1} - x_i)^2}}{max_{i \in \{1,\dots,n\}}(\sqrt{(y_{i+1} - y_i)^2 + (x_{i+1} - x_i)^2})} \tag{2.2}$$

where (x_i, y_i) is the position of the vehicle at the $i - th$ step and the *movement directions* (θ_i) is calculate with the following formula

$$\theta_i = \begin{cases} \arctan\left(\frac{y_{i+1}-y_i}{x_{i+1}-x_i}\right) & x_{i+1} - x_i \geq 0 \\ \arctan\left(\frac{y_{i+1}-y_i}{x_{i+1}-x_i}\right) - \pi & x_{i+1} - x_i < 0 \text{ and } y_{i+1} - y_i \leq 0 \\ \arctan\left(\frac{y_{i+1}-y_i}{x_{i+1}-x_i}\right) + \pi & x_{i+1} - x_i < 0 \text{ and } y_{i+1} - y_i > 0 \end{cases} \tag{2.3}$$

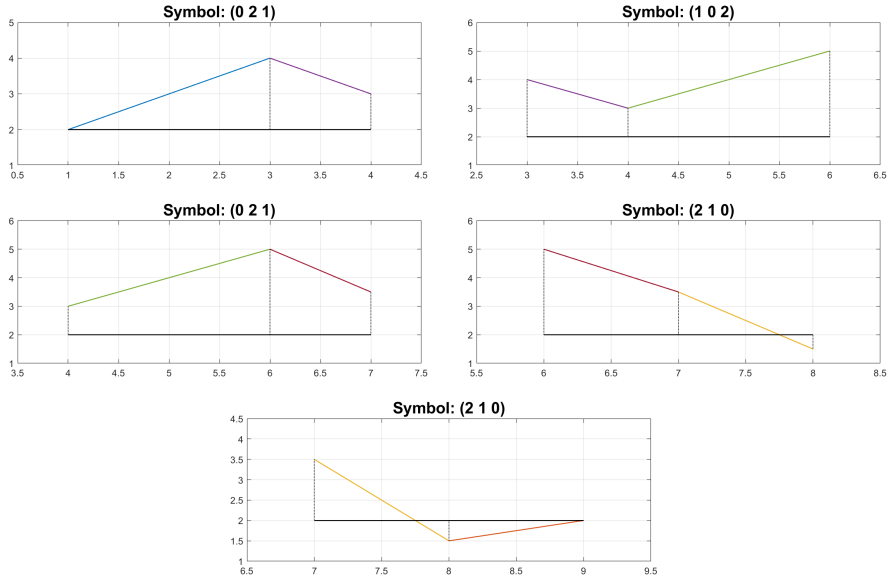


Figure 2.3: Example of symbols considering the trajectory of Figure 2.2. $m = 3$ and the ideal trajectory is $y = 2$ (the black line). The symbols are given evaluating the distance of each points from the ideal trajectory and then ordering those distances in increasing order.

The *relative distances* and the *movement directions* take value respectively in [0,1] and $[-\pi, \pi]$. Both σ and θ are graphically detailed in Figure 2.4. At this point we

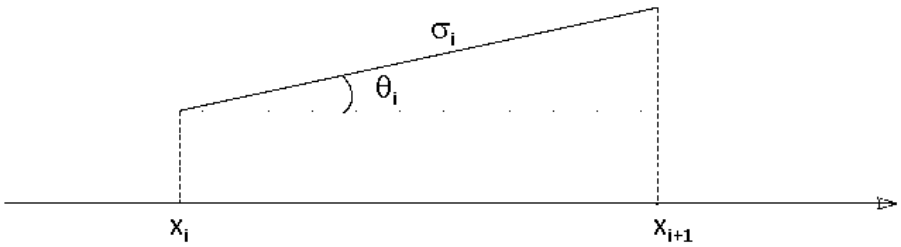


Figure 2.4: A portion of trajectory to identify the distance and the angle used for determine the *Symbol 2*.

give two values ϵ_{dist} and ϵ_{dir}, and they are used to divide the intervals, $2\pi/\epsilon_{dir} \times 1/\epsilon_{dist}$, obtaining a matrix as in Figure 2.5. In this way the symbol is defined by the number associate with the coordinate (θ_i, σ_i). Suppose to consider the example

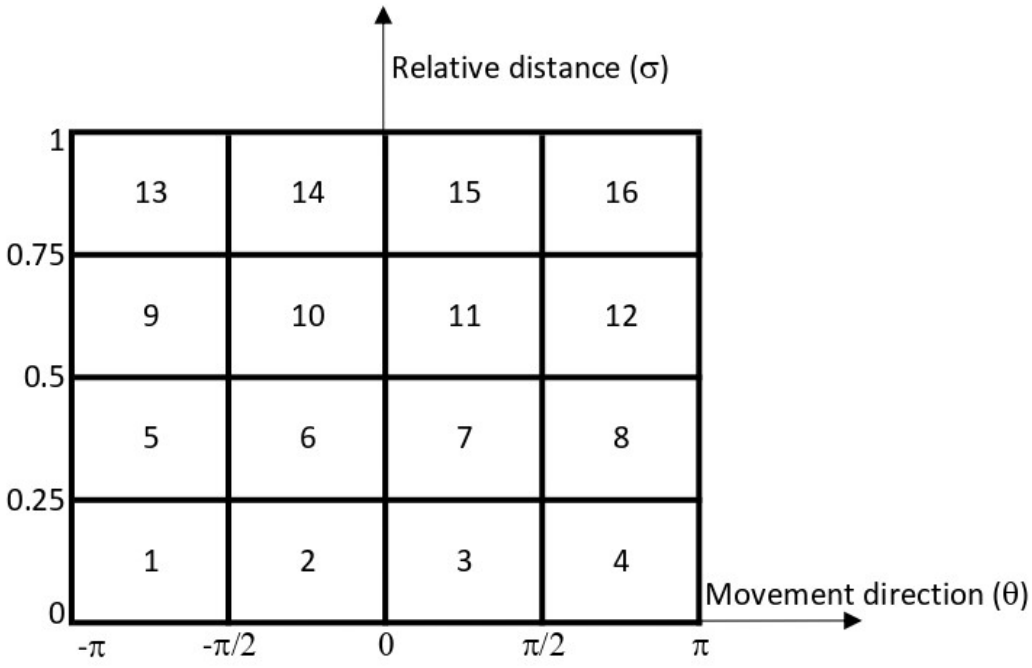


Figure 2.5: On the x-axis the movement direction with step $\epsilon_{dir} = \pi/2$, on the y-axis the relative distance with step $\epsilon_{dist} = 0.25$.

1, shown in Figure 2.2. If we fix $\epsilon_{dir} = \pi/2$ and $\epsilon_{dist} = 0.25$, then the number of symbol are $2\pi/\epsilon_{dir} \times 1/\epsilon_{dist} = 4 \cdot 4 = 16$. The first *movement direction* is

$$\theta_1 = \arctan(\frac{4-2}{3-1}) = \frac{\pi}{4};$$

while the first *relative distance* is

$$\sigma_1 = \frac{\sqrt{(4-2)^2 + (3-1)^2}}{2\sqrt{2}} = 1.$$

The couple $(\pi/4,\ 1)$ corresponds in Figure 2.5 to the symbol "15". Similarly, the following symbols are, and for the cited example the symbolic trajectory will be $s = (15,\ 6,\ 15,\ 10,\ 14,\ 7)$.

2.1.3 Symbol 3

This type of symbol is a different approach that emulates the *Symbol* 2. Using as before the coordinates (x_i, y_i), we evaluate the ideal trajectory, that is the straight line from the starting position to the goal, determined by the last position in the vector of coordinates. We establish the angle between the actual trajectory, with slope m_i, and the ideal one, with a slope of m_{ideal}

$$\theta_i = \begin{cases} |\arctan(m_i) - \arctan(m_{ideal})| & m_i \cdot m_{ideal} > 0 \\ |\arctan(m_i)| + |\arctan(m_{ideal})| & m_i \cdot m_{ideal} \leq 0 \end{cases} \tag{2.4}$$

and we evaluate the relative distance, as in equation (2.2). In this case the *movement direction* θ_i is always positive and it takes values in $[0, \pi]$. Again, in Figure 2.6 is illustrated the representation of the data in a portion of a possible trajectory. As before, we fix ϵ_{dir} and ϵ_{dist} and, then, we associate a symbol that

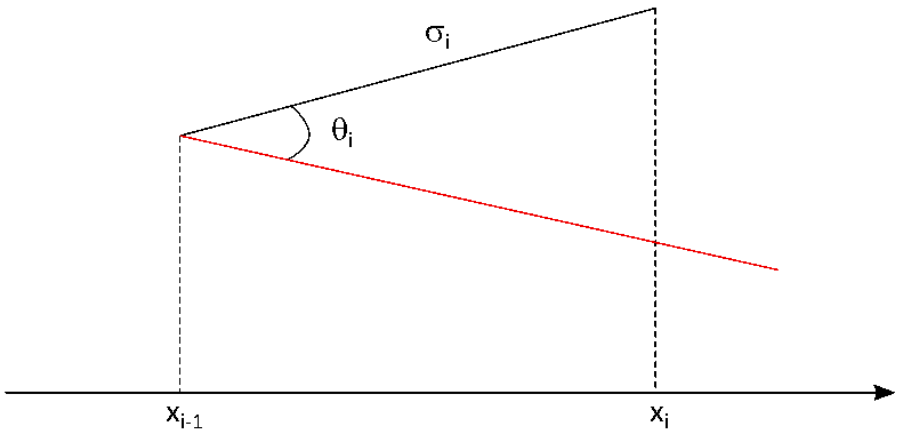


Figure 2.6: A portion of trajectory to identify the distance used for determine the *Symbol 3*. In red the ideal trajectory.

corresponds to one of the entry of a matrix similar to the one in Figure 2.5. Specifically, the interval is $[0, \pi]$ and the related matrix is reported in Figure 2.7.

Referring to the example 1, we now evaluate the symbols with the last described approach. In this specific case the *ideal angle* is zero, and the angle between the actual trajectory and the ideal one is simply the absolute value of the angle defined by the trajectory with the x-axis. Let's fix $\epsilon_{dir} = \pi/4$ and $\epsilon_{dist} = 0.25$, then the number of symbol are $\pi/\epsilon_{dir} \times 1/\epsilon_{dist} = 4 \cdot 4 = 16$. The first couple of value (θ_1, σ_1)

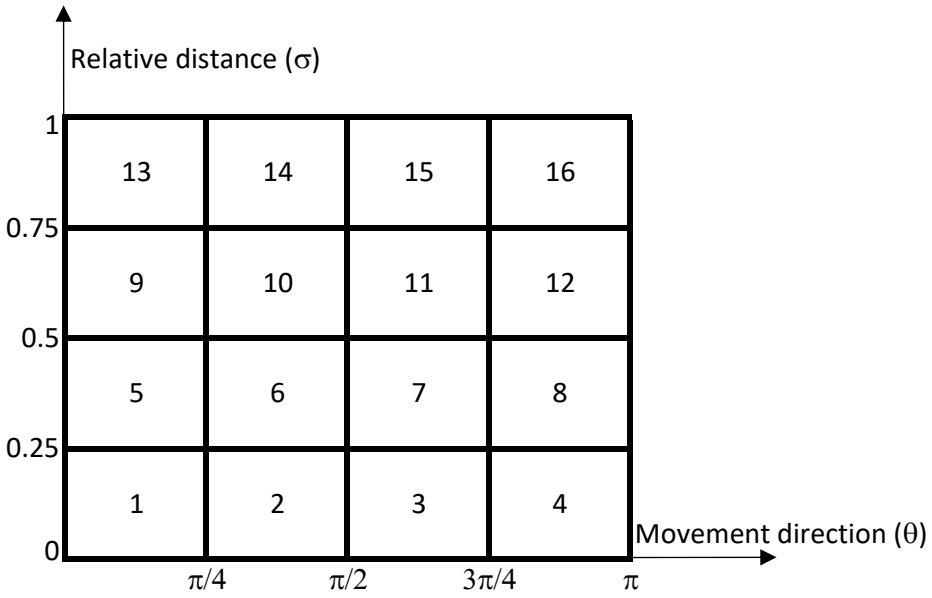


Figure 2.7: On the x-axis the movement direction with step $\epsilon_{dir} = \pi/4$, on the y-axis the relative distance with step $\epsilon_{dist} = 0.25$.

is

$$\theta_1 = \arctan\left(\frac{4-2}{3-1}\right) = \frac{\pi}{4};$$

$$\sigma_1 = \frac{\sqrt{(4-2)^2 + (3-1)^2}}{2\sqrt{2}} = 1.$$

Even if the couple is exactly the same as before the symbol is different. In fact, the matrix has different axis and the values are shifted. The first symbol is 13 and the string of all the symbols that describe the trajectory is $s = (13,\ 5,\ 13,\ 10,\ 14,\ 5)$.

2.1.4 Symbol 4

This symbol is determined using a process similar to the previous symbol. The *movement direction* θ_i is the same evaluated in equation (2.4). This time the *relative distance* is the distance between each point and the ideal trajectory

$$\sigma_i = \frac{|ax_i + by_i + c|}{\sqrt{a^2 + b^2}} \tag{2.5}$$

where a, b and c are the coefficient of the line $ax + by + c = 0$, that identifies the ideal trajectory. With this type of symbol we focus on how much the vehicle diverges from the trajectory. In Figure 2.8 we individuate the distance and the angle on a portion of a possible trajectory, in order to show how the data are manipulated. Even if the *relative distance* is different, the matrix which contains

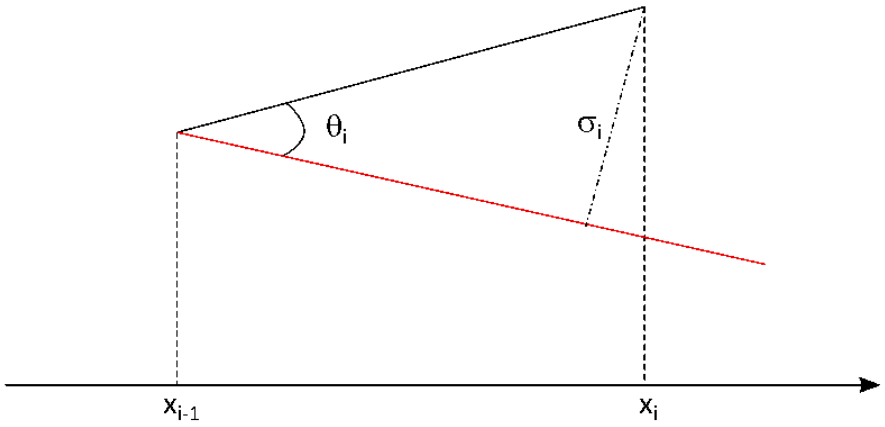


Figure 2.8: A portion of trajectory to identify the distance and the angle used for determine the *Symbol 4*. In red the ideal trajectory.

the symbol is the same shown in Figure 2.7. Anyway in this specific case the y-axis correspond to the distance from the ideal trajectory at the end of the run segment instead of the actual distance traveled.
Evaluating, as before, the symbols for the example 1 we obtain the following symbolic vector $s = (9,\ 5,\ 13,\ 6,\ 2,\ 1)$. In this case, fixing $\epsilon_{dir} = \pi/2$ and $\epsilon_{dist} = 0.25$, the number of symbol are $2\pi/\epsilon_{dir} \times 1/\epsilon_{dist} = 4 \cdot 4 = 16$ again. In Figure 2.9 is shown the angle θ and the distance σ.

2.1.5 Symbol 5

This type of symbol is simpler than the others, and it takes into consideration only the distance from the actual trajectory to the ideal trajectory. In particular, the distance is divided into discrete steps. Each step has a different symbol, from zero, if the vehicle is very close to the ideal trajectory, up to three if the vehicle is getting far away to the ideal trajectory.
It is evident that this symbol is not very specific and it simplifies a lot, maybe too much, the trajectory losing much information. In the following chapter, we will study how those symbols perform and it would be clear how many information we

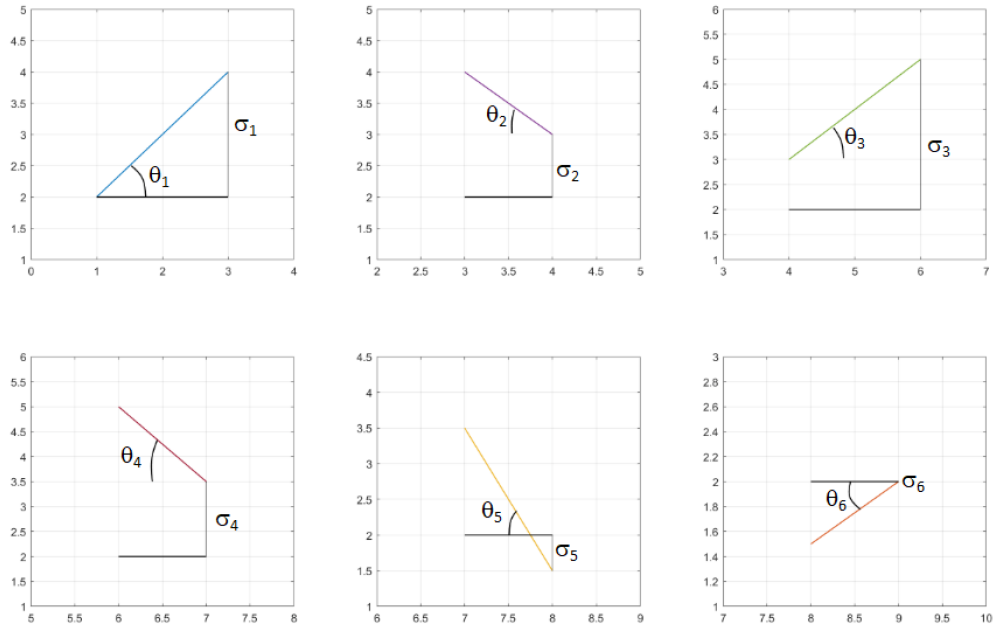


Figure 2.9: For each portion of the trajectory represented in Figure 2.2, the *relative distance* σ and the *movement direction* θ are shown, based on the proposed method to compute the *Symbol 4*.

need to keep in order to have useful data.
As before, we refer to example 1 in order to better explain how the symbol works. In Figure 2.10 we can obtain graphically the symbolic vector $s = (0,\ 2,\ 1,\ 3,\ 1,\ 0,\ 0)$. We underline a few things: first of all, when a position is exactly on the border to the new value it takes the bigger value, on the other hand being over or under the ideal line it doesn't change the value of the symbol, instead the symbols are symmetric with respect to the ideal trajectory. This means that even if we know how far, more or less, the vehicle is from its ideal trajectory we don't know which direction it takes.

2.1.6 Symbol 6

Finally, we derived a last symbol, which is different but related to the previous ones. In particular, we define it as a 3-dimensional symbol, just because we evaluate not only *relative distances* and *movement directions*, but also the velocity. In fact, the third value evaluates the distance traveled from each point and the next one.

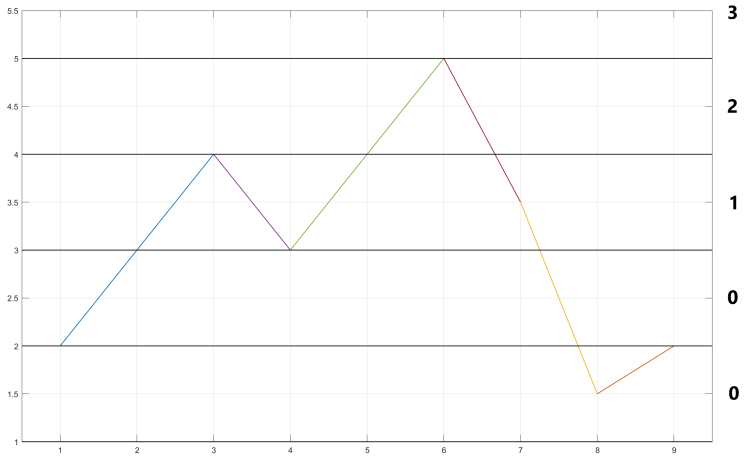


Figure 2.10: The symbols $\{0, 1, 2, 3\}$ are given by the position with respect to the ideal line.

Hence, each point corresponds at the same period of time and the space traveled corresponds to the scaled velocity. Summarizing, we evaluate the *relative distances* as in equation (2.5), the *movement direction* as in equation (2.4) and the *scaled velocity* (v_i), as

$$v_i = \sqrt{(y_{i+1} - y_i)^2 + (x_{i+1} - x_i)^2} \tag{2.6}$$

As in the previous symbols, in Figure 2.11 we illustrate the symbol, highlighting the parameters used to compute the symbol 6. After evaluating all the data, we associate a number, which will be the symbol for the segment, starting with a sort of three-dimensional matrix as in Figure 2.12. In this case, even the evaluation for a simple case as the example 1 it is not trivial. Fixing $\epsilon_{dir} = \pi/2$, $\epsilon_{dist} = 0.25$ and $\epsilon_{vel} = 0.25$, we obtain a total number of symbols of $2\pi/\epsilon_{dir} \times 1/\epsilon_{dist} \times 1/\epsilon_{vel} = 4 \cdot 4 \cdot 4 = 64$. If we calculate the first values we obtain

$$\theta_1 = \arctan\left(\frac{4-2}{3-1}\right) = \frac{\pi}{4}$$

$$\sigma_1 = \frac{2}{3}$$

$$v_1 = \frac{\sqrt{(4-2)^2 + (3-1)^2}}{2\sqrt{2}} = 1$$

The corresponding symbol is 57, the following are $s = (57,\ 21,\ 61,\ 38,\ 50,\ 17)$.

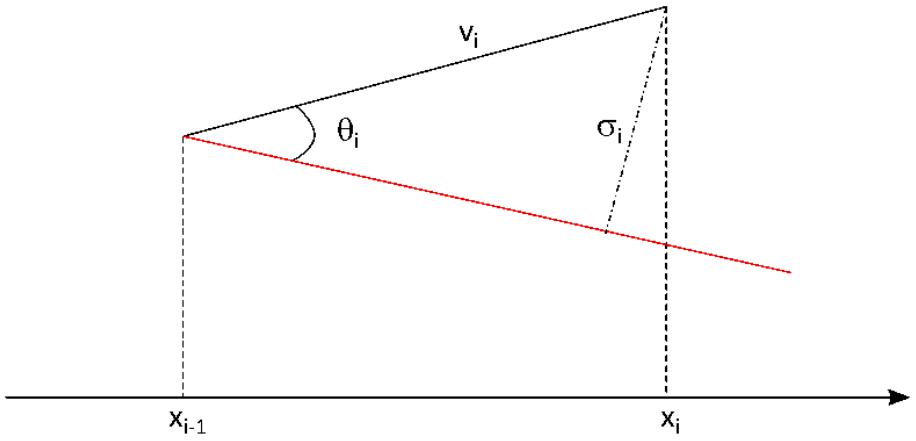


Figure 2.11: A portion of trajectory to identify the distance, the velocity and the angle used for determine the *Symbol 6*. In red the ideal trajectory.

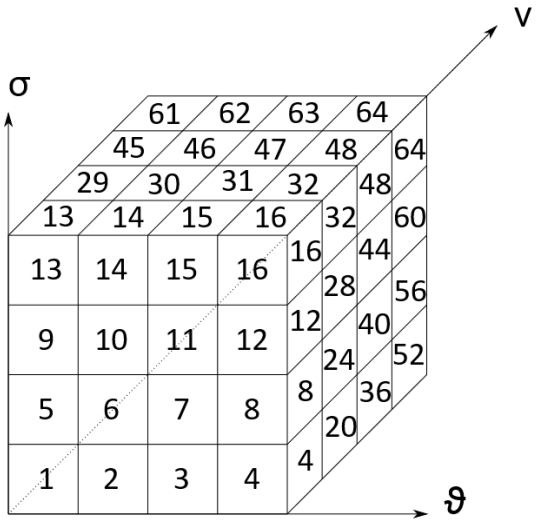


Figure 2.12: The three axis are respectively the *Relative Distance* (σ), the *Movement Direction* (θ) and the *Scaled Velocity* (v). The steps in which every dimensions is plitted are respectively $\epsilon_{dist} = 0.25$, $\epsilon_{dir} = \pi/4$, and $\epsilon_{vel} = 0.25$.

We observe that in *Symbol 1* the total number of symbols used in order to describe the trajectory is $n - m + 1$, where n is the total number of elements in the time series and m is the integer that divides the series. In *Symbol 5* there are exactly n symbols. In the others, the number of symbols needed for completely

describes the trajectory is $n - 1$. We can suppose that the first one could be more solid even if less accurate than the others, while the fifth could be too much sensible to differences in trajectory.

2.2 Transfer Entropy

Transfer entropy is a non-parametric measure of directed, asymmetric information transfer between two processes. In particular, it can be used for coupled dynamical systems in order to identify the direction of information flow and to infer which process causes the other. In this work, we want to understand how many information the two robot exchange and, if possible, which one influence the motion of the other.
For this purpose, let s be the vector with all the symbols of the time sequence and γ a symbol of the space Γ, the space with all the possible symbols. We define the probability of each symbol in the sequence as

$$p(\gamma) = \frac{\sharp\{s_i \text{ is of type } \gamma\}}{\sharp\{s\}} \tag{2.7}$$

where $\sharp$ is the cardinality of the set. This value, $p(\gamma)$ is actually the frequency of the symbol γ. Since we do not know the relation in the movement of the robots, evaluating the frequency of the symbols is the only way to determine the probability of them.

Using the equation (2.7), the Shannon entropy, [9], can be defined as

$$h(s) = -\sum_{\gamma \in \Gamma} p(\gamma) \log(p(\gamma)). \tag{2.8}$$

In equation (2.8) the logarithm has base e, then the unit of measure will be *nat*. In case of a base 2, the unit of measure would have been *bit*. This value is always non-negative. Indeed, since $p(\gamma) \leq 1$ the logarithm will always be non-positive, and the minus before the sum permits the entropy to have a non-negative value.
Anyway, we are more interested in calculating the permutation transfer entropy [pTE] from one sequence $\{y_i\}$ to another $\{x_i\}$. In fact, this value quantifies the information flow between the processes. If pTE is different from zero it means that knowing the process y helps to understand and predict the process x. On the other case, if the pTE is null it means that the process y doesn't add any information to the process x. The formula to derive the pTE is

$$\text{pTE}_{y \to x} = h(x_t | x_{t-1}) - h(x_t | x_{t-1}, y_{t-1}) \tag{2.9}$$

where $h(\cdot|\cdot)$ is the conditional permutation entropy. Given the joint probability $p(x_y, y_t)$ the conditional permutation entropy is calculate as follow

$$h(x_t|y_t) = h(x_t, y_t) - h(y_t) \tag{2.10}$$

This value represents how much uncertainty the knowledge of the sequence x_t has, knowing the sequence y_t.

Since we want to understand the causal relationship between vehicles, we decide not to evaluate the transfer entropy of the whole trajectory but using only parts of the data. In the dynamical window approach, we account as useful data only those data that refer to the interaction between the vehicles, or in our case the two robots. In order to cut the data in smaller parts, we fix a dimension Δ of the window and we evaluate the pTE of the data $x_i \cdots x_{i+\Delta}$. With this process, we intend to estimate the instantaneous pTE. In fact at time t we can observe the Δ previous positions and with them compute the pTE at that time, if its value is different from zero it means that there was a passage of information in the previous time steps.

Chapter 3

Simulations and Results

In this chapter, we firstly present the simulation environment and, then, all the studies we made and the results we obtained. In the first part, we describe the two tools for the simulations, ROS and Gazebo. After, we compare the symbols, how they perform and which results they give in terms of transfer entropy. In the third part, we made a parametric study of the parameters that characterize the best symbols, derived from the previous section. Finally, we give a summary of the results, along with a discussion of what we obtained.

3.1 Simulation Environment

We perform the simulations, made to validate the proposed approach, using two software: ROS and Gazebo. In this section, we will describe them without entering too much in technical details.

3.1.1 ROS: Robot Operating System

The Robot Operating System (ROS) is an open-source meta-operating system for robot software development [22, 23]. It provides hardware abstraction, low-level device control, message-passing between processes and package management [24]. At the moment, ROS is the most popular robotic framework, and the reason for its popularity has to be reached in its philosophical goals.
First of all, it is based on a peer-to-peer network. In this network each ROS process is called *node*, and it communicates with the other nodes thanks to two communication models: *topics* or *services*. The former is based on the model in which each node can be publisher or subscriber of a series of topics, the latter is used for synchronous transactions, i.e. a directed communications between nodes. The second philosophical goal is lingual independence. Indeed, the ROS code

can be implemented with different coding languages, such as C++, Python, Lips. Finally, it is free and open-source, this means that the full source code is publicly available. Moreover, it is tool-based, i.e. every system built on ROS is made of multiple components. In particular, each node is part of a package, and each package is part of a stack. The navigation stack, one of those stack, will be discussed later in this section.

3.1.2 Gazebo simulator

Gazebo is a robot simulator, it offers the possibility of simulate populations of robots in complex indoor and outdoor environments [25]. It has some interesting feature, such as dynamic simulations of physic engines, robot models, high-quality graphics even in three dimensions, it could be used on the cloud and it is free.
For our work, it is important to underline the possibility to use ROS in a Gazebo simulation. In Figure 3.1 the ROS's packages used by Gazebo are presented. They provide the necessary interfaces to simulate a robot in Gazebo using ROS messages and services.

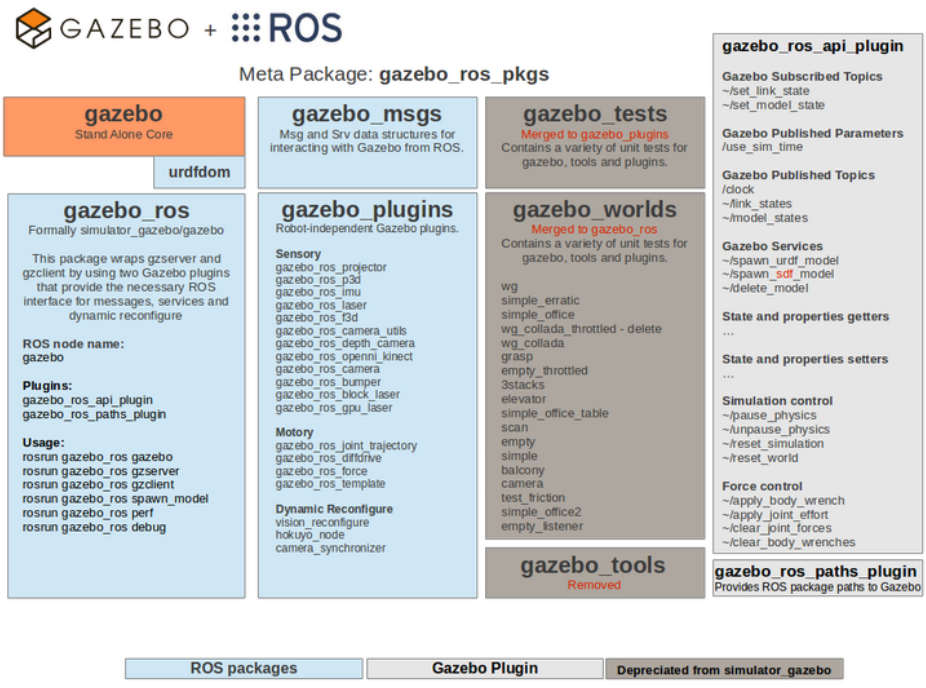


Figure 3.1: ROS packages for the Gazebo interface.

3.1.3 ROS navigation stack

One of the most important stack in ROS is the navigation stack that provides autonomous navigation task to a robot. The 2D navigation stack takes in information from odometry and sensor streams. Given a goal pose, the Navigation Stack determines safe velocity commands that are sent to a mobile base [26] providing the motion from the current robot position and the desired goal pose. It means that an autonomous robot can be modeled and simulated. Nevertheless, in order to move correctly we have to find the answers to three main questions: where is the robot?, where is it going?, and how does it get there?. The solution is shown in Figure 3.2. Starting from the left, the sensors permit to sense of the environment.

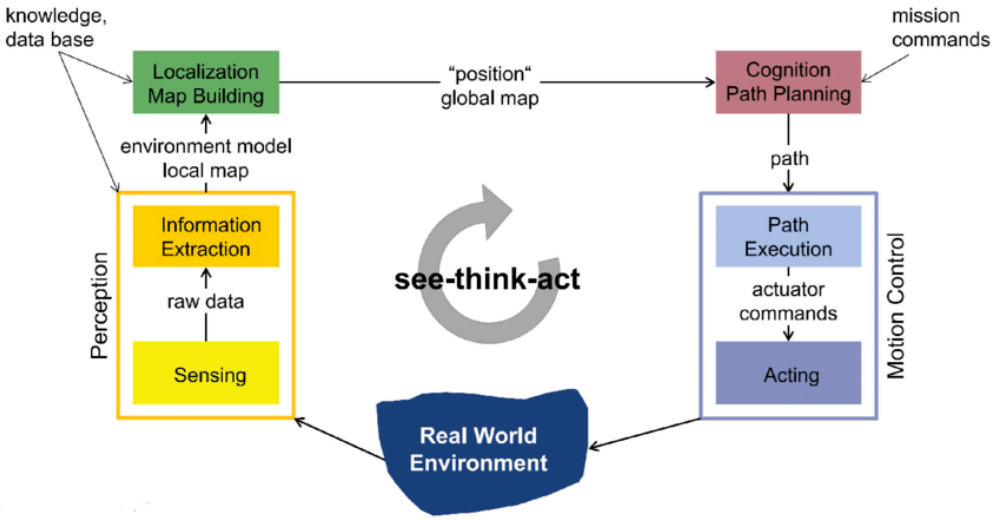


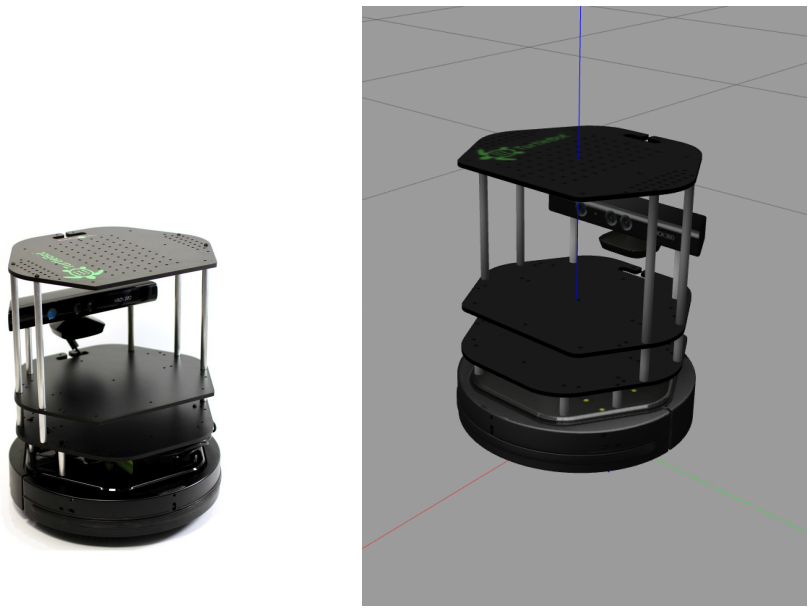
Figure 3.2: The see-think-act cycle. Image from [27]

Sensors data are used to create the map of the environment with the so-called SLAM (Simultaneous Localization and Mapping) algorithms or, if the map is already created, provide the localization task, i.e. the estimation of the robot pose in the map. Knowing the position on the global map and given the goal position, the path planning searches for the path to reach the target avoiding obstacles detected by sensors. This path is executed by a motion control algorithm that computes the control commands to the actuators and, as a consequence, moves the robot. This cycle, called see-think-act, continues until the robot reaches the goal position. This stack allows the simulation of the autonomous robots in a predefined environment.

3.1.4 Experimental simulations

We perform a simple simulation in which two robots navigate in the same environment interacting with each other. The first robot is an unmanned one, controlled by an autonomous navigation algorithm provided by the ROS Navigation Stack. The other robot is manned, i.e. it is moved by a human using a teleoperation ROS node, using the arrow keys on the keyboard. Both robots are Turtlebot 2 robots. Turtlebot 2 is a mobile robot with differential drive equipped with several sensors, such as a Microsoft Kinect depth camera, IMU (Inertial Measurement Unit), wheel encoders, etc. In Figure 3.3 both the real and the simulate Turtlebot 2 are shown.

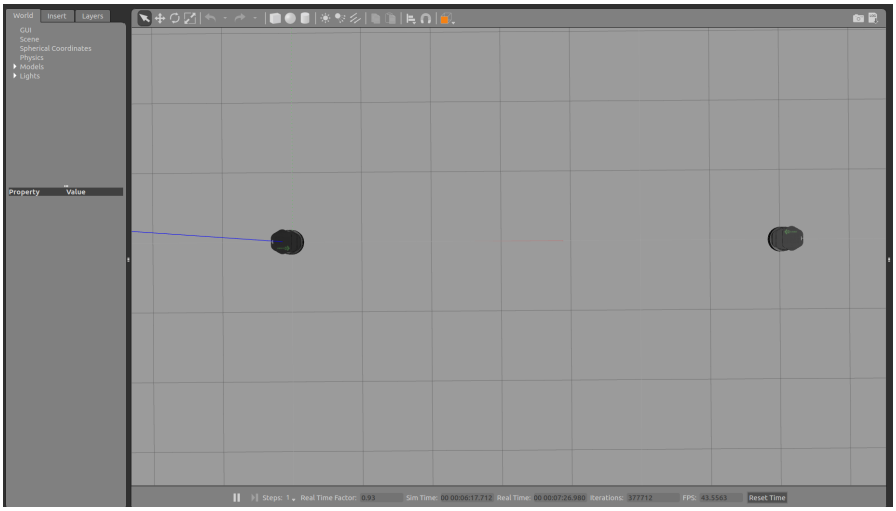
For the simulation, we use the tools described in the previous sections. In particular, both the Gazebo and ROS have a graphical user interface that allows to follow the simulation directly from the screen. In Figure 3.4(a) is illustrated a screen of the simulation in Gazebo, while in Figure 3.4(b) the same scenario is reported by the RViz tool, a ROS topic visualizer, that allows to show topic data in a graphical interface. Moreover, this is the starting position for the simulation we will describe in the following lines.



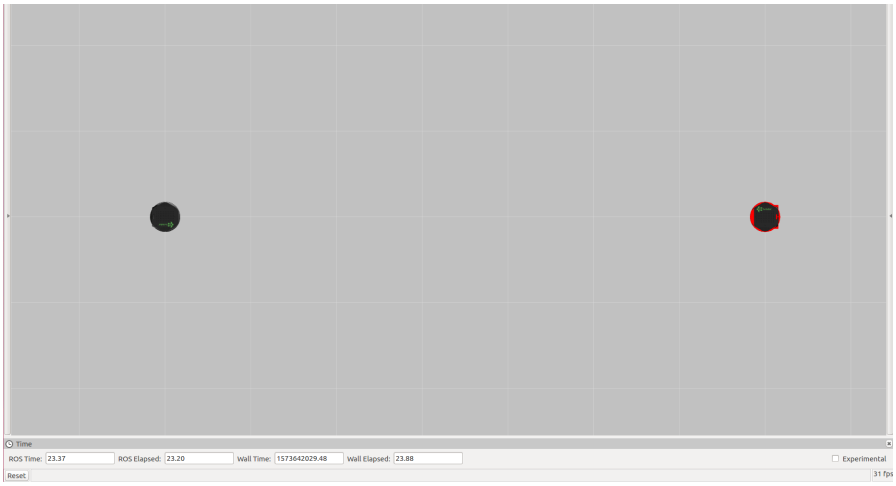
(a) Real Turtlebot 2. (b) Simulated Turtlebot 2.

Figure 3.3: In (a) the real Turtlebot 2 robot, while in (b) the simulated robot in the Gazebo simulator.

Many simulations are performed, but the most interesting one is the test where the two robots move with opposite trajectories and they face each other. To do this, we fixed the goal position of the robot to be coincident with the start position



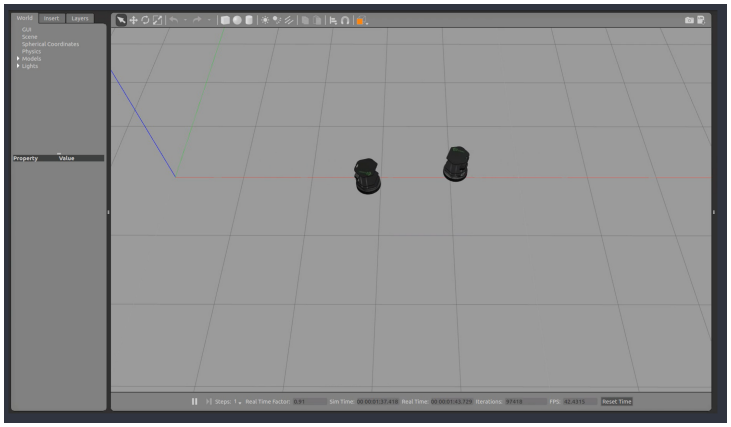
(a) Screen of the simulation in Gazebo.



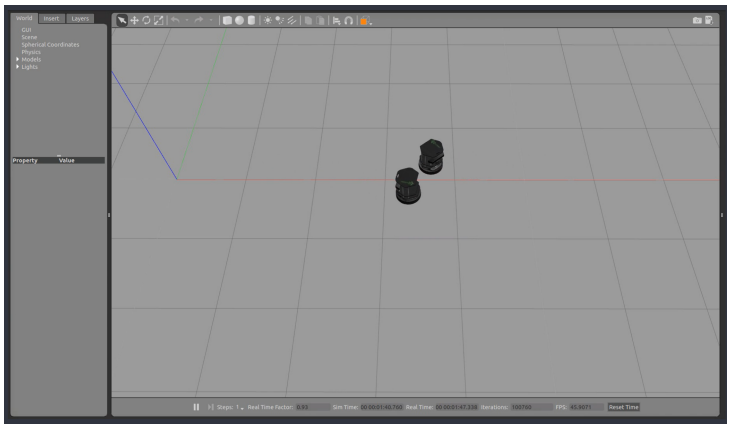
(b) Screen of the simulation in ROS.

Figure 3.4: Comparison between the two simulator environment.

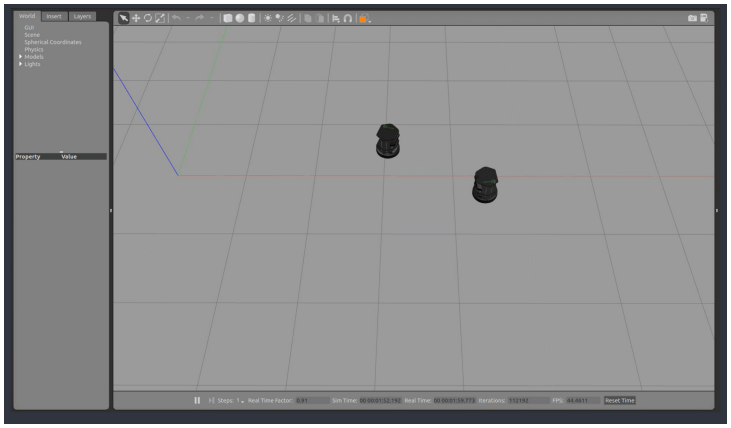
of the opposite robot. The ideal path is the line that links the two starting points, for both robots. During the simulation, they start to modify their path in order to avoid the other robot. Until they met and avoid each other, as illustrated in Figure 3.5a and Figure 3.5b. Finally, they can proceed without any obstacles until they reach the goal positions, as in Figure 3.5c.



(a) Experimental simulation: approach.



(b) Experimental simulation: encounter.



(c) Experimental simulation: departure.

Figure 3.5: Experimental simulation.

3.2 Different symbol approaches

In order to compare the results, we made several simulations, many with the same scenario. For every simulation, we collect the data as the position (x, y) of each vehicle at discrete time steps with a fixed frequency of 4 Hz. The frequency is chosen taking into account the variability and dynamic of the setup. In our case, the two robot have a maximal velocity of 0.2 m/s. A lower frequency could cause the loose of too many information, while a higher frequency would add noise to the data. Anyway, having such a small distance between two consecutive position allows us to approximate the trajectory between two consecutive points with a line without losing information. We apply the symbolic study and then we compute the transfer entropy as presented in chapter 2. We underline the value of parameters. The width of the window for the evaluation of the pTE is $\Delta = 30$, in *Symbol 1* we fix $m = 4$, for *Symbol 2, 3, 4* and *6* we fix $\epsilon_{dist} = 0.2$, $\epsilon_{dir} = \pi/6$, $\epsilon_{vel} = 0.2$, finally for *Symbol 5* we set the dimension of the bands to $d = 0.05$. Moreover we want to specify that the first and last values are not interesting and useful, in fact, it can happen that one of the vehicles already stopped and the other one didn't. All the values are fixed after an accurate empirical work.

The first scenario is the simplest one: two vehicles proceed with parallel trajectories, as in Figure 3.6. In blue the trajectory traveled by the unmanned vehicle and in red the trajectory of the manned vehicle. In this case, we expect to see no relationship between the two vehicles. They move independently and they do not change any information. In fact, the task for each vehicle is to reach the goal and it is not to maintain the same distance from the other vehicle, in which case we expect the transfer entropy to be different from zero. In Figure 3.7 and 3.8 we can observe the permutation transfer entropy [pTE], in blue the pTE from the manned vehicle to the unmanned one [$\text{pTE}_{m\to u}$] and in violet the pTE from the unmanned vehicle to the manned one [$\text{pTE}_{u\to m}$]. As we expected all the values are zeros, except for the first two symbols. The reason is the way in which the symbols are evaluated. Indeed the *Symbol 1* will have a different n-tuple even for small displacements from the ideal trajectory, while the *Symbol 2* does not refer to the ideal trajectory and even a small change in the angle will be noticed.

The second scenario, in Figure 3.9, represents the manned vehicle traveling on its route without changing it, and the unmanned vehicle turns in order to avoid the manned vehicle. As before the red line represents the manned vehicle, and it starts to the left and it goes to the right direction, while the blue line represents the route of the unmanned vehicle, which starts in the right position and it travels through the left direction. In order to better understand the results, we plot the data differently. In Figure 3.10 the x-axis represents the time steps instead of the

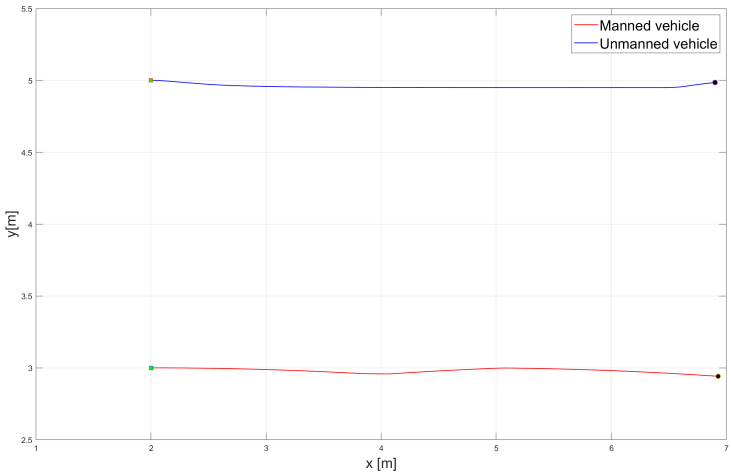


Figure 3.6: The axes are the x and y coordinates and they are measured in meters. In blue the unmanned vehicle, it starts to the left and it goes in the right direction; in red the manned vehicle, as the unmanned one it starts on the left and it routes towards the right direction. For both the robots the starting point is the green square and the goal is the black circle.

space, using this type of representation we are able to select the exact time step in which the unmanned changes its position and, subsequently, we can observe if there are any differences in the value of the entropy. Having the same parameters used in the previous case, we analyze the results. Figure 3.11 and 3.12 shows the pTE, in blue the pTE from the manned to the unmanned vehicle, in violet on the contrary the pTE from the unmanned to the manned vehicle. All the parameters are fixed as before. We observe that the *Symbol 1* reports, as we expect, a positive value in the $\text{pTE}_{m\to u}$ around $t = 55$. In fact, if we focus on Figure 3.10, the turn in the unmanned trajectory is at time $t \sim 55$. The other positive values appear when the two vehicles already pass each other, then we can conclude that they do not give us any information. The *Symbols 2, 3* and *6* give similar results with the strong presence of a positive component of the $\text{pTE}_{u\to m}$. We explain this value observing the velocity of the unmanned vehicle. In fact, the velocity of the unmanned vehicle changes much more than the velocity of the manned one, and those three symbols are the only one which takes in account the velocity. Moreover, we can observe that after $t \sim 80 - 100$ the pTE become zero in both the directions. We expect this result observing the trajectory. Indeed they both proceed in opposite directions without causing the movement of the other.

The third scenario is similar to the second one, but, in this case, the manned

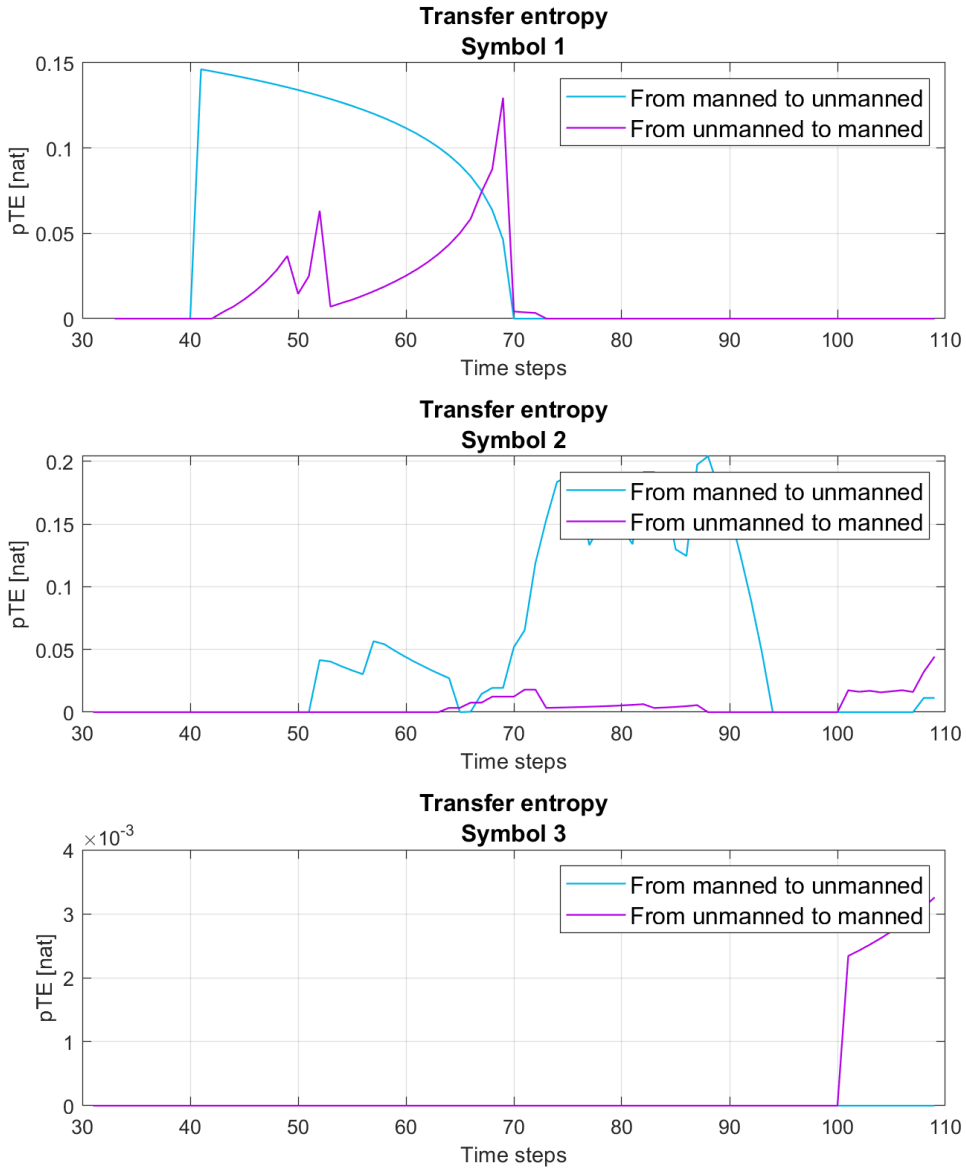


Figure 3.7: The permutation transfer entropy [pTE] for the parallel scenario, symbols from 1 to 3. In blue the pTE from the manned to the unmanned vehicle, in violet the pTE form the unmanned to the manned vehicle. The parameters are $\Delta = 30,\ m = 4,\ \epsilon_{dist} = 0.2,\ \epsilon_{dir} = \pi/6,\ \epsilon_{vel} = 0.2,\ d = 0.05$.

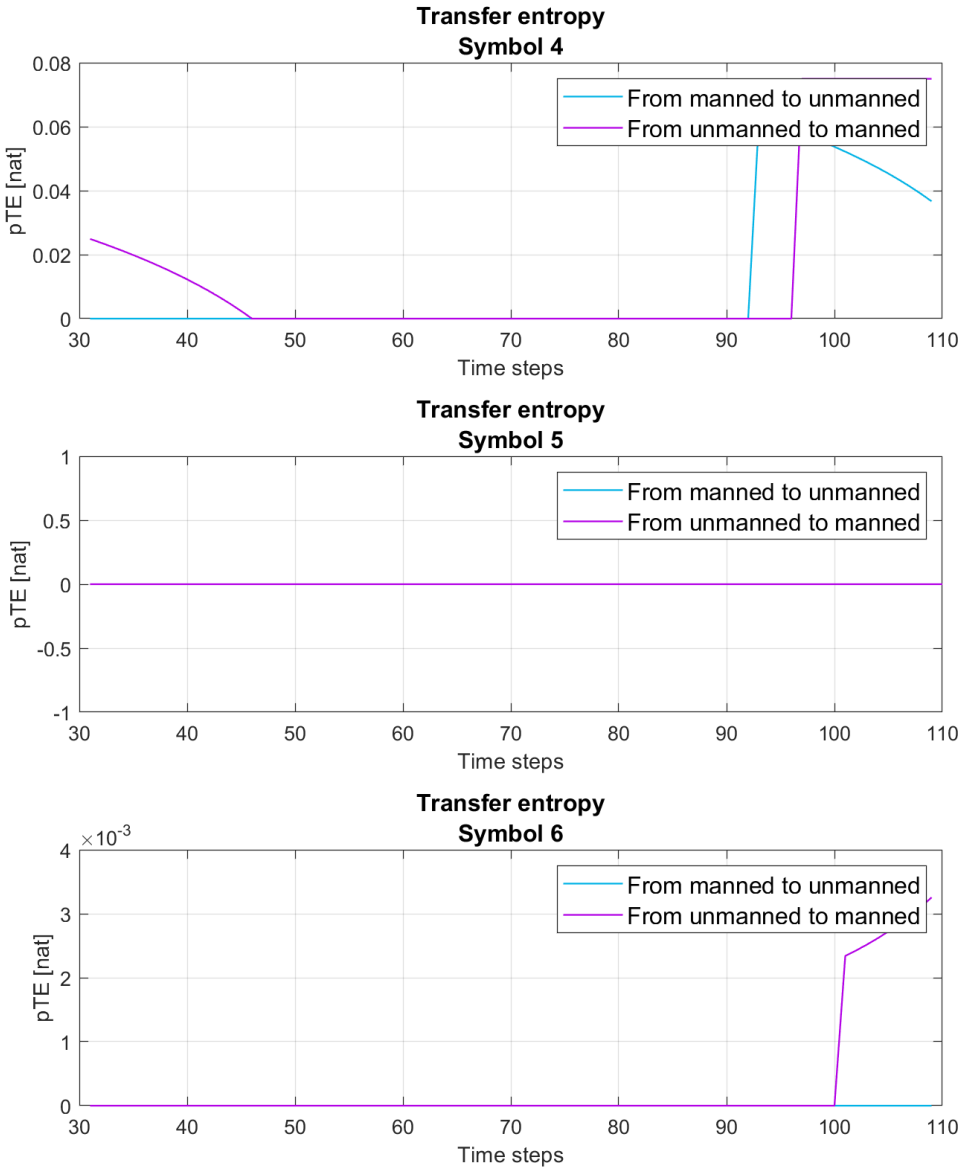


Figure 3.8: The permutation transfer entropy [pTE] for the parallel scenario, symbols from 4 to 6. In blue the pTE from the manned to the unmanned vehicle, in violet the pTE form the unmanned to the manned vehicle. The parameters are $\Delta = 30,\ m = 4,\ \epsilon_{dist} = 0.2,\ \epsilon_{dir} = \pi/6,\ \epsilon_{vel} = 0.2,\ d = 0.05$.

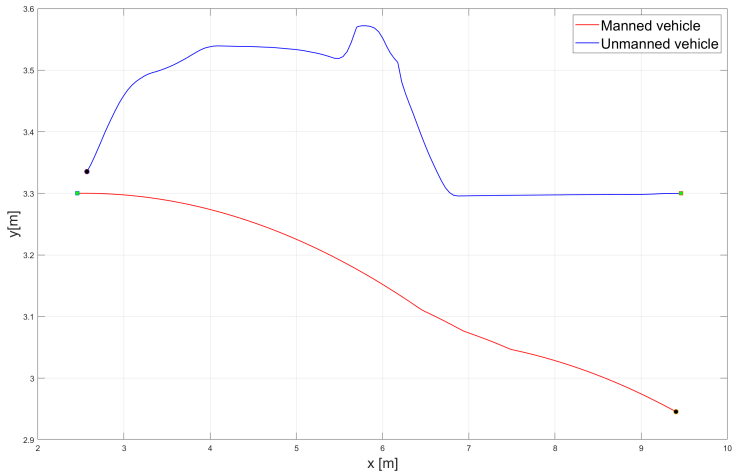


Figure 3.9: The axes are the x and y coordinates and they are measured in meters. In blue the unmanned vehicle, it starts to the right and it goes in the left direction; in red the manned vehicle, opposite to the unmanned one it starts on the left and it routes towards the right direction. For both the robots the starting point is the green square and the goal is the black circle.

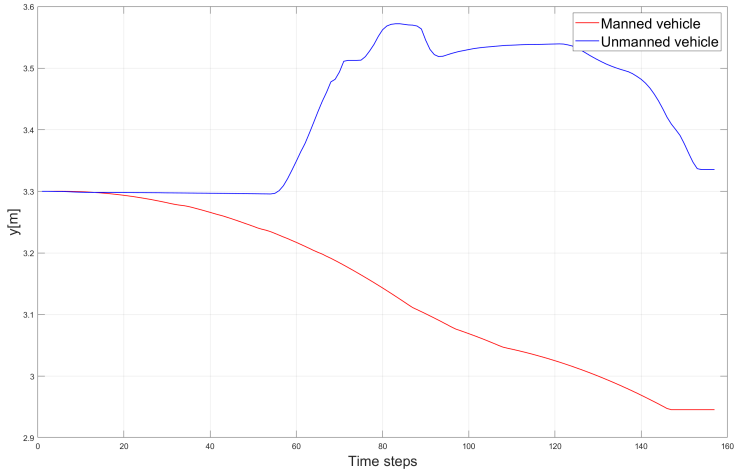


Figure 3.10: The x-axis is the time and the y-axis is the value of the y-coordinate measured in meters as before. In blue the unmanned vehicle; in red the manned vehicle.

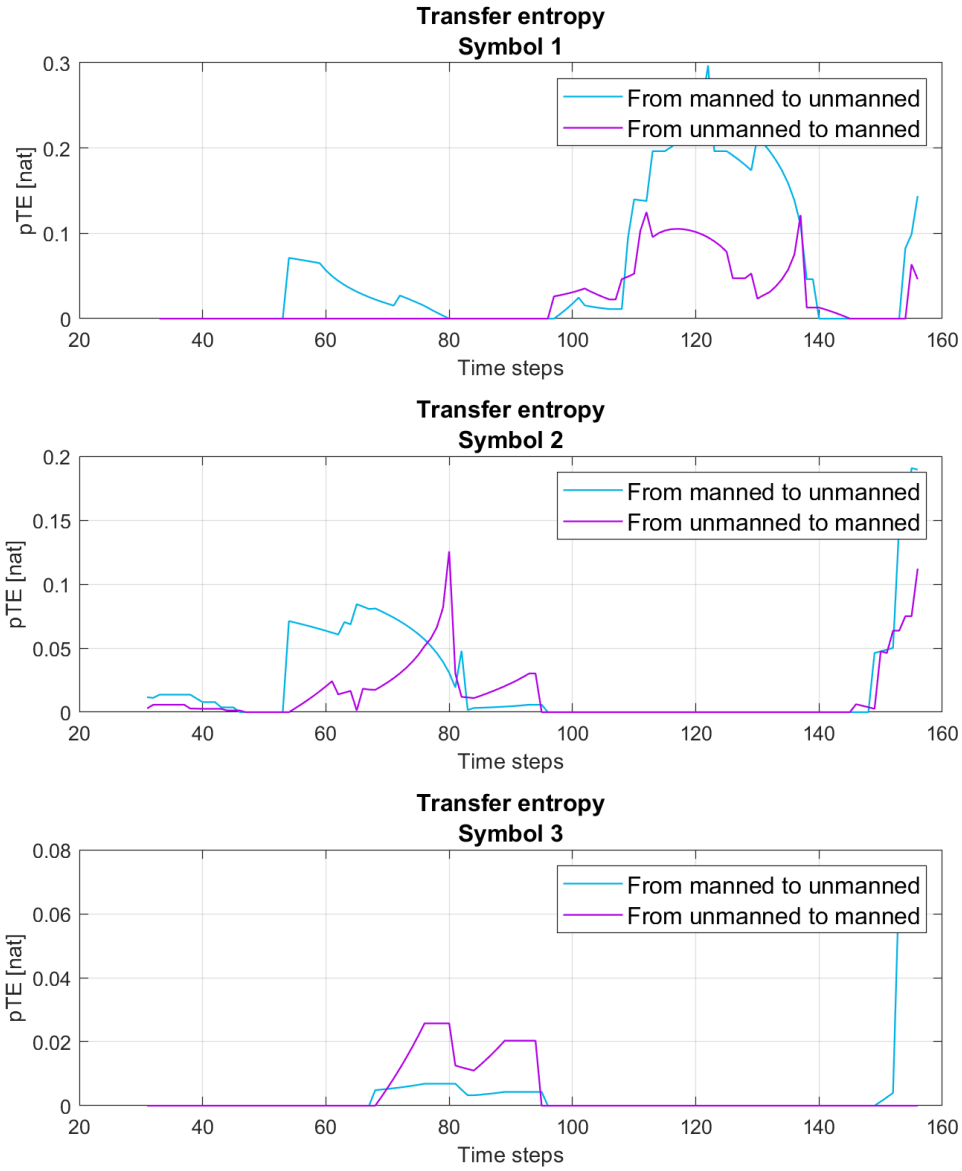


Figure 3.11: The permutation transfer entropy [pTE] for the second scenario, symbols from 1 to 3. In blue the pTE from the manned to the unmanned vehicle, in violet the pTE form the unmanned to the manned vehicle. The parameters are $\Delta = 30,\ m = 4,\ \epsilon_{dist} = 0.2,\ \epsilon_{dir} = \pi/6,\ \epsilon_{vel} = 0.2,\ d = 0.05$.

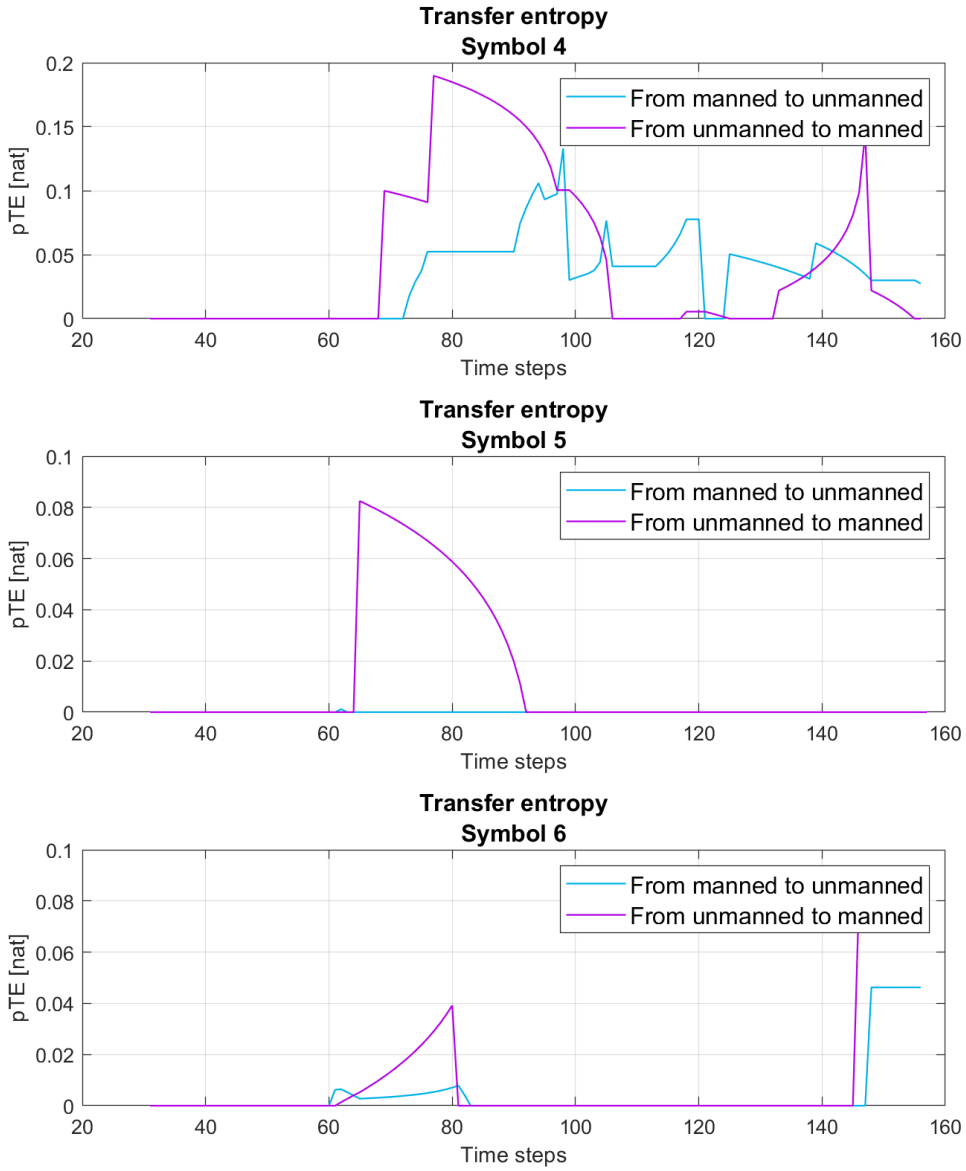


Figure 3.12: The permutation transfer entropy [pTE] for the second scenario, symbols from 4 to 6. In blue the pTE from the manned to the unmanned vehicle, in violet the pTE form the unmanned to the manned vehicle. The parameters are $\Delta = 30,\ m = 4,\ \epsilon_{dist} = 0.2,\ \epsilon_{dir} = \pi/6,\ \epsilon_{vel} = 0.2,\ d = 0.05$.

vehicle tries to avoid the unmanned while the unmanned try to avoid the manned one. We present the data both with the x- and y-axis as the coordinates in meters, Figure 3.13, and with the x-axis that represents the time, Figure 3.14. We can

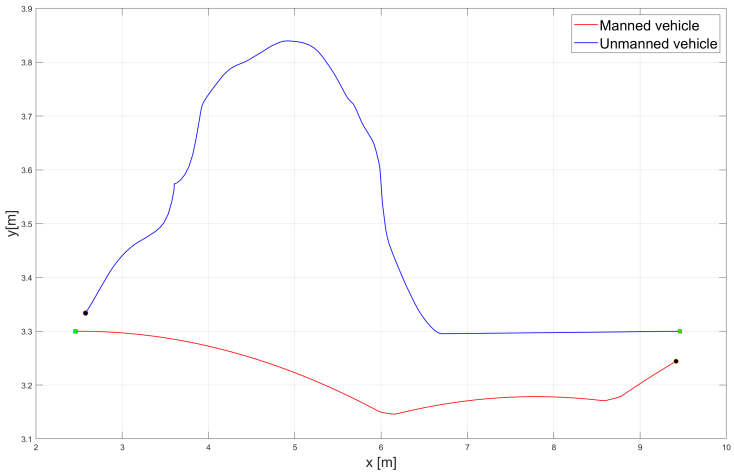


Figure 3.13: The axes are the x and y coordinates and they are measured in meters. In blue the unmanned vehicle, it starts to the right and it goes in the left direction; in red the manned vehicle, opposite to the unmanned one it starts on the left and it routes towards the right direction. For both the robots the starting point is the green square and the goal is the black circle.

notice, in Figure 3.15 and 3.16, that using $\Delta = 30$ the *Symbol 1* is not able to detect the flow of information at time $t = 60$. On the contrary, all the other symbols detect the flow of information when the unmanned turns. We also see that the value of the pTE is higher than before, we expect it to be higher because the trajectory deviates more from the ideal trajectory. Focusing on who causes who, we notice that in the last four symbols the $\text{pTE}_{m\to u}$ is the first one which changes its value from zero to a positive one. Given many other simulations, we can infer that this means that the manned vehicle causes the turn in the unmanned trajectory. The following higher value in the $\text{pTE}_{u\to m}$ is most probably due to the fact that the unmanned trajectory is more irregular and it is more difficult to explain it starting from what we know of the manned trajectory. Or, in other words, the irregularity in the unmanned trajectory sends more information to the manned vehicle rather than the number of information that the manned sent to the unmanned one. For this reason, is easier to explain the movement of the manned knowing the unmanned than otherwise.

Finally, we present the same study with a new scenario. In Figure 3.17 the

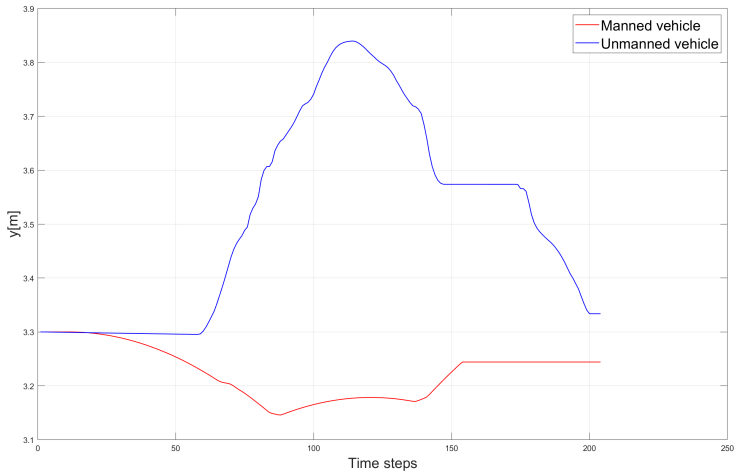


Figure 3.14: The x-axis is the time and the y-axis is the value of the y-coordinate measured in meters as before. In blue the unmanned vehicle; in red the manned vehicle.

unmanned and the manned vehicles intersect their routes. The red line, corresponding to the manned vehicle, start from left and travel towards the right direction, the blue line, corresponding to the unmanned vehicle, do the same, but it has its goal on the opposite side. The two vehicles are obliged to cross the path of the other. This simulation appears more irregular and we expect a higher value of the pTE. Even in this case, we prefer to plot the data along the time axis in order to better understand the pTE values. We now focus on Figure 3.19 and 3.20, the parameters remain the same as the previous simulations. We notice the big difference between a symbol an another. In fact, the *Symbol 1* almost does not detect any flow of information, the value of the pTE is very low. The most reasonable explanation is that the number of times in which the vehicle moves away or approaches the ideal trajectory is low and the position is almost the same during all the trajectory. With the other symbols, it is relatively difficult to say which vehicle causes the other, they both change information with the other, but probably there is not one who obliges the other to change its route. Only using *Symbol 3* we can actually observe a major flow of information from manned to unmanned vehicle.

Using the pTE we are able to detect when a movement is caused by the other vehicle and when it is not. In fact in the second and third cases when the vehicle makes the turn the $\text{pTE}_{m \to u}$ has a peak, with almost every symbol. Another important consideration is that after a certain point, in particular, the point in

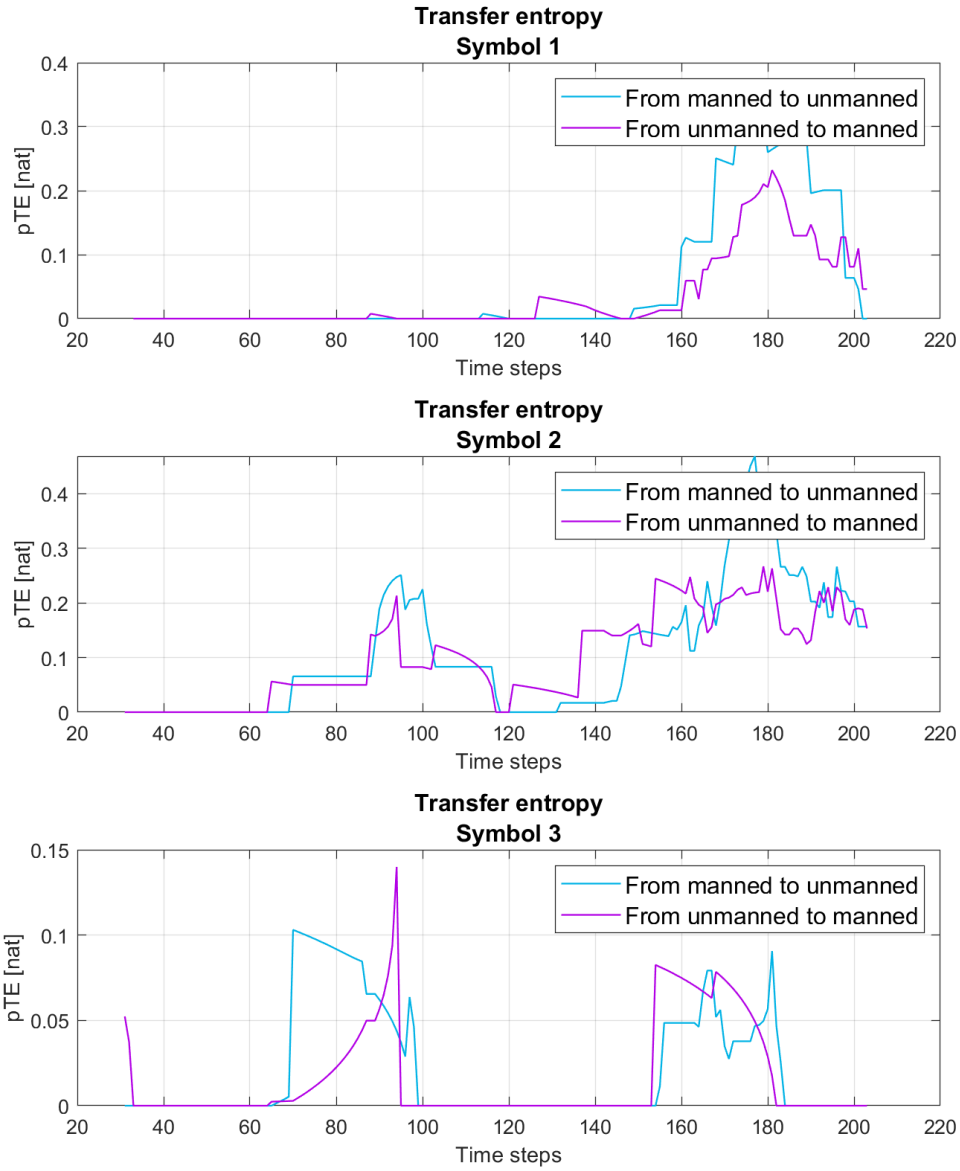


Figure 3.15: The permutation transfer entropy [pTE] for the third scenario, symbols from 1 to 3. In blue the pTE from the manned to the unmanned vehicle, in violet the pTE form the unmanned to the manned vehicle. The parameters are $\Delta = 30,\ m = 4,\ \epsilon_{dist} = 0.2,\ \epsilon_{dir} = \pi/6,\ \epsilon_{vel} = 0.2,\ d = 0.05$.

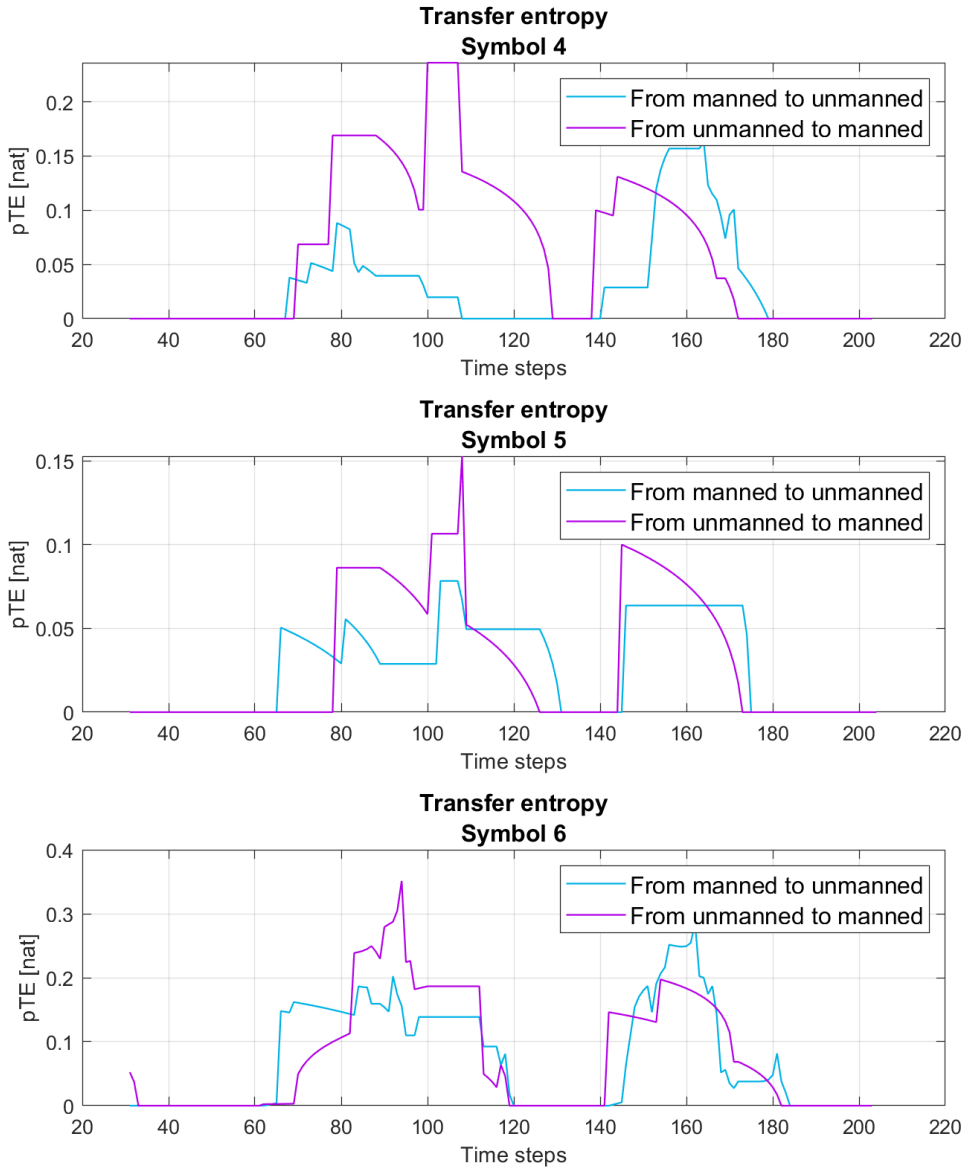


Figure 3.16: The permutation transfer entropy [pTE] for the third scenario, symbols from 4 to 6. In blue the pTE from the manned to the unmanned vehicle, in violet the pTE form the unmanned to the manned vehicle. The parameters are $\Delta = 30,\ m = 4,\ \epsilon_{dist} = 0.2,\ \epsilon_{dir} = \pi/6,\ \epsilon_{vel} = 0.2,\ d = 0.05$.

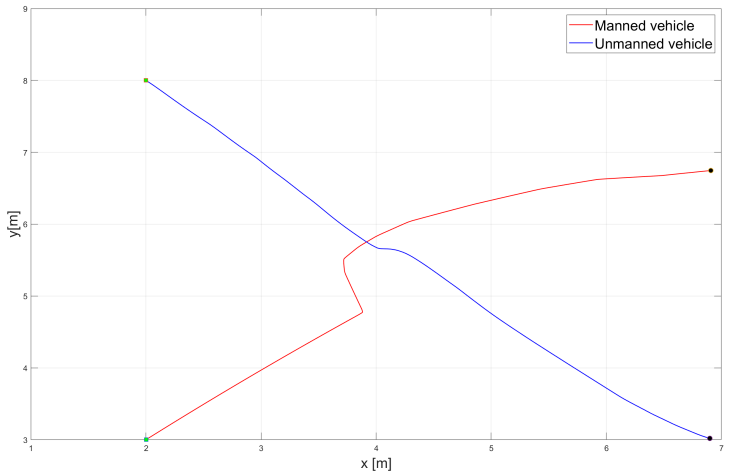


Figure 3.17: The axes are the x and y coordinates and they are measured in meters. In blue the unmanned vehicle, it starts to the left and it goes in the right direction; in red the manned vehicle, as the unmanned one it starts on the left and it routes towards the right direction. For both the robots the starting point is the green square and the goal is the black circle.

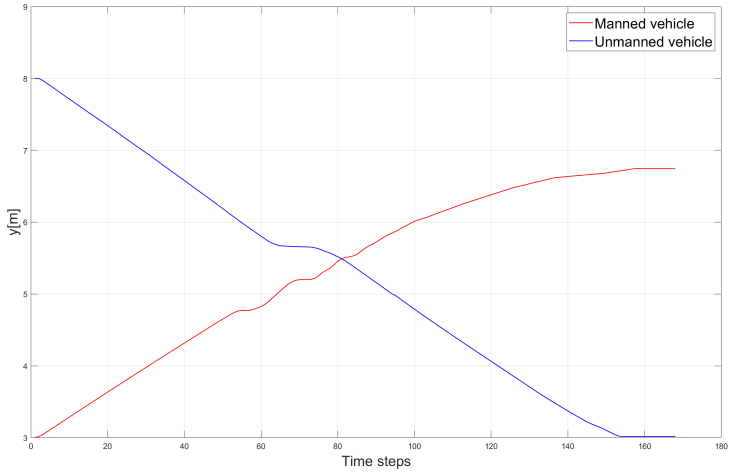


Figure 3.18: The x-axis is the time and the y-axis is the value of the y-coordinate measured in meters as before. In blue the unmanned vehicle; in red the manned vehicle.

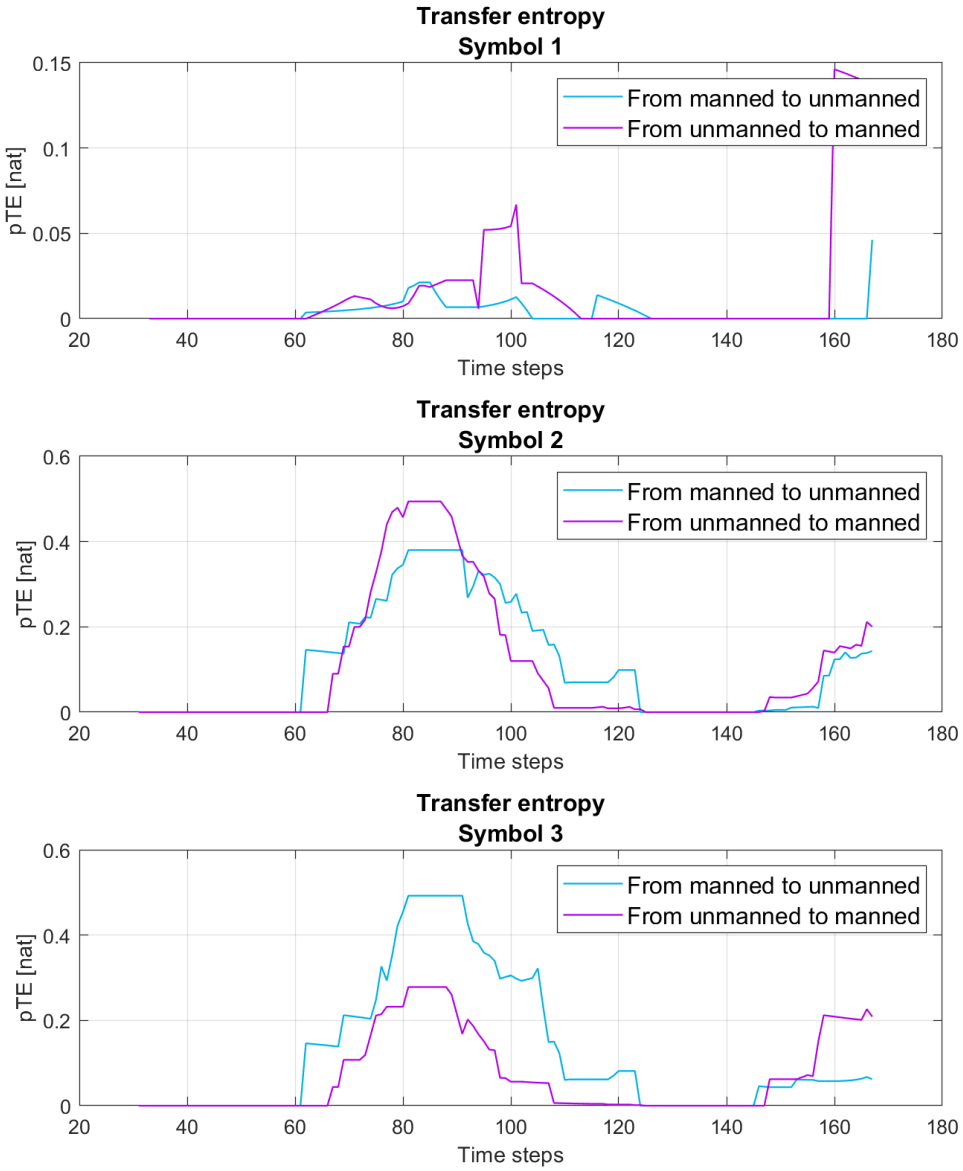


Figure 3.19: The permutation transfer entropy [pTE] for the fourth scenario, symbols from 1 to 3. In blue the pTE from the manned to the unmanned vehicle, in violet the pTE form the unmanned to the manned vehicle. The parameters are $\Delta = 30,\ m = 4,\ \epsilon_{dist} = 0.2,\ \epsilon_{dir} = \pi/6,\ \epsilon_{vel} = 0.2,\ d = 0.05$.

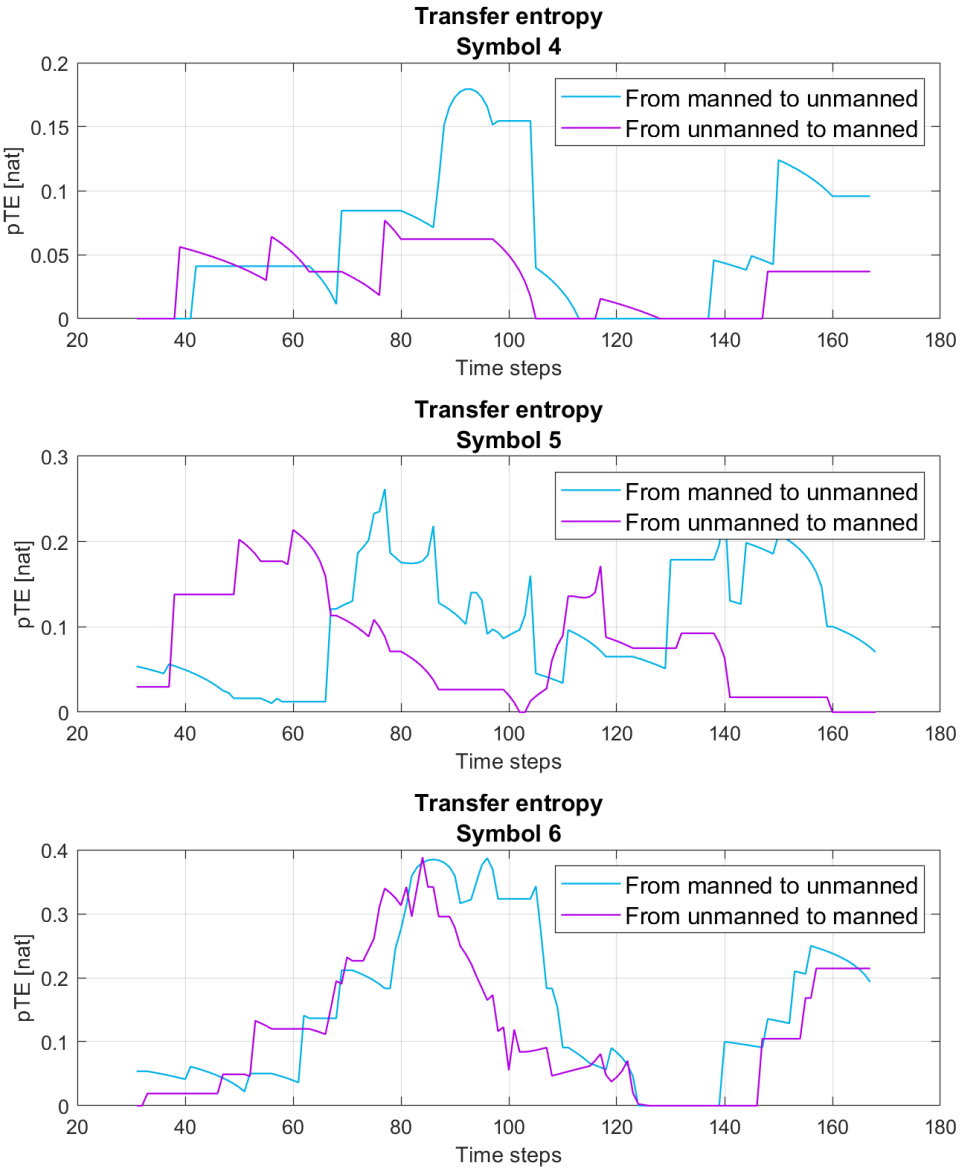


Figure 3.20: The permutation transfer entropy [pTE] for the fourth scenario, symbols from 4 to 6. In blue the pTE from the manned to the unmanned vehicle, in violet the pTE form the unmanned to the manned vehicle. The parameters are $\Delta = 30,\ m = 4,\ \epsilon_{dist} = 0.2,\ \epsilon_{dir} = \pi/6,\ \epsilon_{vel} = 0.2,\ d = 0.05$.

which they surpass each other, the value of pTE loses its importance. In many cases it goes to zero as expected, in other cases due to changes in the trajectory it remains positive, but without any connotation. We also observe that the most significant symbols are the *Symbol 1* and the *Symbol 6*, that is why we choose those two symbol for the following study, i.e. the parametric analysis. Studying more deeply the performance of the symbols we realize that a bigger width of the transfer entropy window is preferable, so in the following, we fix it to $\Delta = 40$.

3.3 Parametric study

In this section, we will analyze the trend of the pTE changing some parameters. We will use only the *Symbol 1* and the *Symbol 6* apply to the second case of study, see Figure 3.9. We also fix the width of the pTE window, $\Delta = 40$.

For the *Symbol 1* the only parameters which can vary is m. In Figure 3.21 there are the results of four different values of m, i.e. $m = 4$, $m = 6$, $m = 10$, $m = 20$. First of all we notice that increasing the parameter m means to increase the number of total possible symbols. In fact, with this type of symbol the possible symbols are $m!$, i.e. the tuples containing all the permutation of the numbers $\{1, \dots, m\}$. The increasing number of symbols make the trajectory more irregular in its symbolization and the value of the transfer entropy increases. In fact, for simplicity we already display in Figure 3.21 the values of pTE at the same point, showing that this value is increasing with m, that is with the number of symbols. After this consideration, with the help of Figure 3.22, we observe that the moment in which the pTE from manned to unmanned has its first peak shifts. For example, the point at $t = 55$ is the moment in which the unmanned has its turn, see Figure 3.22(b). Using the *Symbol 1* we detect the turn m time steps before it happens, but it is actually due to the way in which the symbol is evaluated. Indeed, each symbol takes into account m time steps. Now we focus on the second peak, the one in the pTE form unmanned to manned. We notice, thanks to Figure 3.22(a), that at this point $t = 98$ the two vehicles already surpass each other. It is very unlikely that the two robots have got a change of information at this point. We believe that the values from this point on are no longer significant to determine the causal relationship between the vehicles.

Using the *Symbol 6* we can study the effect of three parameters: ϵ_{dist}, ϵ_{vel}, ϵ_{dir}. Changing those parameters, as changing m in *Symbol 1*, means to change the number of total possible symbols. In fact, if ϵ_{dist} decreases it means that the number of subdivisions increases and so do the total symbols. For this type of symbols the total number of symbols is $\frac{1}{\epsilon_{dist}} \times \frac{1}{\epsilon_{vel}} \times \frac{\pi}{\epsilon_{dir}}$.

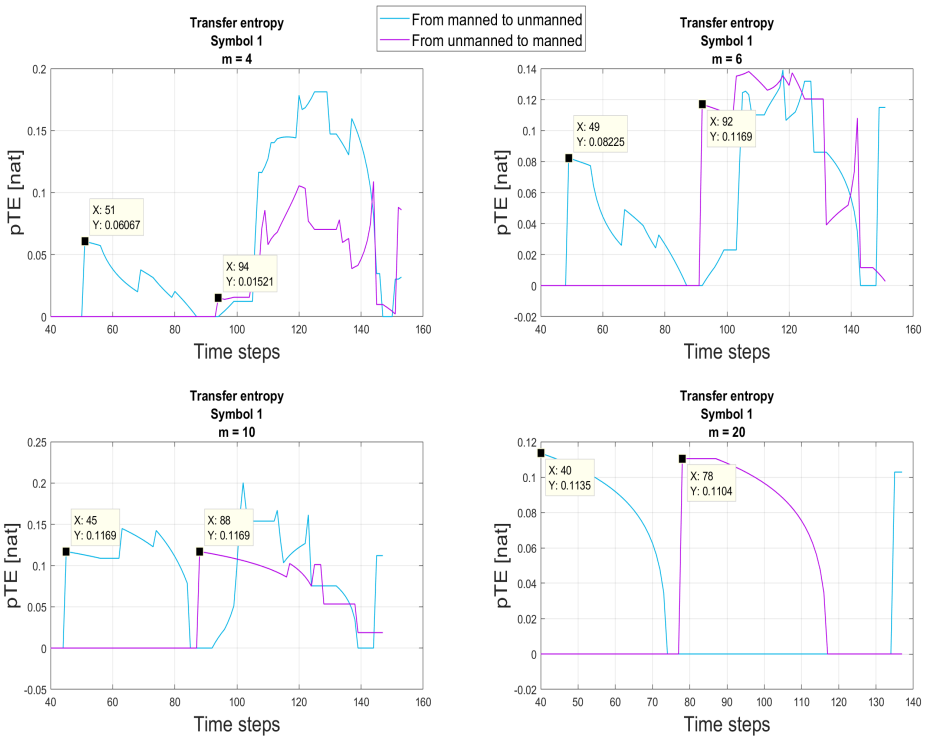
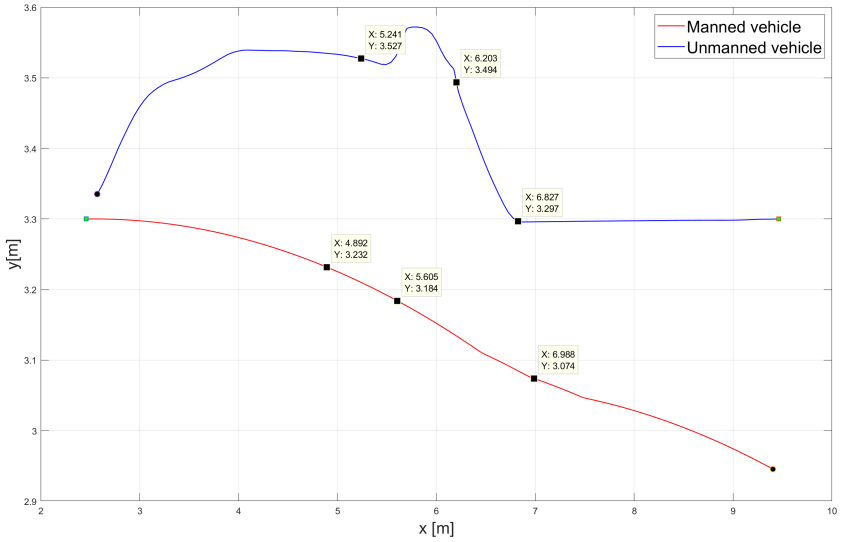
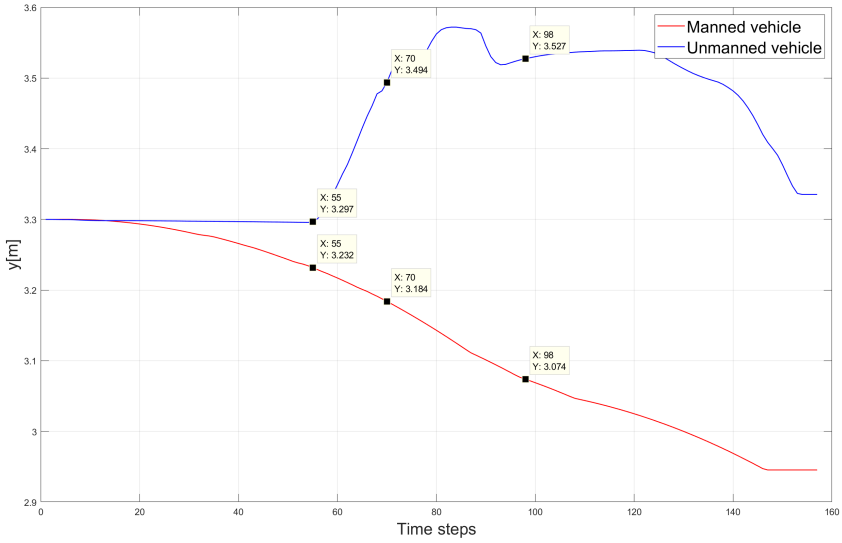


Figure 3.21: pTE varying the value of the parameter m. In blue the pTE from manned to unmanned, in violet the pTE from unmanned to manned.

We start focusing on the first parameters: ϵ_{dist}. In Figure 3.23 there are four different value of it, i.e. $\epsilon_{dist} = 0.5$, $\epsilon_{dist} = 0.2$, $\epsilon_{dist} = 0.1$, $\epsilon_{dist} = 0.01$, it means that we subdivide the interval of the distance from the ideal trajectory in two, five, ten or one hundred parts. As well as the *Symbol 1* the increasing in the number of total symbols corresponds to an increase in the value of the transfer entropy both from manned to unmanned and in the opposite direction. Nevertheless due to the fact that the manned robot differs little from the ideal trajectory the pTE value from manned to unmanned increases very little compared to the pTE from unmanned to manned. Moreover, there is a threshold over which the pTE does not have any significance. In Figure 3.23 the last plot is extremely confusing and it does not give us any information. We notice that even in this case the time in which the pTE from manned to unmanned has the first peak is detected more rapidly with a greater number of symbols. On the contrary the first peak in the pTE from unmanned to manned remains at time $t = 70$. From Figure 3.22 it is clear that the two robot do not surpass each other yet, so it is possible that the unmanned influences the manned one. This influence wasn't detected by the



(a)



(b)

Figure 3.22: The representation of the two trajectories using different reference systems. In blue the unmanned trajectory; in red the manned trajectory. Critical points are labeled for better understanding of the pTE results.

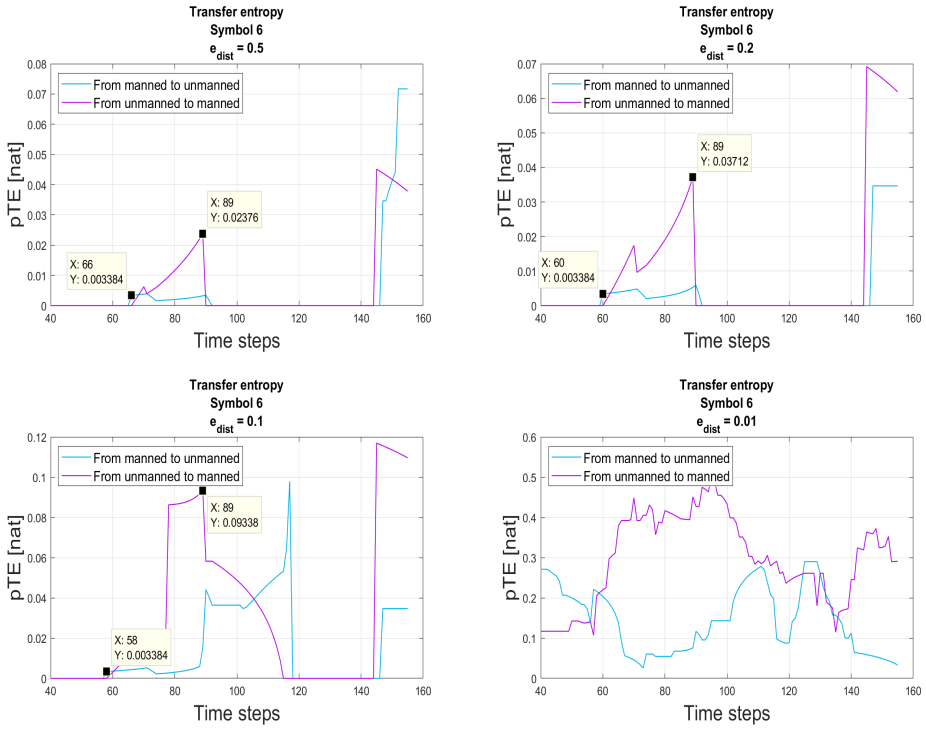


Figure 3.23: pTE varying the value of the parameter ϵ_{dist}. In blue the pTE from manned to unmanned, in violet the pTE from unmanned to manned.

Symbol 1.

The analysis of the second parameter is similar to the first one, we fix the ϵ_{vel} with four different values: $\epsilon_{vel} = 0.5$, $\epsilon_{vel} = 0.2$, $\epsilon_{vel} = 0.1$, $\epsilon_{vel} = 0.01$. Premise that in our simulations the velocity of the robots is quite constant and they both have the same maximum velocity it is clear why in Figure 3.24 the first plot it appears null in almost all the positions. In fact, if $\epsilon_{vel} = 0.5$ it means that the interval of the velocity is divided into two parts. The number of symbols is very limited and the trajectory is probably represented by the same symbol in all its parts. Increasing the divisions in the interval of velocity we observe the usual increase in the pTE values. However it is unlikely that the plot with $\epsilon_{vel} = 0.1$ in Figure 3.24 would be realistic. Comparing these results with the previous ones, it appears clear that the best value for ϵ_{vel} is 0.2.

Finally, we study the behavior of the pTE changing ϵ_{dir}. We notice that changing this parameter does not interfere with the solution, except for a large number of subdivisions, see Figure 3.25. The reason could be that they do not vary a lot. In fact, we can plot the angle. For the manned trajectory, the angle is represented

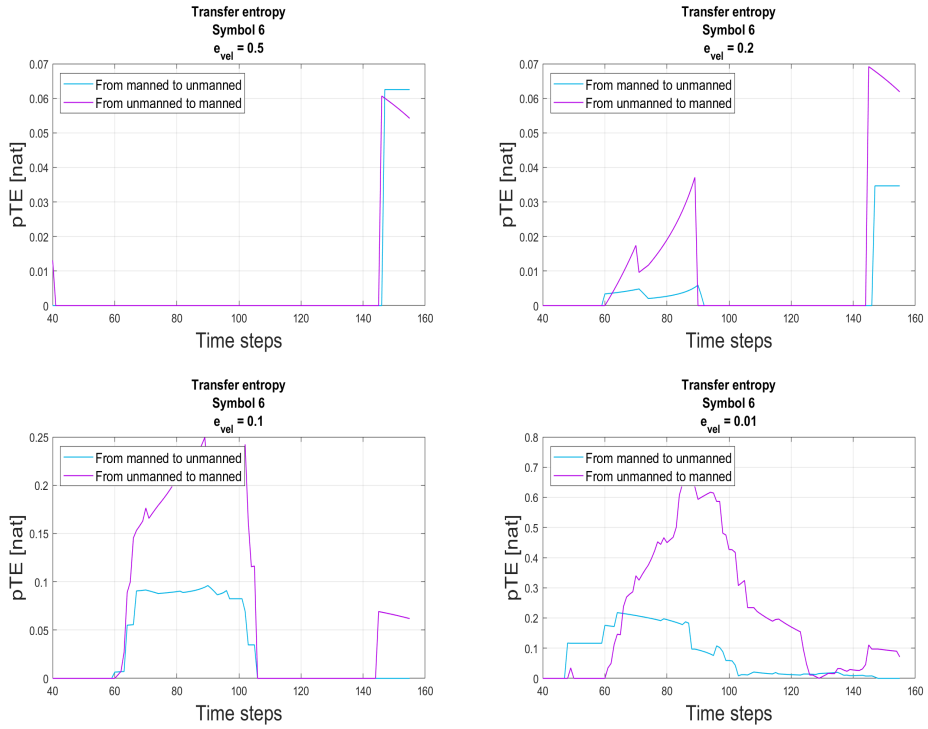


Figure 3.24: pTE varying the value of the parameter ϵ_{vel}. In blue the pTE from manned to unmanned, in violet the pTE from unmanned to manned.

in Figure 3.26, we notice that it does not vary and this is the reason why the parameter ϵ_{dir} it is not primary in this study. However, the angle of the unmanned trajectory vary much more than the manned one, see Figure 3.27. This condition is not sufficient to see a change in the pTE.

3.4 Discussion

The main question was: which robot causes the movement of the other? To find the answer we take into account four scenarios performing several simulations. In the first study, the one about the performance of the symbols, we are able to see that the majority of the time the manned robot is the cause of the unmanned. Indeed, in the second and third scenario the unmanned robot tries to avoid the manned one. As we expect, the major information flow is from manned to unmanned, and the symbols that better detect the causation are *Symbols 1* and *6*. In the third simulation, see Figure 3.13 and 3.14, at time $t = 59$ the transfer entropy reveals that the manned causes the unmanned. This causation is due to the

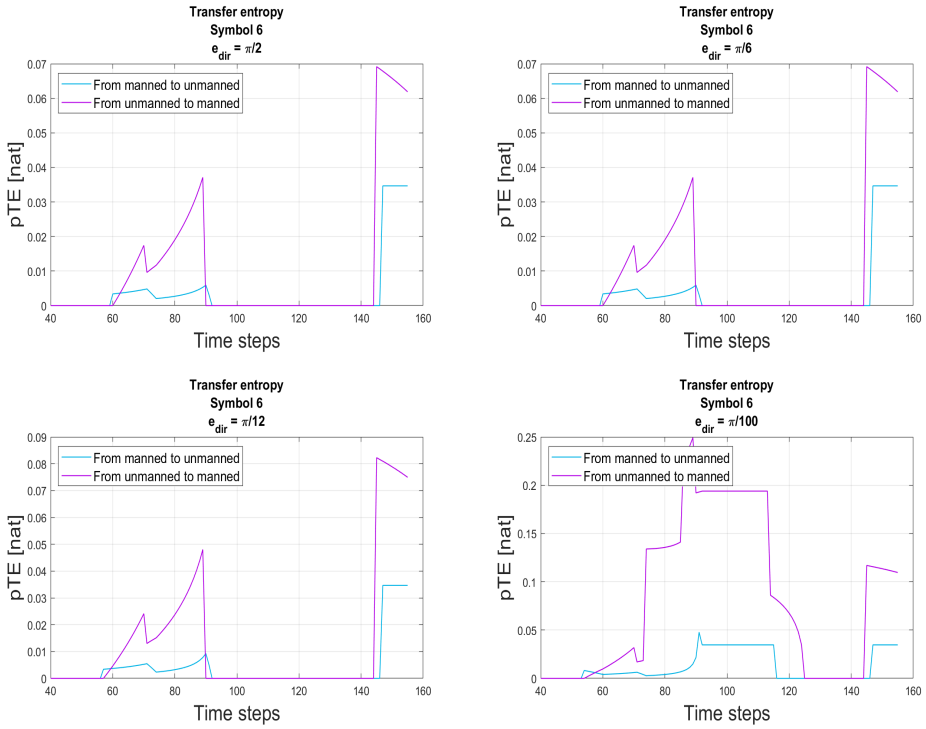


Figure 3.25: pTE varying the value of the parameter ϵ_{dir}. In blue the pTE from manned to unmanned, in violet the pTE from unmanned to manned.

fact that the manned vehicle proceeds on its route, while the autonomous robot, using the on-board sensors, perceives the presence of the obstacle and, then, the navigation algorithm tries to avoid it.
Moreover, the absolute value changes based on the trajectory. In particular, more irregular the trajectory is, higher is the information transferred to the other vehicle. This explains why in almost every plot the $\text{pTE}_{u\to m}$ value increases more than the $\text{pTE}_{m\to u}$ value. Indeed, in the second scenario, where the manned robot follows its trajectory without deviating from it, the higher values of pTE are reached by the flow of information from unmanned to manned.
As said before, usually the most irregular trajectory has the highest values of pTE. Nevertheless, in the second and third scenarios, the peak of pTE from manned to unmanned occurs when the unmanned vehicle changes its route, while the manned one proceeds on its trajectory. This situation, apparently an exception, is the proof that the manned vehicle causes the unmanned vehicle. In fact, the pTE detects the causality. The manned robot, persistent in following its route, forces the unmanned robot to change its path.

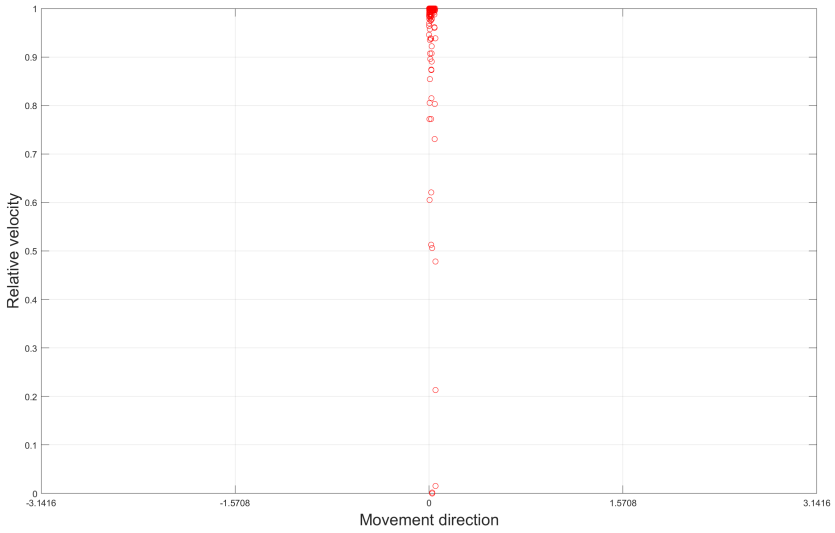


Figure 3.26: The x-axis is the *Movement Direction* it goes from $-\pi$ to π, the y-axis is the *Relative Velocity* and it varies in the interval [0,1]. In red the value of each data of the manned vehicle.

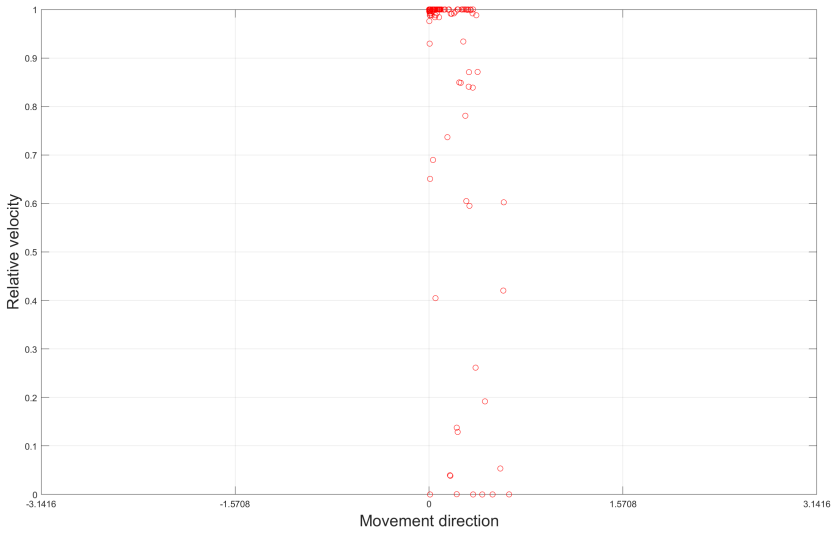


Figure 3.27: The x-axis is the *Movement Direction* it goes from $-\pi$ to π, the y-axis is the *Relative Velocity* and it varies in the interval [0,1]. In red the value of each data of the unmanned vehicle.

After many comparisons we observe that the first and sixth symbols perform better than the others. The first detects with precision the instant in which one robot causes the other. Anyway, it needs to evaluate more information, i.e. a bigger width of the window, which was fixed at $\Delta = 30$ for the first study. The sixth symbol always finds the robot that causes the other, even with less data. Nevertheless, in some cases, e.g. the fourth scenario, it is a bit confusing and not precise. We choose these two type of symbols for the second part of the study. As we expected, the values of pTE increase a lot when the number of total symbols of each type increases. In fact, having more symbols the space is splitted into more sets and the average probability decrease. The more variability in the symbols reflects a more irregular sequence and, as a consequence, a more unpredictable trajectory. Remembering that the entropy is a measure of the degree of uncertainty, it is clear why increasing the total number of symbols is a cause of the increase in the pTE value.

Chapter 4

Conclusions and future works

This thesis aims to study the causal relationship between vehicles. In order to achieve our purpose we use the transfer entropy method, a method already tested and validated in the literature but in other fields.
In particular, we perform our study simulating two robots, one autonomous and one teleoperated by a human. The two robots are involved in four different scenarios. In each scenario, each robot has a starting position and a goal position fixed on the map. During the simulations, the robot position is sampled at a rate of 4 Hz. Then, the data are reworked with the symbolic analysis in order to be suitable for the transfer entropy method. The results we obtained are in line with our prediction. In particular, the plots of transfer entropy have a positive jump when one robot causes the other.

Starting from this thesis, there are many possible developments. One development could be changing the method for evaluating the causality. In the literature, many causation methods are used. It is possible, then, to apply a different method to our study and verify the correctness of the results. On the other hand, different results may help to better comprehend the information contained in the results. A second expansion of this work, it could be done with new simulations using different characteristics. For example, it is possible to vary the velocity of the robots, their trajectory, their tasks, i.e. do not give a fixed point as a goal but, for example, impose the robots to maintain a certain distance between them. Finally, a possible development is making a real experiment for the validation of the method. Recreate the scenarios in a real environment, using an autonomous robot and a piloted one. In this case, a filter should be applied to the data in order to make it useful. In fact, the data may be affected by noise, and their precision could be low.

In conclusion, we would like to address the possible use of this work. Knowing the causal relationship is a piece of important information, and it could be used in many ways. Future works could be the creation of new navigation models and algorithms or a revision of the existing ones.

Bibliography

[1] Ewart J de Visser, Richard Pak, and Tyler H Shaw. From 'automation'to 'autonomy': the importance of trust repair in human–machine interaction. *Ergonomics*, 61(10):1409–1427, 2018.

[2] Deborah R Billings, Kristin E Schaefer, Jessie YC Chen, and Peter A Hancock. Human-robot interaction: developing trust in robots. In *Proceedings of the seventh annual ACM/IEEE international conference on Human-Robot Interaction*, pages 109–110. ACM, 2012.

[3] Daniel J Fagnant and Kara Kockelman. Preparing a nation for autonomous vehicles: opportunities, barriers and policy recommendations. *Transportation Research Part A: Policy and Practice*, 77:167–181, 2015.

[4] Annemarie Turnwald and Dirk Wollherr. Human-like motion planning based on game theoretic decision making. *International Journal of Social Robotics*, 11(1):151–170, 2019.

[5] Pat Langley, Ben Meadows, Mohan Sridharan, and Dongkyu Choi. Explainable agency for intelligent autonomous systems. In *Twenty-Ninth IAAI Conference*, 2017.

[6] Tyler Vigen. https://www.tylervigen.com/spurious-correlations. Online. Accessed: 2019-11-21.

[7] Carlo Cafaro, Warren M Lord, Jie Sun, and Erik M Bollt. Causation entropy from symbolic representations of dynamical systems. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 25(4):043106, 2015.

[8] Jakob Runge, Sebastian Bathiany, Erik Bollt, Gustau Camps-Valls, Dim Coumou, Ethan Deyle, Clark Glymour, Marlene Kretschmer, Miguel D Mahecha, Jordi Muñoz-Marí, et al. Inferring causation from time series in earth system sciences. *Nature communications*, 10(1):1–13, 2019.

[9] Claude Elwood Shannon. A mathematical theory of communication. *Bell system technical journal*, 27(3):379–423, 1948.

[10] George Sugihara, Robert May, Hao Ye, Chih-hao Hsieh, Ethan Deyle, Michael Fogarty, and Stephan Munch. Detecting causality in complex ecosystems. *science*, 338(6106):496–500, 2012.

[11] Peter Spirtes and Clark Glymour. An algorithm for fast recovery of sparse

causal graphs. *Social science computer review*, 9(1):62–72, 1991.

[12] Shohei Shimizu, Patrik O Hoyer, Aapo Hyvärinen, and Antti Kerminen. A linear non-gaussian acyclic model for causal discovery. *Journal of Machine Learning Research*, 7(Oct):2003–2030, 2006.

[13] Patrik O Hoyer, Dominik Janzing, Joris M Mooij, Jonas Peters, and Bernhard Schölkopf. Nonlinear causal discovery with additive noise models. In *Advances in neural information processing systems*, pages 689–696, 2009.

[14] Kun Zhang and Aapo Hyvärinen. On the identifiability of the post-nonlinear causal model. In *Proceedings of the twenty-fifth conference on uncertainty in artificial intelligence*, pages 647–655. AUAI Press, 2009.

[15] Peter Spirtes, Clark N Glymour, Richard Scheines, David Heckerman, Christopher Meek, Gregory Cooper, and Thomas Richardson. *Causation, prediction, and search*. MIT press, 2000.

[16] Maurizio Porfiri and Manuel Ruiz Marín. Symbolic dynamics of animal interaction. *Journal of theoretical biology*, 435:145–156, 2017.

[17] C Stuart Daw, Charles Edward Andrew Finney, and Eugene R Tracy. A review of symbolic analysis of experimental data. *Review of Scientific instruments*, 74(2):915–930, 2003.

[18] Maurizio Porfiri and Manuel Ruiz Marín. Information flow in a model of policy diffusion: an analytical study. *IEEE Transactions on Network Science and Engineering*, 5(1):42–54, 2017.

[19] Raul Vicente, Michael Wibral, Michael Lindner, and Gordon Pipa. Transfer entropy—a model-free measure of effective connectivity for the neurosciences. *Journal of Computational Neuroscience*, 30(1):45–67, Feb 2011.

[20] Lionel Barnett, Adam B Barrett, and Anil K Seth. Granger causality and transfer entropy are equivalent for gaussian variables. *Physical review letters*, 103(23):238701, 2009.

[21] Lei Chen, M Tamer Özsu, and Vincent Oria. Symbolic representation and retrieval of moving object trajectories. In *Proceedings of the 6th ACM SIGMM international workshop on Multimedia information retrieval*, pages 227–234. ACM, 2004.

[22] Morgan Quigley, Ken Conley, Brian Gerkey, Josh Faust, Tully Foote, Jeremy Leibs, Rob Wheeler, and Andrew Y Ng. Ros: an open-source robot operating system. In *ICRA workshop on open source software*, volume 3, page 5. Kobe, Japan, 2009.

[23] ROS, Robot Operating System. http://wiki.ros.org. Online. Accessed: 2019-11-13.

[24] Stefano Primatesta. *Autonomous Navigation for Mobile Robots in Crowded Environments*. PhD thesis, Politecnico di Torino, Italy.

[25] Gazebo. http://gazebosim.org/. Online. Accessed: 2019-11-13.

[26] ROS, Robot Operating System. The 2D navigation stack. http://wiki.ros.org/navigation. Online. Accessed: 2019-11-13.

[27] Roland Siegwart, Illah Reza Nourbakhsh, and Davide Scaramuzza. *Introduction to autonomous mobile robots*. MIT press, 2011.