# The Castaway Problem

Gustavo A. Cardona
*Department of Electrical and Electronic Engineering*
*Universidad Nacional de Colombia.*
Bogota, Colombia
gacardonac@unal.edu.co

***Abstract*—This work contains the report of the algorithm performed to solve the castaway problem. Which consists of a castaway trying to escape from an island where a shark is waiting to hunt him on the border of the island. Considering the shark has greater velocity than the castaway and the latter can be initially located arbitrary, the main questions are, there is any way for the castaway to escape? what is the best strategy for the castaway and shark respectively? all the answers are explained and then simulated to demonstrate the possible strategy taken by each agent.**

## I. Problem Statement and Preliminaries

The problem describes the conditions of how a castaway $ca \in \mathbb{R}^2$ is left on an island $y < \pm\sqrt{1-x^2}$, where $x$ and $y$ are the axis coordinates, a shark $sh \in \mathbb{R}^2$ is in the border $y = \pm\sqrt{1-x^2}$ ready to hunt the castaway. The purpose of the castaway is to escape from the island using a boat, where he can move on the island with speed $v_{ca} = s$ and in the same way, the shark can move on the boundary with speed $v_{sh} = 4s$.

**Assumption 1:** The initial location of the castaway is arbitrary.

**Assumption 2:** Castaway and shark can see each other which means they can know all the time where the other is.

**Objective:** The main objective is to find the way the castaway can escape, this will happen if the castaway reaches the boundary of the island and the shark is not there. On the other hand, if the castaway reaches the boundary and the shark is there, the shark eats him.

Considering the problem takes constant values for the velocity, it can be described by the use of Uniform Motion,

$$\begin{array}{c} x_f = x_0 + v(t_f - t_0) \\ v = \frac{x_f - x_0}{t_f - t_0} \end{array} \tag{1}$$

where $x_f$ is the final position $x_0$ is the initial position, $v$ is the velocity, $t_f$ final time and $t_0$. And in the same way by the Uniform Circular Motion, $w = \frac{v}{r}$, where $w$ is the angular velocity, $w = \frac{\Delta\theta}{\Delta t}$, $\theta$ is the angle, t is the time $arc = \theta * r$.

## II. Strategies and Algorithm

**Question 1:** Can the castaway escape? yes, he can if the correct strategies are applied. Strategies that make him obtain a favorable position that permits him to reach a point in the border faster than the shark do.

**Question 2:** what is the best strategy for the shark? the best strategy is to project a vector from the castaway along to the border and follow that point searching always for the closest path.

**Question3:** what is the best strategy for the castaway? the castaway depends strictly on the initial conditions if a minimal distance to the shark is given at the initial state the castaway can escape just by going straight forward to the border. However, not always that scenario is given so the global solution is to reach a radio where the angular velocity is larger in comparison to the angular velocity of the shark, by doing this the castaway just have to turn around until the minimum distance that the castaway needs to reach the border before the shark does.

### A. Kinematic Model

Based on (1) and the equations for Uniform Circular Motion the kinematic of both agents can be described by a combination of these two behaviors. Taking into account that the shortest distance in a convex space is the euclidean distance any time that the castaway will try to reach the border he has to do it going straight. On the other hand, not only does the shark moves in circles but also the castaway when the strategy to win in angular velocity is played. Finally, the kinematic model used for the castaway in this problem is shown below,

$$\begin{array}{c} ca_x[i] = ca_x[i-1] + v_{ca_x}[i-1]dt \\ v_{ca_x}[i] = v_{ca}cos(\theta_{ca}) \\ ca_y[i] = ca_y[i-1] + v_{ca_y}[i-1]dt \\ v_{ca_y}[i] = v_{ca}sin(\theta_{ca}) \end{array}$$

where, $ca_x$ and $ca_y$ are the position of the castaway in the $x$ and $y$ axis respectively, $v_{ca_x}$, $v_{ca_y}$ are the velocities of the castaway in the $x$ and $y$ axis, $\theta_{ca}$ is the angle of the castaway, $dt$ is the time differential step. Besides, it is important to notice that both the castaway and the shark can present circular motions so the kinematic can be expressed as,

$$\begin{array}{c} ca_x[i] = norm(ca[i-1])cos(\theta_{ca}[i-1] + sgn(\theta_{ca-sh})w_{ca}[i]dt) \\ ca_y[i] = norm(ca[i-1])sin(\theta_{ca}[i-1] + sgn(\theta_{ca-sh})w_{ca}[i]dt) \\ w_{ca}[i] = v_{ca}/r \end{array}$$

where $norm(ca)$ is the norm of the castaway, $sgn()$ is the sign function. $\theta_{ca-sh}$ is the angle between the castaway and the shark, and $w_{ca}$ is the angular velocity for the castaway at radius $r$,

$$\begin{array}{c} sh_x[i] = norm(ca[i-1])cos(\theta_{ca}[i-1] + sgn(\theta_{ca-sh})w_{sh}[i]dt) \\ sh_y[i] = norm(ca[i-1])sin(\theta_{ca}[i-1] + sgn(\theta_{ca-sh})w_{sh}[i]dt) \\ w_{sh}[i] = v_{sh}/r \end{array}$$

where $norm(sh)$ is the norm of the shark, and $w_{sh}$ is the angular velocity for the shark at radius $r$, which is 1 all the time if the shark remains at the border of the island.

### B. Strategy

The strategy implemented to make the castaway escape from the island all the time no matter what the initial conditions are, is through the use of a state machine. States that allow the castaway all the time to takes a decision that put him in advantage regarding the shark due to the best strategy the shark can play is to follow the projection of the closest points from the castaway to the island border. The state machine is depicted in Fig. 1 in the same way the transitions between states, here I am considering that all the time the shark is playing its best response. The states are:

**Task Allocation:** this state is in charge of evaluating all the time the distance of the shark to the closest point from the castaway to the border, and the distance from castaway to the same point which will be called the target point. If the distance from the shark to the target is 4 times bigger than the castaway to the target the task allocated is going border. Otherwise, the state is switched to a winning radius. It is worth to mention that the distance to the target from the shark is an arc while the distance from the castaway is the euclidean distance.

**Winning Radius:** meaning that the castaway has to go to a radius bigger than $r > 0.214$ but smaller than $r < 0.25$ due to if we consider the shark can move at velocity 4 times bigger than the castaway. In the case, they are opposed to each other from the origin, which is the maximum distance the shark has to cover to reach the target this distance is equal to $\pi$ so the time the shark need is $t_{sh} = \pi/4$ so if we want the castaway to reach the target at the same time he needs to be at least $\pi/4$ from the border to have $t_{ca} = \pi/4$. As a result, the distance from the border to let the castaway reaches the target first is $r > 0.215$. On the other hand, when the castaway turn around the origin the shark will start to do it as well, it is important to notice that as the shark can not change the turning radius the castaway do, so considering the Uniform Circular Motion it is noticeable that when the turning radius of the castaway is smaller than the shark at some point the castaway will start to have bigger angular velocity. The radius where the castaway starts to has a greater angular velocity that the shark is $r_{ca} < 0.25$ due to the differences in the linear velocities.

**Turn Around:** Once the castaway has reached the winning radius the castaway has to turn around until the shark will be on the opposite side of the island. Taking into account that the intersection of the radius that allows the castaway to win and have bigger angular velocity is small, the castaway is set to try to stay in the middle of $0.215 < r < 0.25$.

**Go to Border:** as its name indicates this state is chosen when the algorithm assures that the castaway can reach the target first than the shark. Here the castaway stops to turn around and begins to going straight to the target, which is located at the closest point in the island border.

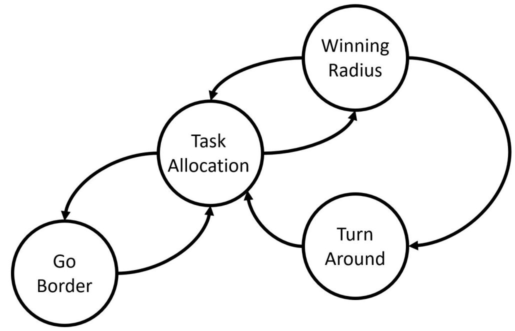


Fig. 1. State Transitions

## III. Simulations and Results

The simulation was performed in a combination of two software, Matlab, and V-Rep pro edu, the latter has virtual quadrotor model that can be used in realistic simulations. Additionally, the simulator offers the possibility to modify the environment as need it. In order to show the results obtained in the algorithm, the communication between Matlab and V-Rep was established by the use of an API.

I considered some important cases where the castaway behaves differently, allowing to show the switching strategy that the algorithm does. First, let's consider when the castaway and the shark are on opposite sides of the island and the distances difference is favorable to the castaway.

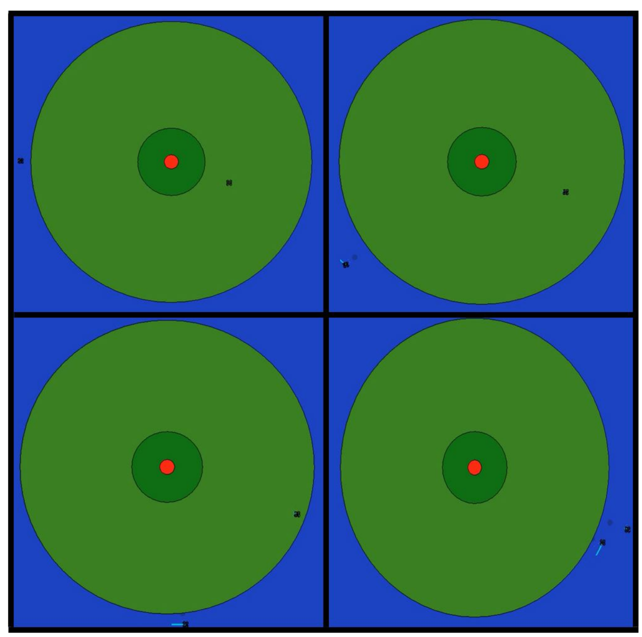

Fig. 2. First case: the castaway has enough distance to reach the target first than the shark in the initial conditions.

For all of the following simulations, the red circle is the origin, there are two other circles. First, one in dark green representing the minimum radius that the castaway needs to win to the shark if the latter is on the opposite side. Second, the bigger circle representing the size of the island, making the

dark line as the border of the island. The quadrotor outside the blue area represents the shark and the quadrotor in the green area is the castaway.
In Fig. 2 can be shown in the first scenario, where the quadrotor is initialized on a distance where he knows that he can reach the border faster than the shark. That is why the sequences of images trying to show how either the castaway quadrotor, and the shark quadrotor are approaching to the target, but the castaway reaches it first. Considering the initial conditions the simulation performed the behavior expected due to the castaway is always evaluating if he can go straight to the island or not.

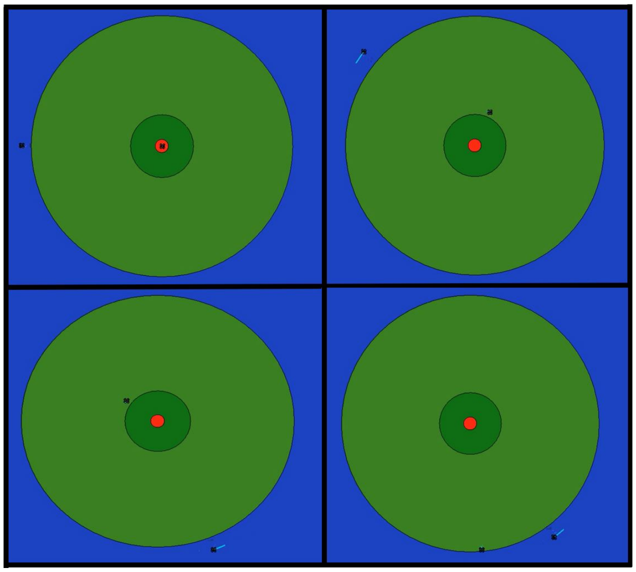

Fig. 3. Second case: the castaway is settled in the origin as initial conditions.

The second scenario is shown in Fig. 3 the castaway is settled in the origin. The strategy of the castaway changed in this case due to no matter the direction the castaway takes, if he goes straight to the border the shark will defeat him. As a result, what the castaway does to win is going into the opposite direction of the shark to the winning radius. Then he starts to turn around. As consequence that the castaway has greater angular velocity, after some laps the castaway can achieve an advantageous distance. Once this distance is reached the castaway starts to go straight to the border where the last image of the sequence show he reaches the border first.

The third scenario is shown in Fig. 4 the castaway is settled in front of the shark and out of the winning circle. The consequences here are that just by linear velocity the castaway will never going to win to the shark. Once again the strategy played by the castaway is going to the wining circle and start to turn around until the time wasted to reach the border is longer than the one wasted by the castaway. Here the castaway starts to go to border as shown in the last image in the sequence when he reaches the border first than the shark.

Finally, a fourth interesting scenario is shown in Fig. 5. Here the castaway is initialized inside of the winning circle and the shark on the opposite side of the island. This time is impossible for the castaway to move straight to the border and reach it first than the shark. The reason why the castaway starts to move away from the shark but approaching the winning circle. Castaway turns around the wining circle until he has an advantageous distance to the target and then starts to reach the target straight forward.

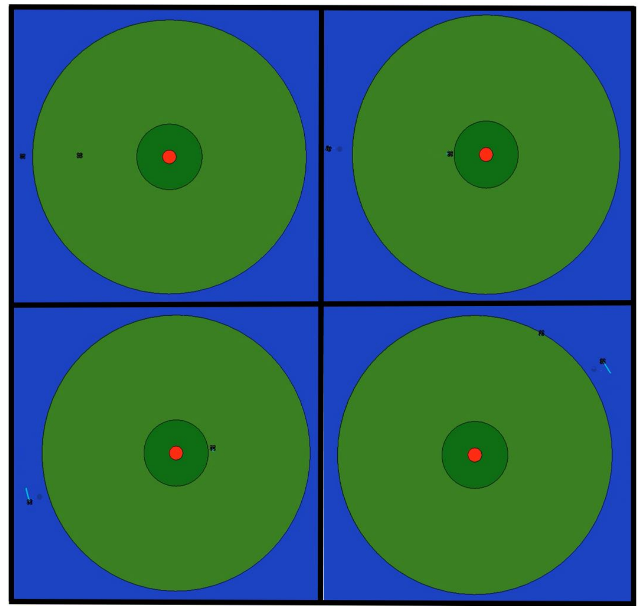

Fig. 4. Third case: the castaway is settled right in front of the shark.

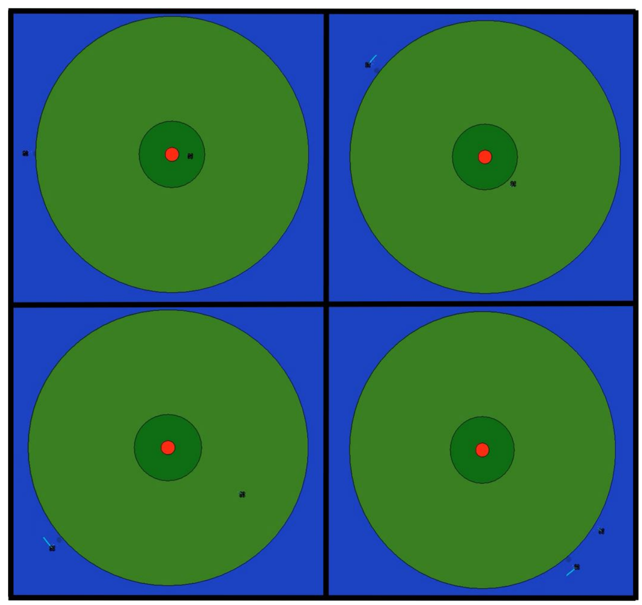

Fig. 5. Fourth case: the castaway is settled inside of the winning radius.

**Run Simulation:** In the GitHub repository there are four videos regarding this paper report with the four scenarios. Even though if the user wants to run the simulation, it is necessary to install either Matlab and V-Rep. Copy all the file to a local folder, open the V-Rep scene run it and then run the Matlab file.