

# Steering Behaviour for Computer based Intelligent Agents

Bhagya Shree Jain  
Department of CS & E  
FET, SSTC, SSGI,  
Junwani, Bhilai (C.G.)  
India

Shrikant Tiwari  
Department of CS & E  
FET, SSTC, SSGI,  
Junwani, Bhilai (C.G.)  
India

Sagar Chandrakar  
Department of CS & E  
FET, SSTC, SSGI,  
Junwani, Bhilai (C.G.)  
India

## ABSTRACT

This paper presents steering behaviours through intelligent agents populating virtual or simulated environments. We present a reliable effort towards automatic path planning of intelligent computer agents with the integration of trajectory optimisation and obstacle avoidance techniques. Our approach will increase the level of accuracy in movement and would be reliable in the sense that the best computable path will always be available. Behaviour programming is facilitated by a set of presentations of the environment that uses convenient frames of reference in two dimensional coordinate systems. Four different steering ways are examined for movement of intelligent agents. Combinations of steering behaviours can be used to achieve high level goals of movement or locomotion. Our approach can be tailored to suit any particular need which requires automation in locomotion.

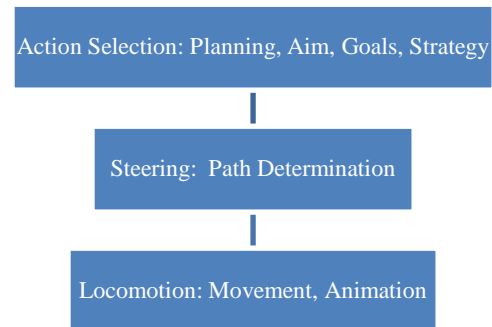
## Keywords

Virtual environments, intelligent agent, steering behaviour

## 1. INTRODUCTION

Steering Behaviours present solutions for one major requirement of intelligent agents in animation, games, virtual reality and robotic movement: the ability to navigate around their world may be real may be simulated in a life-like and improvisational manner. These Steering Behaviours are largely independent of the particulars of the agent's means of movement in there space. Combining different type of steering behaviours can be used to achieve higher level goals from simple (for example: movement from one place to another while avoiding obstacles, following specific path or join that group of agents...) to complex (steering ships [1], traffic directing [2]). This paper presents some of the efforts we made with the goal of allowing a friendly control over a large number of intelligent agents in an interactive virtual environment. Our work has been specially aimed for the control of intelligent agents in the number of scenarios for simulation experiments [3,4].

Intelligent agents are intended for use in computer animation and various interactive media such as games and virtual reality. These agents have little ability to improvise their actions. This paper will use the term virtual to denote these agents which, rather than being simulations of an electrical or mechanical device in the real world, are instead intelligent agents in a virtual world. In this paper the term behaviour is used to refer to the improvisational and life type actions of intelligent agents. The behaviour of an intelligent agent can be better understood by dividing it into layers. Figure 1 shows a division of motion behaviour for intelligent or autonomous agents into a hierarchy [5]: action selection, steering, and locomotion.



**Figure 1: A Hierarchy of motion behaviour [5]**

This paper will focus the middle layer of the behavioural hierarchy that is steering. It will briefly describe a simple model of the locomotion, but only in enough detail to provide a solid foundation for the discussion of various steering behaviours. There will be some discussion of action selection, but primarily in the context of basic steering behaviours [5, 6].

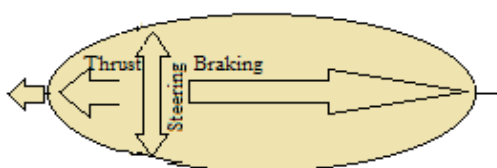
## 2. ENVIRONMENT AND INTELLIGENT AGENT

Simulation environments are composed by a few to large amount of different agents, each one with its own unique behaviour and all of them interacting and competing with each other. The preparation of locomotion simulation scenarios inside such environment is one of the most time and effort consuming tasks. The possibility to include intelligent elements that react to each other using its internal defined behaviour can be dramatically simplified by the creation of realistic scenarios. Intelligent agents or elements must be guided by lines of code, consisting of sequence of commands. Depending upon the experimental configuration, changes can be done in destination, behaviour, acceleration, speed or manipulation of the intelligent agent [3, 4]. The approach taken in this paper is to make steering behaviours as entirely independent from the underlying locomotion scheme. A simple locomotion model will be presented in order to make the discussion of steering behaviours more accurate. This locomotion model will be based on a simple vehicle model.

This Simple vehicle model is based on a point mass approximation [5]. On the one hand it is very simple and computationally cheap physically-based model. On the other hand, it cannot be a very interesting physical model because point masses do not exist in the real world as any physical object with mass must have a non-zero radius and hence a moment of inertia. This use of a very simple non-physical vehicle model is merely for convenience and intended to be

without loss of generality. We can always substitute it with a more plausible, more realistic physically based vehicle model.

A point mass is stated by a position property and a mass property, in addition to it, the simple vehicle model includes a velocity property. The velocity is changed or modified by applying forces. Since this is a vehicle, these forces are generally self-applied, and hence limited. Take a case where a typical force which adjusts a vehicles velocity is thrust, generated by the vehicles own power plant, and hence limited in magnitude by the capacity of the power plant. For the simple vehicle model, this notion is stated by a single Maximum force parameter. Most vehicles are characterized by a maximum speed. Typically this limitation is due to the interaction between acceleration due to their finite thrust and the deceleration due to drag, friction or through the momentum of reciprocating parts. As an alternative to real simulation of all these limiting forces, the simple vehicle model includes a Maximum Speed variable. This speed limit is enforced by a kinematics of the vehicles velocity vector. Finally, the simple vehicle model includes an orientation, which when taken together with the vehicles position form a velocity-aligned local coordinate space to which a geometric model of the vehicle can be attached. In this vehicle model, there is exactly one vector quantity which is a desired steering force where the control signal passed from the steering behaviours to the locomotion behaviour consists. More realistic vehicle models can have very different sets of control signals. For example a car has a steering wheel, accelerator and brake each of which can be represented as scalar quantities. A generic control system can be described as an agent that uses the information provided from the input set to perform the required transitions of state and value then evaluate the output information sent to the controlled system [3]. It is possible to scale and tabulate a generalized steering force vector into these scalar signals: the side component of the steering vector can be interpreted as the steering signal; the forward component of the steering vector can be mapped into the accelerator signal if positive, or into the brake signal if negative. These mappings may be asymmetrical, for example a car can decelerate due to braking much faster than it can accelerate due to engine, as shown in Figure 2.



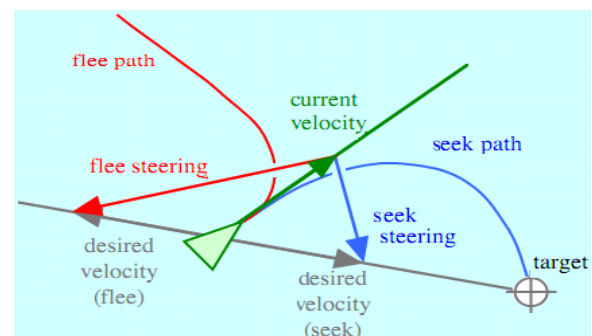
**Figure 2: Asymmetrical steering forces [5]**

Because of its simplicity in assumption of velocity alignment, this simple vehicle model cannot simulate physical effects [7]. Furthermore this model allows the moving object to turn when its speed is zero. Real vehicles cannot do this and in any case they need undesirably large changes in orientation during a single time step. This problem can be solved by placing few additional constraint like a change of orientation [8, 9], or by limiting the lateral steering component at low speeds, or by simulating moment of inertia.

### 3. STEERING BEHAVIOUR

Now in context of steering behaviours lets assume that locomotion or movement is implemented by the simple vehicle model described above, and is parameterized by a single steering force vector. Therefore the steering behaviours are described in terms of the calculation of a vector [10]. Now we will discuss four different steering ways.

Seek [5] acts to steer or moves the Intelligent agent towards a specified position in any two dimensional space as shown in Figure 3. This behaviour adjusts the intelligent agent so that its velocity is practically aligned towards the target. Note that this is different from an attractive force which may produce an orbital path around the target point. The desired velocity is a vector in the direction from the agent to the target. The value of desired velocity could be maximum speed, or it could be the current speed, depending on the particular application. Flee [5] is simply the inverse of seek and acts to steer the agent away so that its velocity is aligned away from the target. The working can be seen in Formula Set 1.



**Figure 3: Seek and Flee [5]**

<p><b>Seek</b></p> <p>Desired Velocity = (Current Position – Target Position) * Maximum Speed</p> <p>Steering Force = Desired Velocity – Current Velocity</p> <p><b>Flee</b></p> <p>Desired Velocity = (Current Position – Target Position) * Maximum Speed</p> <p>Steering Force = Current Velocity - Desired Velocity</p>
---

**Formula Set 1: Seek and Flee**

Arrive behaviour [5] is identical to seek where the intelligent agent is far from its target, but instead of moving through the target at full speed, this behaviour causes the agent to slow down as it approaches the target, eventually slowing to a stop coincident with the target, as shown in Figure 4. Its implementation is similar to seek: a desired velocity is determined pointing from the intelligent agent towards the target. Outside the stopping radius desired velocity is set to maximum speed, inside stopping radius, desired velocity is brought to zero.

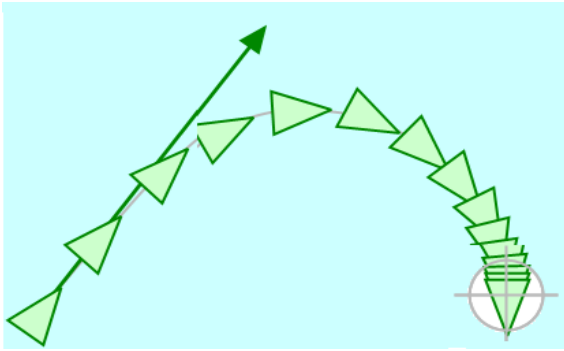


Figure 4: Arrive [5]

Wander [5] is a type of random steering as shown in Figure 5. One easy implementation is to generate a random steering force for each frame, but this produces rather simple motion. A more interesting approach would be to retain steering direction state and make small random displacements to it each frame. Thus for each frame it would have radically different behaviour [8]. The steering force takes a random run from one direction to another. The working can be seen in Formula Set 2.



Figure 5: Wander [5]

<p><b>Arrive</b></p> <p>Distance = Target Position – Current Position</p> <p>Change Speed = Maximum Speed *(Distance /Reducing Distance )</p> <p>Desired Velocity = (Changed Speed /distance) * Distance</p> <p>Steering Force = Desired Velocity – Current Velocity</p> <p><b>Wander</b></p> <p>Steering Force = Random Velocity &lt;= Maximum Velocity</p>
--

Formula Set 2: Arrive and Wander

#### 4. OBSERVATIONS

There is relationship between the mass and the speed of intelligent agents for optimum performance. The correlation is shown in Figure 6.

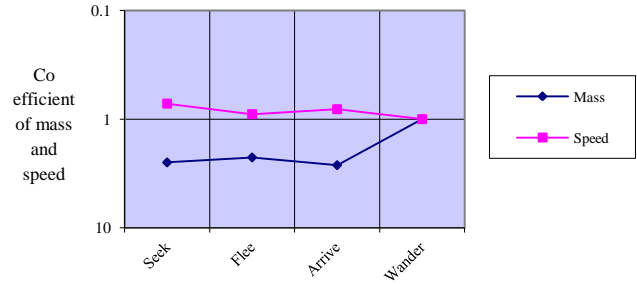


Figure 6: Relationship in qualities for optimum performance

The Table 1 shown below gives the generalized view of the desired qualities of the intelligent agents for better or optimum performance. The qualities specified are desired for the intelligent agent to reach its goal quickly and in more efficient manner. These qualities give the intelligent agent stability. As shown in the table some behaviour depends upon the mass and speed whereas wander behaviour does not.

Table 1: Desired qualities in intelligent agent for better performance.

Behaviour	Desired Qualities
Seek	More Mass, Less Speed
Flee	More Mass, Less Speed
Arrive	More Mass, Less Speed
Wander	Qualities does not affect

The behaviour of intelligent agents also depends upon environment like whether the display environment is mirrored or un-mirrored. Our observation is for un-mirrored environment. Figure 7 shows the correlation between mass, speed and time taken to complete the objective with the various behaviour of the intelligent agents.

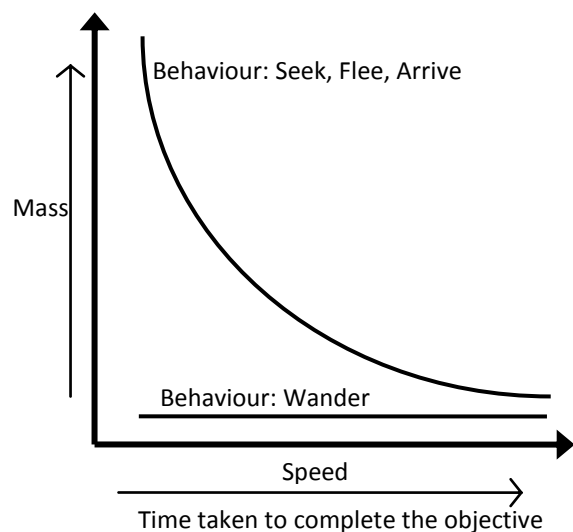


Figure 7: Correlation of various factors of intelligent agents in different behaviour.

The wander behaviour has no relation among its qualities where as other behaviour shows better performance with more mass and less speed. It should be observed that these observations are for un-mirrored environment. For mirrored environment these factors would change significantly. The different combination of qualities has different output behaviour or result in movement of intelligent agents.

## 5. CONCLUSION

This paper defines few common steering behaviour of computer simulated intelligent agents with improvisational action and path following. It presented a decomposition of constructing motion behaviours for intelligent agents into level of hierarchy. It has defined a minimal implementation of the locomotion level in terms of a 'simple vehicle model' which can be extended to robots. It has presented a collection of simple steering behaviours.

## 6. REFERENCES

- [1] Viorel Nicolau, "Modeling Aspects of Nonlinear Steering Machine of Conventional Ships", in Proceeding of Methods and Models in Automation and Robotics Conference, pages 848-851. August 2013.
- [2] Wen He, Guisheng Chen, Shuming Tang, Deyi Li, Mu Guo, Tianlei Zhang, Peng Jia and Feng Jin, "A Scaled-Down Traffic System based on Autonomous Vehicles: A New Experimental System for ITS Research", in Proceeding of Intelligent Transportation Systems Conference, pages 13-18. September 2012.
- [3] J. Miguel Leitao, A. Augusto Sousa and F. Nunes Ferreira, "Graphical Control of Autonomous, Virtual Vehicles", in Proceeding of IEEE Vehicular Technology Conference, pages 507-511. May 2000.
- [4] Hongling Wang, Joseph K.Kearney, James Cremer and Peter Willemsen, "Steering behaviors for autonomous vehicles in virtual environments", in Proceeding of IEEE Virtual Reality Conference, pages 155-162. March 2005.
- [5] Craig W. Reynolds, "Steering Behaviors for Autonomous Characters" in Proceeding of Game Developers Conference, pages 763-782. May 1999.
- [6] Bruce M. Blumberg and Tinsley A. Galyean, "Multi-Level Direction of Autonomous Creature for Real-Time Virtual Environments" in Proceeding of SIGGRAPH 95 Conference, pages 47-54. August 1995.
- [7] Alejandro Moreno Astorga, David Moreno-Salinas, Dictino Chaos García and Joaquín Aranda Almansa, "Simulation Benchmark for Autonomous Marine Vehicles in LabView" in Proceeding of OCEANS Conference, pages 1-6. June 2011.
- [8] Lyle N. Long, Scott D. Hanford, Oranuj Janrathitikarn, Greg L. Sinsley and Jodi A. Miller, "A Review of Intelligent Systems Software for Autonomous Vehicles" in Proceeding of Computational Intelligence in Security and Defence Applications Conference, pages 69-76. April 2007.
- [9] Jorge Estrela da Silva, "A dynamic programming approach for the control of autonomous vehicles on planar motion" in Proceeding of Autonomous and Intelligent Systems Conference, pages 1-6. June 2010.
- [10] D.K. Liu, X. Wu, A. K. Kulatunga and G. Dissanayake, "Motion Coordination of Multiple Autonomous Vehicles in Dynamic and Strictly Constrained Environments" in Proceeding of IEEE Cybernetics and Intelligent Systems Conference, pages 1-6. June 2006.