

Human machine interface to manually drive rhombic like vehicles such as transport casks in ITER

Pedro Lopes

Abstract—At the International Thermonuclear Experimental Reactor (ITER), the cargo transfer operations will be done remotely by an autonomous system, the CTS. Which is a large vehicle with a rhombic-like configuration. In this paper are shown two methods to drive rhombic-like vehicles in manual mode. Focusing in each wheel or on the vehicle center. Two devices were developed in order to test each method, one relies on a gamepad, the other on a joystick a rotational disc that was built with an encoder connected to a micro-controller. Both devices were tested by an experience user and by 12 people without experience with rhombic-like vehicle or the devices, a total of 120 successful trials were recorded and their results analyzed.

I. INTRODUCTION

The future of energy generation goes through nuclear fusion which is a sustainable and safe method to produce energy, unfortunately releases some radiation, which activates nearby materials. Handling of these materials must be done by manipulators and autonomous vehicles [1]. This will happen at ITER, the cargo transfer operations are done remotely by an autonomous system, the Cask Transfer System (CTS).

CTS is a large vehicle with a rhombic-like configuration. It has two drivable and steerable wheels across its longitudinal axis, which makes it omni-directional. In nominal operations is autonomous, however, certainly will occur situations where the vehicle must be driven remotely by an operator.

The CTS has three different operation modes, automatic, semi-automatic and manual. In automatic and semi-automatic modes the mission trajectory is computed by software. In automatic mode the wheel orientations and velocities are controlled by software, in semi-automatic mode the wheel orientations are controlled by software, but their velocities are managed by an operator, in manual mode the operator has full control of the vehicle.

While driving, an operator needs to be aware of the vehicle position, dimensions and the scenario where the vehicle operates. In this paper will be evaluated two driving methods for rhombic-like vehicles. One that focus on the wheels, and other that focus on the vehicle center. Typically, large transporters such as CTS have an handheld device to control each set of wheels, usually the devices have two joysticks one for each wheel, an example can be seen in Figure 1 on the left image, which is a driving device from Faymonville . However, there are vehicles that are driven at its center, like Kuka's omnimove, where it is used a 3-axis joystick to control the vehicle position and heading, as it is shown on the right image of Figure 1. The top image is a concept for a desk to be used at ITER to drive



Fig. 1. Manual driving devices

the CTS, it was designed along this paper and aims to drive the vehicle at its center, with a rotational disc and a joystick.

An interface and two devices were developed to drive the vehicle with each method. The device to drive focusing in each wheel uses the two joysticks of a gamepad to command each wheel. The device to drive focusing on the vehicle center has two parts, a joystick to command the linear center velocity vector, therefore changing the vehicle position, and a rotational disc to change the angular velocity, hence changing the vehicle heading.

The two devices that were developed need to be tested and evaluated. To do this, two types of tests were done, by an experienced user and by twelve people without experience with the devices or rhombic-like vehicles were selected to drive the CTS with each device inside of a simulation of ITER's tokamak building (TB). The goal is to drive the vehicle from a start to an end point, while avoiding collisions and keeping a safety distance with the walls. During the tests, a set of metrics is being saved, which will be studied in order to identify driving patterns on each device/user and to decide what is the best driving device.

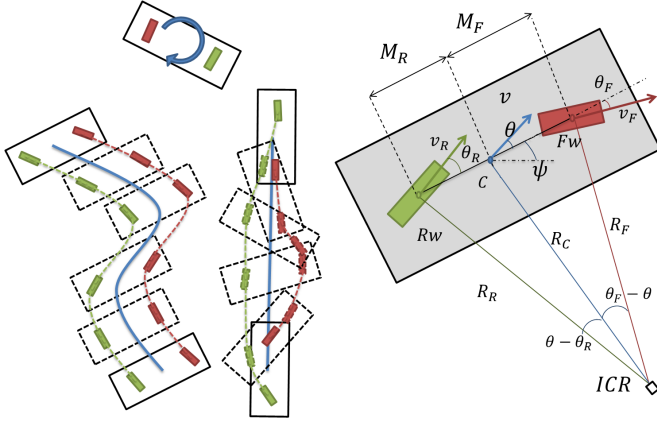


Fig. 2. Motion Capabilities and System Model.

A. Paper structure

Section II tackles the necessity to have a safe way to manually drive rhombic vehicles, and the vehicle kinematic model is explained. In Section III are proposed two solutions to drive the rhombic-like vehicles and explained the differences between them. In section IV are shown how the two devices were developed. In Section V are shown the results from testing the devices with an experienced user and with two groups of people. Lastly are presented the conclusions and future work on this topic.

II. PROBLEM STATEMENT

A. Problems while driving rhombic-like vehicle

The majority of the problems of driving rhombic-like vehicles are caused because of the amount of variables an operator needs to manage. He needs to drive two wheels at the same time, while taking into account the vehicle heading, dimensions and scenario where it operates. In Figure 2 on the left image is shown the types of movements it can make, and on the right is shown the vehicle model. For simplicity let's call the wheels, front and rear and from here on, unless stated otherwise, the green color is used to denote the rear wheel and its variables, and red color for the front wheel.

By driving the vehicle focusing on each wheel the operator needs to manage all those variables, the most common mistake is to switch the front for the rear wheel, and the vehicle motion is the opposite of what was wanted. Also, by being too focused on the wheels, usually the operator neglects the vehicle dimensions and the distance to the nearest obstacle.

Because of the harsh environment, at ITER's TB it is impossible to have the operator near the vehicle to drive it, so it needs to be remotely operated. The scenario needs to be virtually generated with the data acquired from sensors placed on the vehicle and environment. The scenario is narrow and the margin between the vehicle and an obstacle is small, which increases the difficulty of driving. Therefore a good and safe method to drive the vehicle must be found, one that is user friendly and does not hinder the vehicle movements.

B. Vehicle Kinematics

A rhombic-like vehicle has two steerable and drivable wheels across its longitudinal axis, which makes it omni-directional. In order to develop a device to drive the vehicle, first the kinematic model must be defined. Following are the vehicle kinematic equations that were obtained using the Instantaneous Center of Rotation (ICR), the image on the right side of Figure 2 was considered as the model of the vehicle.

The variables that will be controlled are the linear velocity vector, which will be split in its two components, value v and orientation θ , and the vehicle heading ψ which will be controlled via the angular velocity $\dot{\psi}$.

Using the ICR and considering no wheel slippage, the line segments R_F, R_R, R_C are orthogonal to their respective velocities, v_F, v_R, v_C that have as orientations θ_F, θ_R and θ , this information and knowing the distance from the wheel axis M_F and M_R , three triangles are defined $\langle ICR, R_w, C \rangle$, $\langle ICR, F_w, C \rangle$ and $\langle ICR, R_w, F_w \rangle$.

In order to keep the vehicle and its wheels together equation (1) needs to be met:

$$v_F \cdot \cos \theta_F = v_R \cdot \cos \theta_R \quad (1)$$

Because the wheels and the body are always together, the angular velocity is the same for all the points of the vehicle body [2], with:

$$\dot{\psi}_F = \dot{\psi}_R = \dot{\psi} \quad (2)$$

Because of equation (1) a duality is found in the linear and angular velocity equations which can be seen in equations (3), (4), (5) and (6). To find the equation for θ (7) only the wheel orientations and positions are needed. **Linear velocity value duality** v :

$$v = \frac{v_F \cdot \cos \theta_F}{\cos \theta} \quad (3)$$

$$v = \frac{v_R \cdot \cos \theta_R}{\cos \theta} \quad (4)$$

Angular velocity duality $\dot{\psi}$:

$$\dot{\psi} = \frac{v_R \cdot \sin(\theta_F - \theta_R)}{\cos \theta_F \cdot (M_R + M_F)} \quad (5)$$

$$\dot{\psi} = \frac{v_F \cdot \sin(\theta_F - \theta_R)}{\cos \theta_R \cdot (M_R + M_F)} \quad (6)$$

Linear velocity orientation θ :

$$\theta = \arctan\left(\frac{M_R \cdot \tan \theta_F + M_F \cdot \tan \theta_R}{M_F + M_R}\right) \quad (7)$$

III. PROPOSED SOLUTION

A previous work [3] already developed a device to drive the vehicle at each wheel. It uses a gamepad and with its joysticks was possible to command the wheel orientations while the trigger buttons their velocities. It is allowed to send positive and negative values for v_F and v_R but the velocity vectors of each wheel need to make an angle between $[-90, 90]$ degrees with each other, these restrictions makes the vehicle hard to

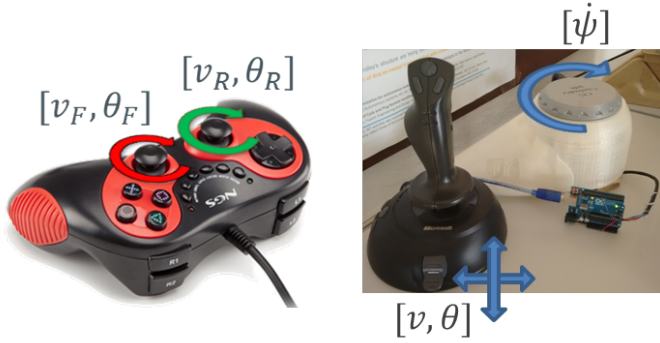


Fig. 3. Developed devices.

control in a driving perspective. From this point on this device will be addressed as Gamepad v.1.

In the author's opinion, Gamepad v.1 was not a good device to drive the vehicle. In order to make a more user friendly device, modifications are needed on how the wheel velocities and orientations are changed. The developed solution only uses the gamepad's joysticks to change the wheel orientations and velocities. The velocity value is always positive and the wheels are free to have any orientation. From this point on this device will be addressed as Gamepad v.2.

In order to develop a device that is safer and easier to drive, the proposed solution is to drive the vehicle focusing at its center instead of the wheels. By doing so, the amount of variables that an operator needs to manage decreases. He still needs to be aware of the vehicle dimensions and scenario, but it is oblivious to wheel velocities and orientations. At the vehicle center, the operator is controlling the vehicle heading ψ , and the center velocity vector value v and orientation θ .

IV. IMPLEMENTATION

A. Developed devices

Two devices were developed in this work, Gamepad v.2 and JRD. Gamepad v.2 uses a NGS Maverick gamepad. Each joystick commands a wheel, v_F , v_R , θ_F and θ_R , the operator is also free to drive the vehicle like a car, either locking the front or the rear wheel, the device is shown on the left in Figure 3. The gamepad is plugged in the computer which is periodically checking which are the button and joystick values.

JRD uses a Microsoft Sidewinder 2-axis joystick to command the linear velocity vector, v and θ , and a Lika I65-P encoder for the angular velocity $\dot{\psi}$. It connected to an Arduino for data acquisition, which is connected to the computer. In order to have an ergonomically device, a support and disc were designed and produced with a 3D printer, the complete device can be seen in on the right side of Figure 3.

B. From the vehicle center to each wheel

JRD needs a transformation of the inputs, because they are the outputs of the kinematic system v , θ and $\dot{\psi}$. Three methods to execute the transformation were studied, the first method relies in decomposing the motion in pure translation and rotational components, this will be called Split method from

here on. The second is the inversion of the kinematic system, that is made possible by an approximation of the system via a first degree Taylor expansion and using the Moore-Penrose pseudo-inverse [4] to obtain v_F , v_R , θ_F and θ_R , it will be called Pseudo-Inverse method. The last studied method was developed by Alonzo Kelly, and is the forward rate kinematics of a general bicycle model applied to the CTS, it can be seen in [5], it will be called A Kelly method.

$$\theta_F = \frac{a \cdot \theta_{TF} + b \cdot \theta_{RF}}{a + b} \quad (8)$$

$$\theta_R = \frac{a \cdot \theta_{TR} + b \cdot \theta_{RR}}{a + b} \quad (9)$$

$$v_F = \frac{a \cdot v_{TF} + b \cdot v_{RF}}{a + b} \quad (10)$$

$$v_R = \frac{a \cdot v_{TR} + b \cdot v_{RR}}{a + b} \quad (11)$$

Equations (8), (9), (10) and (11) compose the Split method, where θ_{TF} , θ_{TR} , v_{TF} and v_{TR} are the linear components of the motion, these are taken directly from the joystick. θ_{RF} , θ_{RR} , v_{RF} and v_{RR} are the rotation components of the motion, where:

$$\theta_{RF} = \theta_{RR} = \frac{\pi}{2} \quad (12)$$

$$v_{RF} = -v_{RR} = \dot{\psi} \quad (13)$$

The constants a and b are used to tune each component.

Equation (14) shows the system used to implement the Pseudo-inverse method.

$$\begin{bmatrix} v_F(k) \\ v_R(k) \\ \theta_F(k) \\ \theta_R(k) \end{bmatrix} = \begin{bmatrix} v_F(k-1) \\ v_R(k-1) \\ \theta_F(k-1) \\ \theta_R(k-1) \end{bmatrix} + J^+(u_0) \cdot \begin{bmatrix} v(k) - v(k-1) \\ \dot{\psi}(k) - \dot{\psi}(k-1) \\ \theta(k) - \theta(k-1) \end{bmatrix} \quad (14)$$

Where $J^+(u_0)$ is the Moore-Penrose Pseudo-Inverse of the Jacobian of the kinematic system using as input the last values of v , $\dot{\psi}$ and θ .

A set of simulated commands was produced to test the three methods. With crab-like and movements where θ remains unchanged, the three methods behave well. In order to have better results, the following sets were used as input:

- **Constant θ set:**

- v : Logarithmically rising until it saturates in 0.2 m/s.
- θ : Constant value of 20 deg.
- $\dot{\psi}$: Constant value of 0.1 rad/s.

- **Variable θ set:**

- v : Logarithmically rising until it saturates in 0.2 m/s.
- θ : Linear increment from 20 to 80 deg.
- $\dot{\psi}$: Constant value of 0.1 rad/s.

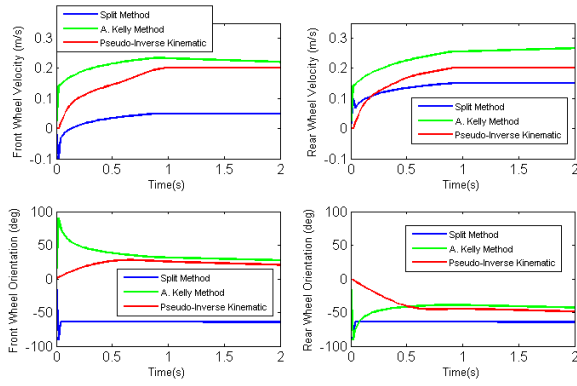


Fig. 4. Wheel velocity and orientation evolution with the third set

C. Simulated results

1) *Constant θ set:* In Figure 4 on top is shown the velocity evolution for the vehicle wheels. The curve shapes that were produced show that the Pseudo-inverse method are closer to A. Kelly method. The Split method has a little superelevation at the start, whereas the other two methods are smoother, and do not have that behavior.

In Figure 4 on bottom is shown the orientation evolution of the vehicle wheels. It can be seen that the Pseudo-inverse and A. Kelly methods converge to the same values. Regarding the wheel orientations both A. Kelly and the Split methods show spikes on the orientation values at the beginning of the test, Split method is the fastest followed by A. Kelly.

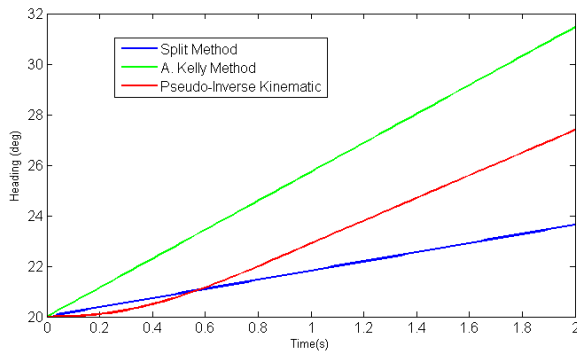


Fig. 5. Vehicle heading evolution with the third set

The vehicle heading is expected to change linearly, in Figure 5 is shown its evolution for the three methods. The vehicle heading values with the Pseudo-inverse method are closer to A. Kelly than to Split method. As can be seen the heading on all methods is linearly increasing, in the Split and A. Kelly methods it happens instantly, whereas in the Pseudo-inverse graph shows a transient behavior until time around $t=0.3$ s, after that it behaves linearly as the other two methods.

2) *Variable θ set:* The linear velocity orientation θ is being incremented from 20 to 80 degrees, which should have no influence on the vehicle heading comparing to the previous set.

By looking at Figure 6 A. Kelly and Pseudo-inverse have the same type of behavior and the Split method continues to

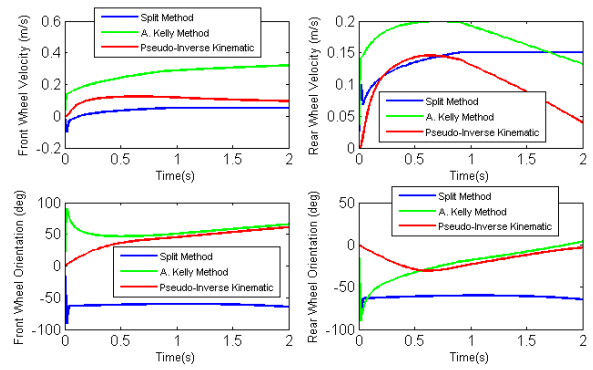


Fig. 6. Wheel velocity and orientation evolution with the fourth set

show abrupt changes. However the assumption that the vehicle heading should have the same results as the ones obtained in the previous set is not true for the Pseudo-inverse and split methods, as can be seen in the bottom part Figure 6 and more detailed in Figure 8 where is plotted the difference between the heading values for each methods in both sets.

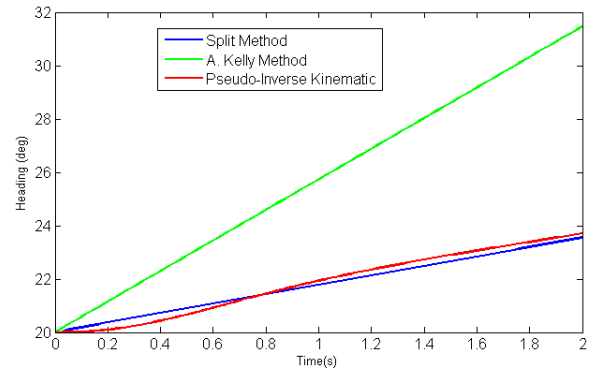


Fig. 7. Vehicle heading evolution with the fourth set

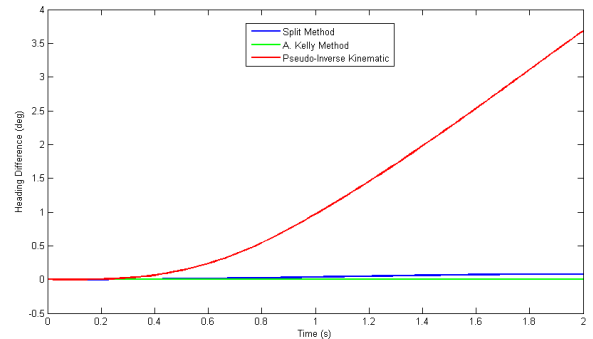


Fig. 8. Heading difference between the third and fourth sets

It can be seen in Figure 8 that at the end of the test there is a difference of approximately four degrees between the sets using the Pseudo-Inverse method. A. Kelly method has no differences and with the Split method the differences are minimal. This means that changing the orientation of the velocity vector does not affect A. Kelly method, which is the only method in which the heading with both sets.

3) *Conclusions:* The most suited methods are the Pseudo-inverse and A. Kelly because of the way they evolve throughout time. Their evolution seems more "natural", in opposition with the Split method results that show abrupt changes especially on the wheel orientations. For the usability tests will be used the A. Kelly method, because overall it obtained better results, the heading difference shown in Figure 8 is the reason why Pseudo-inverse was not chosen.

V. TEST RESULTS

To evaluate the performance of each device, two types of people will be evaluated. An experience user which is the author of this paper and twelve people without experience with rhombic-like vehicles or the devices being tested. T

A. Test scenarios, metrics and devices

The three devices, Gamepad v.1, Gamepad v.2 and JRD were tested on three different scenarios, which can be seen in Figure 9, the top image is of scenario 1, the bottom is of scenario 2 and the image on the right is of scenario 3. The vehicle is placed at the starting points in each map, the red diamonds correspond to checkpoints. In the first scenario the user needs to follow a trajectory, the red curve, in the second he must reach all three checkpoints in order to complete the tests, and in the last scenario he must navigate the vehicle inside the TB, going from the lift to one port.

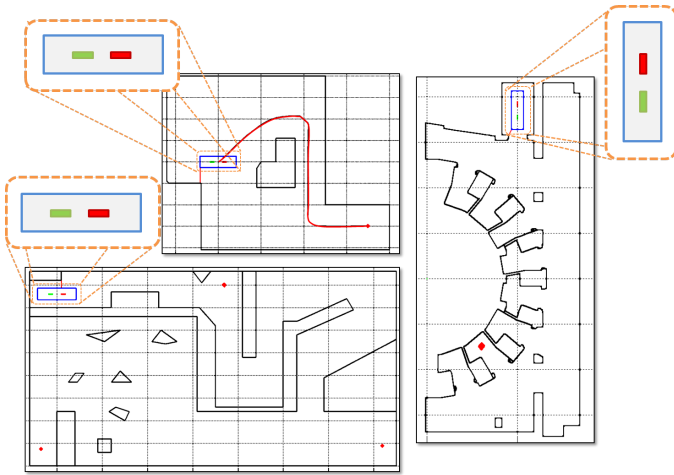


Fig. 9. Scenarios used in the tests

Following is the list of the evaluated metrics:

- **Number of collisions** - This metric counts how many times the vehicle crashed with a wall of the scenario.
- **Safety distance:** - The distance in meters between the wall and vehicle.
- **Energy:** - It is the product between the linear and angular velocities, summed along time, and it is calculated for the rear and front wheel.
- **Time Duration:** - The time that a user takes to complete a trial, it is measured in seconds.
- **Path Length** - Length of the generated path while driving the vehicle, it is measured in meters.

- **Velocity and wheel orientations:** - The different wheel velocities and orientations can be related to the driving stability, motor strain and driving patterns.
- **Vehicle Heading:** - The obtained values can be associated with the level of oscillation the vehicle experiences while moving and to driving patterns.
- **Distance to a trajectory:** - Metric only used in scenario 1, it measures the distance between the vehicle center and the path.

Most of the results are represented in graphs, the colored curve is always the average value, and the gray shaded are the standard deviation, unless stated otherwise.

B. Experience user results

1) *Scenario 1:* This scenario is used to evaluate the trajectory following performance of each device, 10 trials with each device were made. In Figure 10 on the left are shown the trajectories generated while driving with each device, and on the right is shown the average distance to the trajectory. Looking at the graphs the worse device is Gamepad v.1, many oscillations in the generated trajectories and has the higher values of the distance to the trajectory. The other two devices have similar results, JRD has slightly lower values of distance to the trajectory.

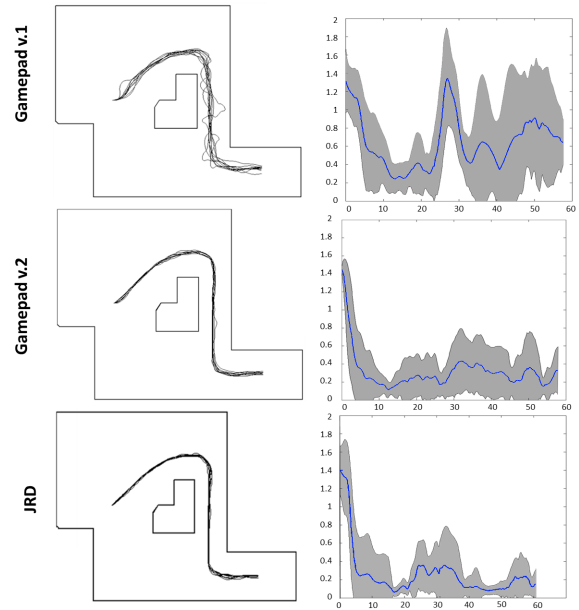


Fig. 10. Trajectories and distance to the goal trajectory.

2) *Scenario 2:* This scenario is used to evaluate the number of collisions and the distance between the vehicle and nearest wall, 5 trials were made with each device. Collisions only happened with Gamepad v.1, having a total of 49 collisions, the method to finish a trial needed to be changed allowing collisions to happen, as it was impossible to have a trial without collisions.

In Figure 11 are shown the trajectories that resulted from driving the vehicle on the left, and the distance to the nearest wall on right. Once again, the trajectories generated with

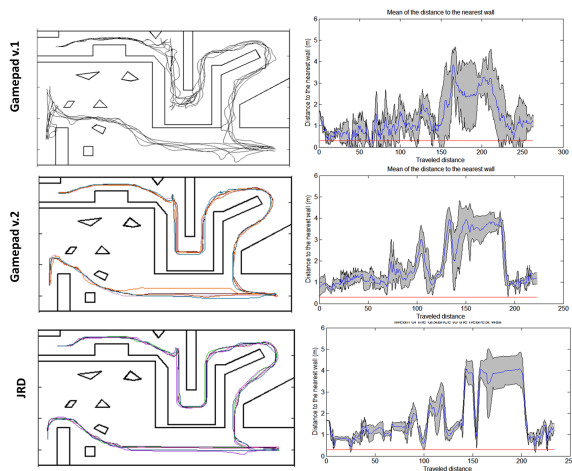


Fig. 11. Trajectories and distance to the nearest wall.

Gamepad v.1 have many oscillations and the safety distance is also lower with it. Gamepad v.2 and JRD have similar results, with JRD producing the smoothest trajectory curves and less oscillation in the safety distance.

3) *Scenario 3*: This scenario is used for the overall performance of the devices, a total of 10 trials with each device were done. In Figure 12 is shown the trajectories that resulted from driving with each device on the left, and the map split in zones depending on how easy is to drive in them on the right. As in the previous two scenario, Gamepad v.1 produced the worst trajectories with a high amount of oscillations. The other two have similar results producing smooth trajectories.

Gamepad v.1 had a total of 30 collisions, no collisions were made with the other devices.

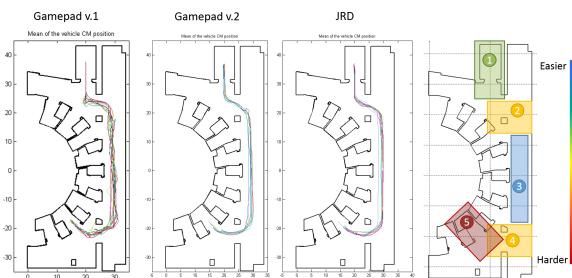


Fig. 12. Resultant trajectories of scenario 3.

In Figure 13 on the left are shown the vehicle heading and the distance to the nearest wall on the right. All three devices have similar heading results, however Gamepad v.1 is the most different with more oscillations on zone 5 both JRD and Gamepad v.2 have graphs with similar shapes. Regarding safety distance all devices have similar values with Gamepad v.1 with the lower results especially in zone 5.

In Figure 14 is shown the duration of each trial with the three devices, it is noticeable that trials with Gamepad v.1 takes almost twice the time than the other two. In order to drastically reduce the test with Gamepad v.1, the vehicle velocity was reduced, which explains these results.

4) *Experienced user - Conclusions*: The results with the experienced user revealed that Gamepad v.1 is not a good

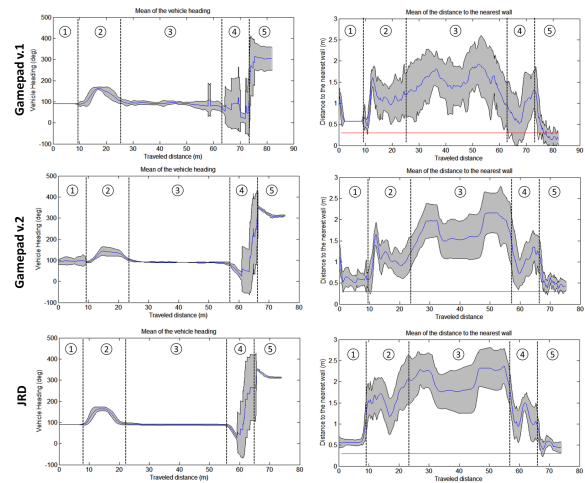


Fig. 13. Vehicle heading and safety distance.

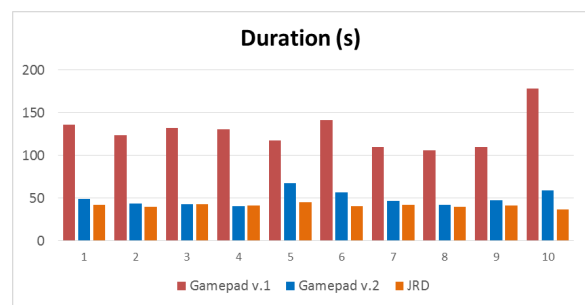


Fig. 14. Duration of each trial.

device to drive the vehicle. In order to discover what is the best device, the wheel velocities and orientations must be studied. In the group test the evaluated devices are the Gamepad v.2 and JRD because of Gamepad v.1 performance, and the evaluated scenario will be scenario 3. It is chosen because it is easier to spot patterns and only one path to the end goal can be made without backtracking.

C. Test groups results

Two groups were formed, each with 6 people, the first starts with Gamepad v.2 and the second with JRD, after 5 successful trials they switch device and complete other 5 trials which will take place in scenario 0 and 3. Scenario 0 has the same map of scenario 1 and is used for introduction and tutorial purposes. It is where the users are introduced to the vehicle and devices, after it has become acquainted with both devices, the test proceed to the next scenario. In scenario 3 a user needs to drive the CTS from the inside of the lift to a docking port without any collision. After 5 successful trials, he switches device and repeats the process. The people that took part in the tests have an age range from 11 to 50 years old, with an average of 26 years. All of them have experience with computers and 58% had experience with console games or gamepads. A total of 200 trials were performed, each user had to complete the trial five times, which means going from the start position to the end goal without collisions, with a total of 120 successful trials.

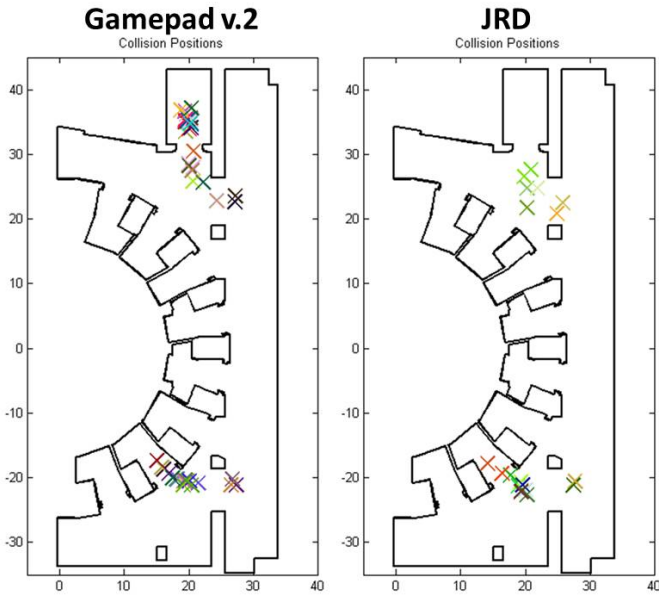


Fig. 15. Collision positions.

1) *Collisions*: Figure 15 shows the positions where collisions happened. Looking at both images it can be seen that they happen almost in the same places, with the exception of the high number of collisions in the starting zone with Gamepad v.2. In order to leave that zone, the vehicle needs to go straight down, and for some users this was not an easy task.

In Table I is shown the total number of collisions. I can be seen that Gamepad v.2 has the higher number of collisions. Most collisions happened in Group 1 because they had to learn how to navigate the scenario with Gamepad v.2 first. The same behavior is seen in Group 2 with the high amount of collisions with JRD.

TABLE I
TOTAL NUMBER OF COLLISIONS

# Trial	Gpad						JRD					
	1	2	3	4	5		1	2	3	4	5	
Group 1	28	8	4	2	1	43	0	1	5	1	0	7
Group 2	13	0	4	3	0	20	3	3	2	0	2	10
Total	41	8	8	5	1	63	3	4	7	1	2	17

D. Safety distance

In Figure 16 is shown the average distance to the nearest obstacle, the top image refers to Gamepad v.2 results and the bottom to JRD's. The blue curve, and the standard deviation, the gray area. It can be seen that the values and shapes of the curves are identical. As expected the lower values are located at the star and end of the trial. The curve generated by Gamepad v.2 shows more oscillations than JRD results, this means that the trajectories produced by Gamepad v.2 have more oscillations than JRD.

1) *Vehicle heading and trajectory*: In Figure 17 is shown the results for the vehicle heading, the top refers to Gamepad

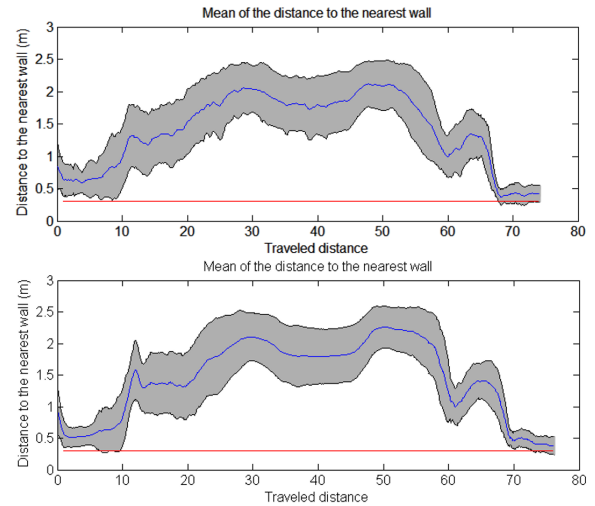


Fig. 16. Average of all minimum distances to the nearest wall

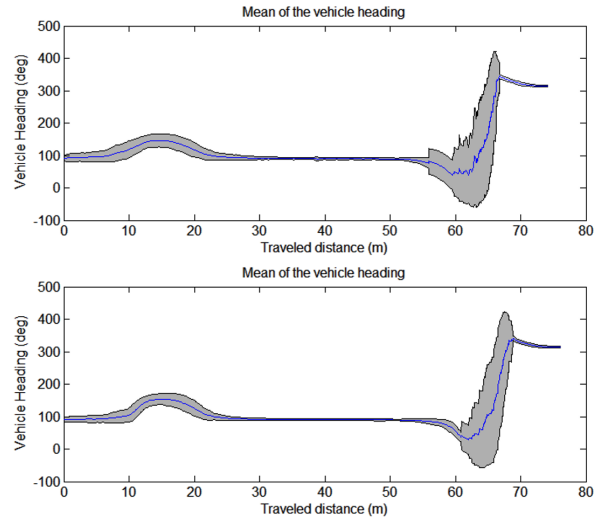


Fig. 17. Average of the vehicle heading

v.2 and the bottom to JRD. Both graphs have similar shapes and values because there is only one possible path to complete the test without backtracking. In Figure 18 are the trajectories of the vehicle center, on the left for Gamepad v.2 and on the right for JRD. The shapes are similar, however with JRD the trajectory has more straight parts, where with Gamepad v.2 a smooth curve was obtained.

2) *Wheel orientations and velocities*: In Figures 19 and 20 are shown the wheel orientations for Gamepad v.2 and JRD respectively. Looking at JRD results, both wheels have similar results with a low standard deviation which means most users drive similarly. With Gamepad v.2, the front and the rear wheel results are distinct. The rear wheel results are similar to JRD results, however the standard deviation of the front wheel is very high, this is because some users drove the vehicle like a car, fixing the front wheel orientation and focused only on the rear wheel.

In Figures 21 and 22 are shown the wheel velocities for Gamepad v.2 and JRD respectively. With JRD the velocity results for both wheels have the same shape. With Gamepad v.2

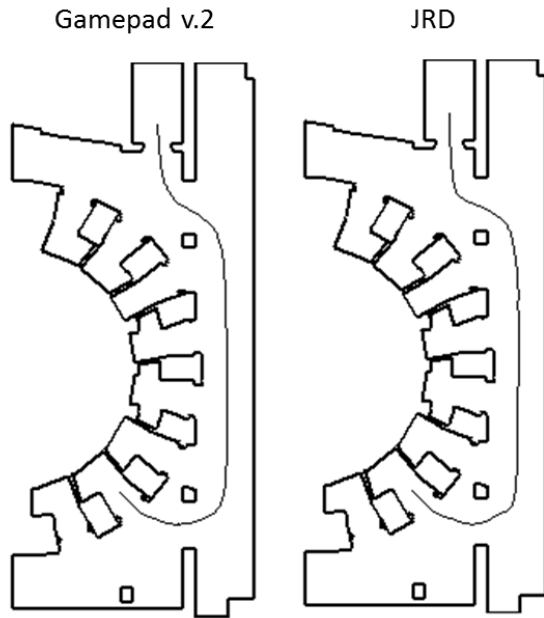


Fig. 18. Obtained trajectories

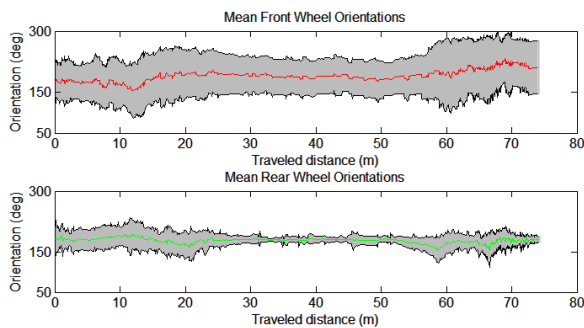


Fig. 19. Average of the wheel orientations with Gamepad v.2

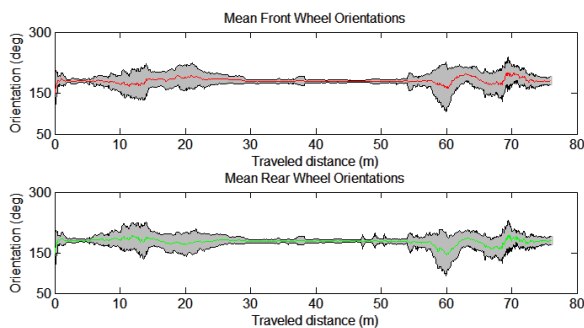


Fig. 20. Average of the wheel orientations with JRD

the velocity results show the same behavior as the orientations, the front wheel has a large standard deviation because some users while focusing on the rear wheel, left the front wheel still, without motion.

3) *Duration, length and energy for all users:* In Figure 23 are the duration, length and energy for each wheel. The duration and length values for Gamepad v.2 and JRD are close, with JRD values being slightly higher. Looking at the energy

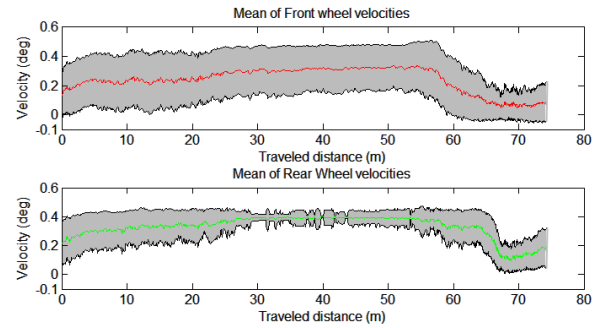


Fig. 21. Average of the wheel velocities with Gamepad v.2

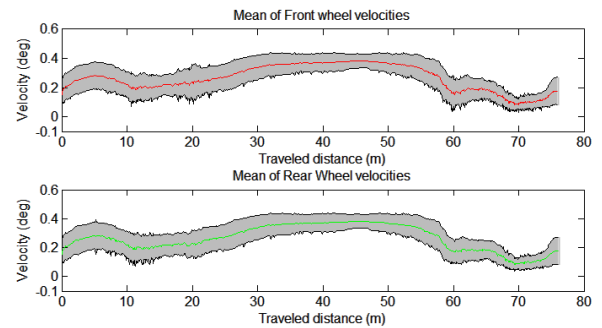


Fig. 22. Average of the wheel velocities with JRD

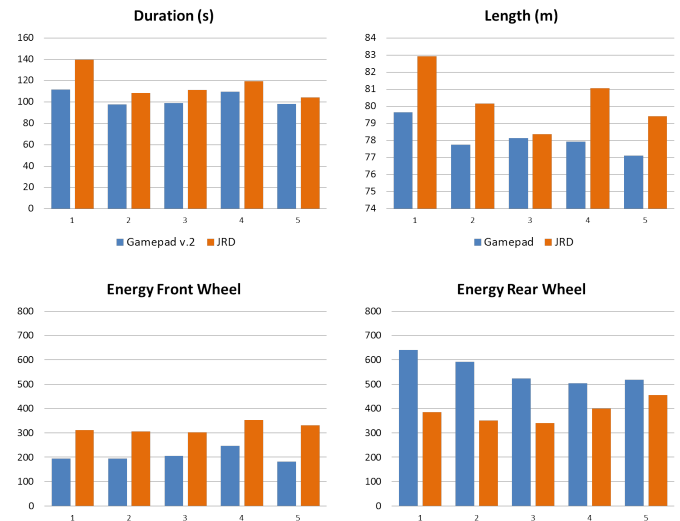


Fig. 23. Duration, length and energies for all users

in each wheel, with JRD both wheels have similar values. With Gamepad v.2 the rear wheel energy values are much higher than the front wheel, this happens because users drive the vehicle like a car with Gamepad v.2.

E. User feedback

All users from the test groups agreed that driving the vehicle at its center with JRD was easier and more intuitive than driving each wheel with Gamepad v.2 and the total number of collisions with Gamepad v.2 corroborates this opinion.

However 5 users said that with time and practice Gamepad v.2 should be better than JRD given the freedom that it provides.

The majority of the users focused mainly on the rear wheel of the vehicle when driving with Gamepad v.2, driving the vehicle mainly as a car or simply let the front wheel stay still. Only two users had successful trials controlling each wheel independently. This method to drive the vehicle was never used with JRD.

Some users expressed the need of having the velocity tune near the encoder instead of the built-in linear potentiometer on the bottom part of the Joystick, other suggested to use the buttons at the top of the Joystick's shaft to tune it.

VI. CONCLUSIONS AND FUTURE WORK

The devices that were developed enable a manual driving mode for the vehicle, however there are pros and cons with each device. With Gamepad v.2 the vehicle retains all of its motion capabilities, but by commanding each wheel the operator tends to commit more mistakes. With JRD the vehicle motion capabilities are limited because the wheel cannot be controlled independently and by doing this the vehicle becomes easier to drive the operator.

The test results concluded the previous statement, due to the high amount of collisions and the need to drive the vehicle like a car when driving with Gamepad v.2. The same feature was also present in JRD but was not used, the motion limitations with JRD did not have a negative impact while driving the vehicle in the evaluated scenarios. Some people however due to past experience with gamepads put the hypothesis that with proper training JRD results will be surpassed.

Future work can be done by investing more time into testing with more people and on different scenarios with different goals. For example a scenario where in every trial the user needs to go to s different location. This test will be needed in order to prove that JRD motion limitations do not hinder the operations that need to be done.

REFERENCES

- [1] C González Gutiérrez, C Damiani, M Irving, J-P Friconeau, A Tesini, I Ribeiro, and A Vale. Iter transfer cask system: status of design, issues and future developments. *Fusion Engineering and Design*, 85(10):2295–2299, 2010.
- [2] Elwood Russell Johnston. *Vector mechanics for engineers: statics and dynamics*, volume 1. Tata McGraw-Hill Education, 2009.
- [3] Joao Bivar and Alberto Vale. Behavior of digital and analog controller devices for manual driving of rhombic like vehicles. *2011 19th Mediterranean Conference on Control & Automation (MED)*, Jun 2011.
- [4] Roger Penrose. A generalized inverse for matrices. In *Mathematical proceedings of the Cambridge philosophical society*, volume 51, pages 406–413. Cambridge Univ Press, 1955.
- [5] Alonzo Kelly. A vector algebra formulation of kinematics of wheeled mobile robots, 2010.