

# Application of vector algebra and physics in designing steering behaviours of autonomous agents for realistic display in 2 dimensions\*

Richard Čerňanský

Slovenská technická univerzita v Bratislave  
Fakulta informatiky a informačných technológií  
xcernansky@stuba.sk

November 3, 2022

## Abstract

In this article we will focus on the problematic of steering behaviours of autonomous agents in software and display methods. The importance of this topic hides in usability in animation, film effects and mainly in games. If we want the user to have the best realistic experience from movement of the objects possible, we have to obey specific rules of physics that will be explained. Applying the knowledge, we will then see methods how to simulate a seek and arrival steering behavior. As a tool we will use a p5.js library of JavaScript language.

## 1 Introduction

Development in animation and games forces the creators to make the games more and more realistic. One of the most important aspects of realistic perception of a game is the movement of objects and the animation itself. Designing movement behaviors of these objects is then necessary.

In this article, we will call the objects autonomous agents. More about them will be explained in part 2.1. After understanding the concept of autonomous agents, we will state a problem in part 3.2. In part 4 we will have a closer look at the problem and derive some formulas. Finally, we will simulate the steering behavior, stated as a problem (p.3.2) in part 5 by writing the code in p5.js (p.5.2).

## 2 Autonomous agents

### 2.1 Definition of autonomous agents

The autonomous agents generally refer to an entity that chooses how to act in its environment without any influence from a global plan or a leader. In games, these agents are not controlled by a player, but they are important part of the game because

---

\*Semestrálny projekt v predmete Metódy inžinierskej práce, ak. rok 2022/23, vedenie: Mirwais Ahmadzai

their action can significantly influence the flow of the game. For example, a villain who runs away from police decides on his own that he wants to escape and starts an action. There are three key components of autonomous agents we want to keep in mind.

- An autonomous agent has limited ability to perceive its environment. It makes sense that if autonomous agent must decide on its action, it should be somehow aware of the environment it is located in. The question here is how much limited the ability is. If we wanted the object to be all-knowing creature aware of everything else around it, we need to give it an access to information about everything. On the other side if we wanted it to have just very narrow view of the environment, let's say just a few pixels around it.
- An autonomous agent processes the information from its environment and calculates an action. The action is represented as a force that influences the object. For example, a police officer sees a thief and is attracted to him. The attraction is represented by a force pointing towards his location.
- An autonomous agent should have no leader. This is not something that defines every autonomous agent. Sometimes you need to state some global rules that it must follow, but mostly you want the object to decide on its own, calculate its own actions.

## 2.2 Types of behaviors according to the number of objects involved

Craig Reynolds (odkaz na clanok) in his paper introduces some types of behaviors that could appear talking about autonomous agents and how they behave. They divide into two main groups:

- Simple behavior for individuals and pairs
- Combined behaviors and groups

In this article we will focus on the first mentioned.

## 3 Problem statement

### 3.1 Braitenberg vehicle

Braitenberg vehicle is an entity that is a hypothetical self-operating machine that can make decisions about how to behave in an environment based on its sense perception. (odkaz na knihu Vehicles by Valentino Braitenberg).

Vehicle A steers away from the light source and vehicle B steers towards the light source. It is not just about steering away or towards the sun. We could say that these vehicles feel emotion about the object. Vehicle A feels fear from the sun and vehicle B is attracted to it. We want the term vehicle to be clear because in the paper when I mention vehicle, I will be referring to the concept of Braitenberg vehicle and its characteristics.

### 3.2 Problem to solve

Let's say we want to simulate a simple behavior for a pair (odkaz na 2.2). A vehicle (imagine a car) that wants to arrive to a specific position (f. e. parking lot). We could

say that in the car there is a driver. He perceives the environment with his eyes and sees the location of the parking lot. Processes his distance from the place and calculates and action. He starts steering towards the parking lot (attraction towards the parking lot is the same as vehicle B in 3.1). Notice the word steering here which is very important. We do not want to simulate an unrealistic car that is immediately able to turn around and head towards the target in a maximum speed. We want the animation to be smooth and realistically looking. What rules do we have to follow if we want to simulate a situation like this?

## 4 Seek and arrive – a pursuit of a static target

### 4.1 Simple vehicle model

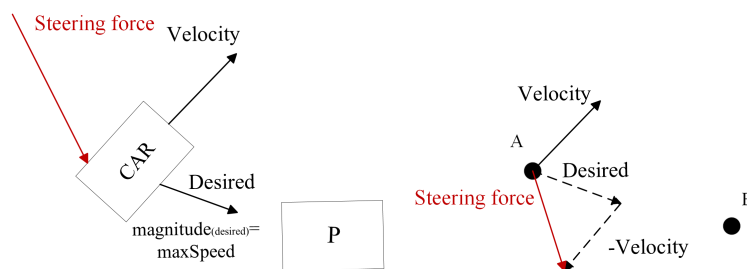
Craig Reynolds (odkaz na clanok) in his paper (on page 6) presents a simple vehicle model which I need to introduce to you. There are 6 attributes that our vehicle will have.

Simple Vehicle Model: {mass- scalar; position- vector; velocity- vector; maxForce - scalar; maxSpeed- scalar; orientation- N basis vectors}

### 4.2 Characteristics of seek and arrive

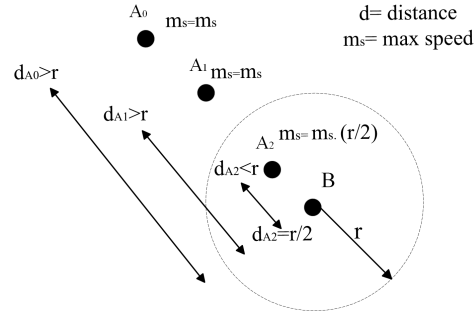
#### 4.2.1 Seek

Seeking steers a character towards a specified position in a space. This behavior aligns the vector of a velocity towards the target. But how do we calculate the steering force if we do not want the vehicle to steer completely right after seeing the target? We can diagram our problem from 3.2.



#### 4.2.2 Arrive

When we want to simulate arrival, we must think about gradually decreasing the speed of the vehicle and eventually stop it.



From the diagram we can assume two things:

$$d > r \Rightarrow speed = maxSpeed$$

$$d \leq r \Rightarrow speed = (d/r).maxSpeed$$

## 5 Simulating the behavior in p5.js

### 5.1 What is p5.js?

p5.js is a JavaScript library for creative coding, with a focus on making coding accessible and inclusive for artists, designers, or educators. It is a collection of pre-written code and provides us with tools that simplify the process of creating interactive visuals with code in the web browser. p5.js is free and open source. We will use it to simulate the problem from chapter 3.2.

### 5.2 Writing the code

(Just a comment. Here I plan to explain certain parts of the code according to the explanation from 4.2. The code is already done, and you can check it here. Clicking on the red play button in the left corner runs the simulation.)

#### 5.2.1 class Vehicle

## 6 Other possible steering behaviors

## 7 Conclusion

### Acknowledgement

I would like to thank my friend Lukáš Částven for introducing the problem to me and arousing my interest in the topic. . .