Proactive Steering Toward Oriented Targets

R. Boulic

Ecole Polytechnique Fédérale de Lausanne (EPFL), Virtual Reality Laboratory, Switzerland

Abstract

In this paper we introduce a real-time steering controller ensuring the reach of a (possible mobile) target position and orientation, without requiring to build/update the full trajectory to that target. We name it the funnelling control. The final orientation is achieved through the continuous adjustment of the heading direction. This control mode is proactive in the sense that it anticipates the success/failure of the reach and adjusts the desired speed accordingly. Both features rely on an heterogeneously sampled table of radial-tangential seek angles obtained when the controller reaches a desired position target without prescribed orientation. By construction, the control update has a constant computing cost, even with variable target characteristics. Its low update cost makes it particularly suited for controlling a large number of mobile entities in real-time. The present exposition is made for an obstacle-free context.

Categories and Subject Descriptors (according to ACM CCS): I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism---Animation; I.2.9 [Artificial Intelligence]: Robotics---Autonomous vehicles

1. Introduction

In this paper we address the problem of the real-time steering control of a mobile entity so that it reaches a target position with a prescribed orientation. The target may be mobile without incurring additional computing cost. One key aspect of our approach is its low cost as no trajectory needs to be established and sampled between the current and the desired state. It makes it particularly suited for real-time simulations with a large number of moving entities. The solution is described in the 2D for a mobile entity moving forward with bounded accelerations and speeds (forward and angular). We also limit the normal acceleration (product of linear and angular speeds) but not the curvature. We now briefly review the state of the Art to stress the main differences with prior method solving this kind of control task.

2. Background

There have been numerous works in mobile robot control addressing this type of problem. All approaches are structured in two stages: first building a trajectory enforcing desired mathematical properties over all the path to the target. For example, Dubins has established that the shortest oriented path with prescribed minimal radius of curvature R_{min} is C1 continuous and composed of line and circle arc segments of radius R_{min} [Dub 57]. Recent works are able to build analytic trajectories compatible with non-holonomic constraints of existing mobile robots [LL01],[FS04]. This first stage is generally coupled with a planning algorithm to solve the obstacle avoidance problem. The second stage consists in progressing on the resulting trajectory with proper sampling.

While Robotics can handle the real-time control of one mobile robot, Computer Animation generally targets animating potentially large numbers of characters with potentially mobile targets and obstacles. In that context, Reynolds has described in great details a panoply of local reactive steering behaviors covering most of the existing needs [Rey99]. One important class of characters is pedestrians; Helbings and Molnar have reproduced the motion of pedestrians groups in some well-known contexts (e.g. corridor) by subjecting them to “social forces” resulting from the combination of potential fields [HM95]. Metoyer and Hodgins have associated similar potential fields to user-supplied natural path and proposed a higher level management of pedestrian collision avoidance [MH04]. Brogan and Johnson have built a walking path model from measurements from which they construct a heading chart ensuring trajectories with minimal radius of curvature towards a goal [BJ03]. A global solution exploiting probabilistic roadmaps is described in Pettre et al. [PLS03]; the resulting path is modelled with piecewise Bezier splines. An intermediate approach, offering a good compromise between the reactivity to a dynamic environment and local minima avoidance, is proposed in Go et al. [GTK04].

Our approach belongs to the family of local reactive steering control from Reynolds as it does not have to build the trajectory to the target state. Our main contribution is to introduce a controller that also constrains the orientation at the target position; no intermediate point needs to be built between the current location and the target, whatever distant it is. One important aspect is the constant cost of the update even within a variable context (target position or orientation, tolerances, desired forward speed). In addition, the proposed control mode is proactive in the sense that it anticipates the success/failure of the reach and adjusts the desired speed accordingly.

In the remainder of the paper, we first provide key definitions in the seek mode (Section 3) prior to detail the funnelling control (Section 4). Section 5 presents various results while Section 6 concludes on foreseen future work.

3. The Seek mode

The seek behavior has been defined in Reynolds [Rey99] as the act to steer the character towards a specified position in global space. We adopt a similarly simple vehicle model with unit mass and unit yaw moment of inertia. In addition, the vehicle simulated throughout this paper is non-holonome. Our seek control is defined from the knowledge of the current vehicle state, especially its forward speed v_c and angular speed ω_c, and its current goal: the desired forward speed v_d, and the relative position of the target (Figure 1a). The two controllers exploited in the seek mode are generalized in Section 4 for the funnelling control.

- The angular speed controller drives the scalar angular acceleration aiming to *align the forward direction with the heading direction*. In the Seek mode the heading direction coincides with the instantaneous radial direction joining the current position of the target position (Figure 1b). The acceleration is proportional to the direction error α; a damping term smoothens high frequency variations.
- The forward speed controller has an implicit lower priority compared to the angular control; it controls the scalar forward acceleration aiming to achieve the desired forward speed v_d. This objective is *relaxed* automatically to a lower value v_{dr} so that the normal acceleration remains under a maximum value A. We have: $v_{dr} = \mathrm{Min}(v_d , A/\omega_c)$.

One limitation of such a control architecture, especially when used with a simple Euler integration scheme, is the possibility to converge to an infinite orbital trajectory around the target position. In such a case we consider the target as being *unreachable*. Our proposal for funnelling control solves this issue too, as explained in the next section. Figure 1c highlights the radial-tangential angle, noted η, which lies at the core of our funnelling control method. This is the angle made between the initial and the final radial direction for a given target position. It characterizes the controller in seek mode (i.e. reaching a static target position with unconstrained orientation while aiming to achieve a desired forward speed). Figure 2 displays the heterogeneous sampling of target positions we use to identify the η angle in the seek mode. It is completed with a sampling of the initial forward speed, the initial angular speed and the desired forward speed. The resulting table $T\eta$ contains 43200 elements stating whether each sampled target is reachable, and if it is the case, the η angle, otherwise we store the closest desired speed ensuring reachability or zero if there is none. The table preprocessing time is about 5 minutes on a Pentium IVm 2GHz. It can be exploited by an arbitrary number of simultaneously controlled mobile entities. Its first immediate benefit is to make the seek mode capable of adjusting the desired speed so that the target becomes reachable in all cases. The principle is the following: at each time step, the desired forward speed is replaced by the value stored in the table whenever the target, expressed in local coordinates, appears as unreachable.

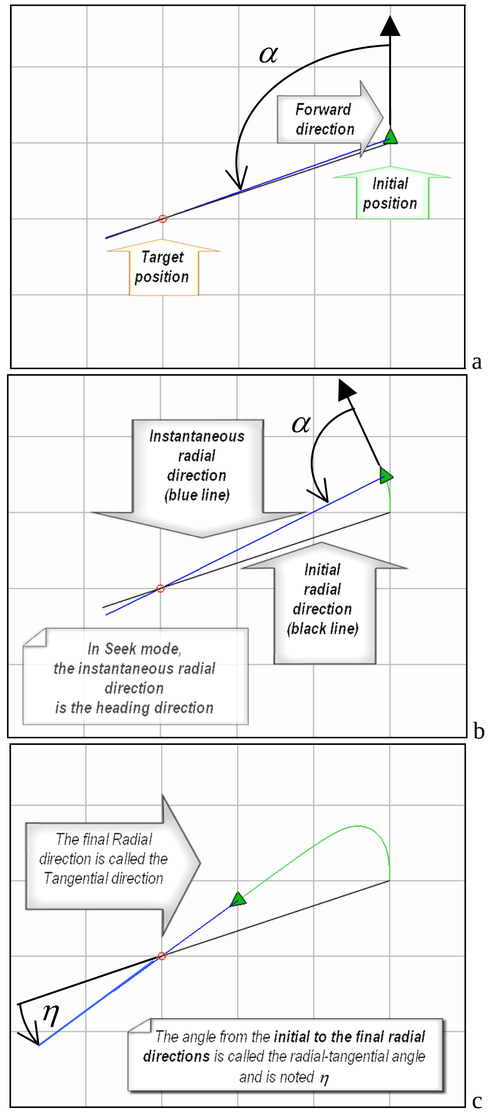

Figure 1: *Definition of the η angle in Seek mode*

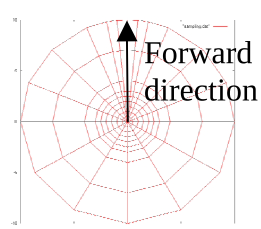

Figure 2: *Heterogeneous sampling of the position targets in seek mode for which we evaluate the η angle.*

4. Principle of the Funnelling Control

Our proposal for achieving a target orientation at the target position (shown in red in Figure 3) is to exploit the error $\Delta\eta$ between the target direction and the current tangential direction built from the current η angle (interpolated from Tη). When $\Delta\eta$ is null, we are back in the seek mode where the heading direction coincides with the radial direction. Otherwise, the heading direction is rotated around the radial direction by an amount μ (Figure 3a). The angular acceleration is computed with the same controller as in the seek mode but as the heading direction is not always the radial direction, the resulting trajectory now ensures reaching both the target position and orientation (Figure 3b).

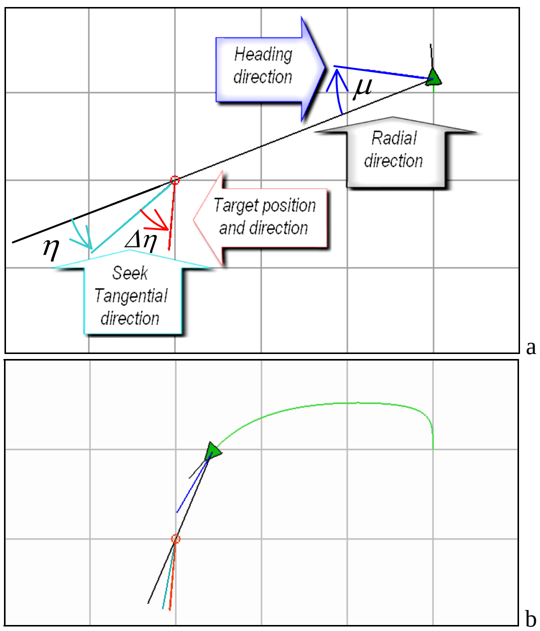

Figure 3: *Definition of $\Delta\eta$ and μ (a), resulting trajectory toward the position and orientation target (b)*

One funnelling update step is summarized below:

```
Update relative distance D
Update radial angle α
If closest Tη element appears unreachable
  Get the stored desired forward speed v_dt
Interpolate η in Tη for(D,α, ω_c , v_c, v_d)
Build the corresp. tangential direction
Update Δη as the error between the target
  direction and the tangential direction
Bound  μ=  -Δη with  a  Maximum  amplitude
  interpolated in Tη for(D, Δη,0, v_c, v_dt)
Feed  the  seek  controllers  with  the  new
  heading   direction   and   the   temporary
  desired linear speed = Min(v_dt, A/ω_c).
```

Figure 4: *Funnelling update step algorithm*

Although simple, the relation linking μ to $\Delta\eta$ has to be bounded to prevent divergence for large values of $\Delta\eta$. We found that the retained upper bound (Figure 4) posesses the essential quality of converging to zero when D grows to infinity. One interesting consequence of this choice is an assymptotic seek behavior for distant targets.

5. Control mode relaxation near the target

In the close neighborhood of the target, both the angles α and η may vary wildly from one iteration to the next, resulting in high accelerations and curvatures. For this reason, we prefer a gracefully degraded solution to an unaesthetic exact solution resulting in a last second U_turn… When such a context is detected, we propose to achieve the position target and the orientation targets *sequentially* instead of simultaneously. The pseudo-code from Figure 5 indicates the variables and thresholds used to make the choice of the current control mode, noted M_c, among: Position_only (seek), Orientation_only (regardless of a target position), Funnelling, or End_of_Steering.

```
M_c := Funnelling
If(D < D_tol  AND (α+η+Δη)< ε_tol)
        M_c:= End_of_Steering
Else if(D < D_tol)
        M_c:= Orientation_only
Else if(D < D_η OR (α+η+Δη)< ε_tol)
        M_c:= Position_only
```

Figure 5: *Choice of the current control mode M_c with D being the current radial distance, D_η, the minimal η sampling distance, D_{tol} the position tolerance and ε_{tol} the orientation tolerance.*

6. Results

The accompanying video details the cases from Figures 1 and 3 and three more cases: dynamic target with halting, successive targets with long distances, performance case. The real-time video recording has been made with Camtasia TM, hence reducing the frame rate and introducing a few animation glitches. In all examples, the current control mode is translated graphically through the target design as follow: a red circle in seek mode, a blue triangle in orientation_only mode, and a red circle and triangle in funnelling mode. The circle radius (resp. the triangle shape) expresses the position (resp. orientation) tolerance.

In the dynamic target case (Figure 6) pairs of mobile entities are attracted towards each other so as to meet face to face. Such scenario occurs in real life when two persons suddenly recognize each other and change their path so as to exchange salutations such as shaking hands. Other purposes requesting a face to face relative orientation can easily be imagined (fight, deliver a parcel, etc…). This context is very expensive for alternate approaches that would have to update and sample the full trajectory at each time-step

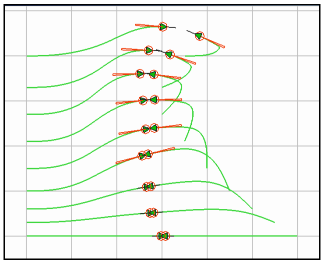

Figure 6: *Dynamic targets with final halting*

The next case display trajectories obtained for a wide variety of relative orientations and distances to (successive) targets. The grid size being one unit - as for all other examples - it is interesting to notice that for distant targets the behavior is first similar to a seek trajectory prior to smoothly starting to enforce the final orientation goal when arriving closer to the target (Figure 7).

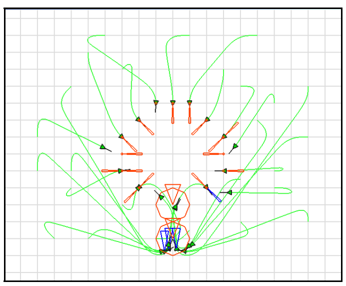

Figure 7: *A wide variety of cases with final halting*

The last case highlights the performances and the quality of the resulting traces (a 500 points polyline is drawn for each mobile target). Here too the trace bundles left for long distance targets show three sections: the initial turning to point toward the direction of the target, as if in pure seek mode, followed by a quasi straight line section that gradually turns into the last curved section enforcing the desired orientation (Figure 8).

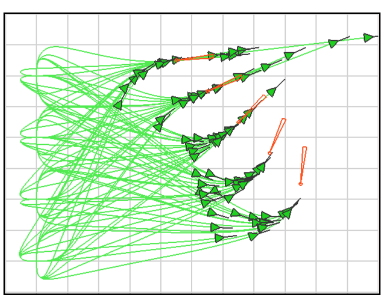

Figure 8: *80 mobile entities are assigned to five targets*

7. Discussion and future work

The control update cost is constant; it includes in the worst case two interpolations of the η angle. Measurements give an average value of 12.5µs per control update (without display) on a Pentium IVm 2GHz. This is a maximum as for a distant target (30 units) the average control update cost drops to 9.5µs. The proposed steering control successfully handles dynamic targets as shown in the shake-hand scenario. Besides, the reachability boolean and the interpolated η value could be exploited by a higher level of planning as a quantitative indication about the achievability of the current target. Our approach can be combined with other local steering behavior from Reynold [Rey99] to handle 3D vehicles moving on a 2D terrain movements or for avoiding obstacle (e.g. by combining our obstacle-free velocity with the velocity resulting from Reynold obstacle avoidance steering force).We plan to refine this approach by predicting the near-future trajectory, hence solving highly dynamic and crowded cases where planning finds its computational limits.

8. Acknowledgement

Many thanks to Achille Peternier for the Mvisio viewer.

9. References

[BJ03] Brogan, D. C., Johnson N. L. 2003. Realistic Human Walking Paths. In *Proc. of Computer Animation and Social Agents , 94-101.*

[Dub57] Dubins, L. E. 1957, On curves of minimal length with a constrain on average curvature and with prescribed initial and terminal positions and tangents, *Amer. J. Math*., 497–516.

[FS04] Fraichard, T., Scheuer A. 2004, From Reeds and Shepp's to Continuous-Curvature Paths, *IEEE TRA*, 20(6), 1025-1035.

[GTK04] Go, J., Thuc, V., Kuffner, J. J. 2004. Autonomous Behaviors For Interactive Vehicle Animations. In *Proc. of Eurographics/ACM SIGGRAPH Symposium on Computer Animation, 9-18.*

[HM95] Helbings, D., Molnar P. C. 1995, Social force model for pedestrian dynamics, *Physical Review E*, 51(5), 4282-428.

[LL01] Lamiraux, F., Laumond, J.-P. 2001, Smooth Motion Planning for Car-Like Vehicle, *IEEE TRA*, 17(4), 498-502.

[MH04] Metoyer, R. A., Hodgins, J. K. 2004. Reactive Pedestrian Navigation from Examples. *The Visual Computer*, 20(10),635-649.

[PL03] Pettre, J., Laumond J.-P., Simeon T. 2003. A 2-Stages Locomotion Planner for Digital Actors". In *Proc. of Eurographics/ACM SIGGRAPH Symposium on Computer Animation, 258-264.*

[Rey99] Reynolds, C. W. 1999. Steering Behaviors For Autonomous Characters. In *Proc. 1999 Game Developer's Conference, 763-782.*