Technical Note: Determining the Closest Approaches between Two Piece-wise Linear Routes

Stephen Fitzpatrick
Kestrel Institute

August 2019

Abstract

This report details how to determine the points of closest approach of two objects, each moving along a piece-wise linear route.

1 Introduction

The motion of an object may be specified in terms of a *route*, that is, a sequence of *way points*, in which each way point provides a *position in space* (in arbitrary dimensions) and a *time* (with time strictly increasing between successive way points). The object has a constant velocity between successive way points, and thus moves along a sequence of straight line segments between successive way points. For example, Table 1 shows a 2-dimensional route with way points at times 0, 10, 20 & 60.

Time	Position
0	(0, 0)
10	(2, 2)
20	(4, 2)
60	(0, 0)

Table 1: A route as a sequence of way points

Given two objects, each moving along its own route, the distance between the objects typically varies over time. (The distance between two positions is assumed to be *Euclidean* — that is, the square root of the sum of the squares of the differences along each dimension of the objects' positions.)

This report details how to determine the positions and times at which the distance between the objects is minimized as they progress along their routes — that is, the routes' *closest approaches*. Note that this is the *instantaneous* distance that is minimized; an alternative is the smallest distance between the all of the positions covered by one route, without regard to time, and all of the positions covered by the other route, without regard to time. The latter is not considered in this note.

First, the closest approach is determined for two objects, each moving along a single straight line segment, with the two motions having common start times and common end times. This is then generalized to piece-wise linear routes.

2 Closest Approach between Two Objects Moving Along Straight Line Segments with Common Start Times and Common End Times

Given an object moving with constant velocity from starting position $\vec{s}$ at time t_s to ending position $\vec{e}$ at time $t_e > t_s$, the object's position at some intermediate time t ($t_s \leq t \leq t_e$) is given by:

$$p(k) = \vec{s} + k(\vec{e} - \vec{s}) \tag{1}$$

where $k = (t - t_s)/(t_e - t_s)$, the *interpolation parameter*, is the fraction of the route covered up to time t.

Note that $k = 0$ at $t = t_s$, $0 < k < 1$ for $t_s < t < t_e$, and $k = 1$ at $t = t_e$. That is, k has value 0 for the start of the motion, 1 for the end of the motion, and intermediate values in between. Since k is linear in t, the object's motion is linear in k.

2.1 Distance between Objects

For two objects, with starting positions $\vec{s_i}$ and ending positions $\vec{e_i}$, for $i = 1$ and $i = 2$, and with common starting time t_s and common ending time t_e, the difference in the objects' positions at a given k ($0 \leq k \leq 1$) is:

$$\begin{aligned} \Delta\vec{p}(k) &= \vec{p_1}(k) - \vec{p_2}(k) \\ &= \vec{s_1} + k(\vec{e_1} - \vec{s_1}) - \vec{s_2} - k(\vec{e_2} - \vec{s_2}) \\ &= (\vec{s_1} - \vec{s_2}) + k[(\vec{e_1} - \vec{e_2}) - (\vec{s_1} - \vec{s_2})] \\ &= \Delta\vec{s} + k(\Delta\vec{e} - \Delta\vec{s}) \end{aligned}$$

where $\Delta\vec{s} = \vec{s_1} - \vec{s_2}$ is the (vector) difference between the starting points, and $\Delta\vec{e} = \vec{e_1} - \vec{e_2}$ is the (vector) difference between the ending points.

Note that $\Delta\vec{s} + k(\Delta\vec{e} - \Delta\vec{s})$ is the interpolation, by k, between the objects' initial separation, $\Delta\vec{s}$, and their final separation, $\Delta\vec{e}$.

If the two objects are moving in parallel, then the difference in their positions is constant. In particular, $\Delta\vec{e} - \Delta\vec{s} = \vec{0}$, so $\Delta\vec{p}(k)$ is constant, with value $\Delta\vec{s}$.

The *distance* between the objects, $D(k)$, is

$$D(k) = ||\Delta\vec{p}(k)|| \tag{2}$$

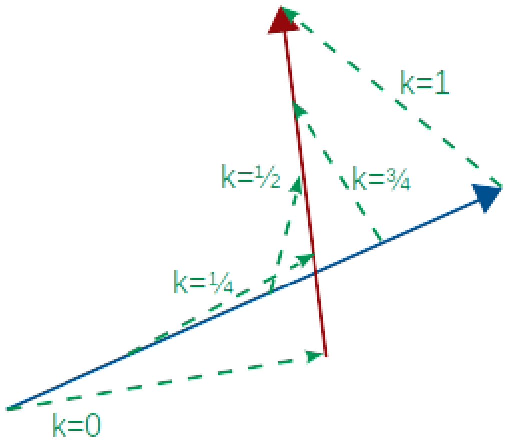

Figure 1: Closest approach between two objects moving in straight lines

where $||v|| = \sqrt{\sum_{j=1}^{n}(v_j)^2}$ is the 2-norm of the vector $\vec{v}$, where $\vec{v}$ has n dimensions, and v_j denotes the j^{th} dimension of $\vec{v}$ (for $j = 1, \ldots, n$).

For example, Figure 1 shows the movements of two objects:

- One object follows the blue line, starting at the left end and moving right.
- The second object follows the red line, starting at the bottom and moving up.
- The green, dashed lines show the offsets between the two objects for various values of k.

Figure 2 shows how the distance between the objects varies as k varies. Note $D(k)$ is the instantaneous distance between the objects (k being linear in t) — this is the distance for which the minimum is to be found.

In contrast, the atemporal distance can be defined as the minimum distance between any position covered by one route and any position covered by the other route:

$$\min_{p_1 \in P_1, p_2 \in P_2} ||p_1 - p_2||$$

where $P_i = \{p_i(k) \mid 0 \leq k \leq 1\}$, for $i = 1$ or $i = 2$, is the set of all positions covered by route i.

In Figure 1, the instantaneous distance has a minimum at $k \approx 0.65$, with a value of 0.30, whereas the atemporal distance between the two routes is 0 since they cross. The crossing point is *not* the closest approach because the two objects are not at the crossing point at the same time.

2.2 Computing the Minimum Distance

Minimizing the inter-object distance can be achieved by minimizing the square of the distance, which is simpler.

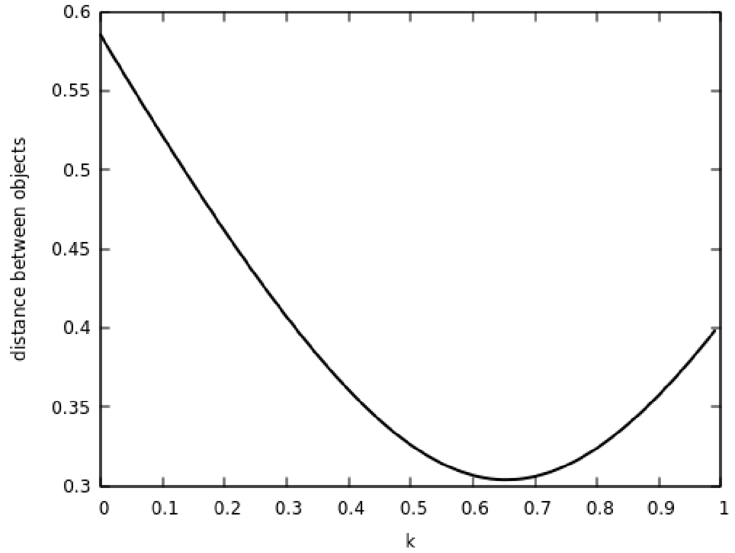

Figure 2: Distance between objects as a function of k; minimum is at $k \approx 0.65$

The square of the distance is:

$$\begin{aligned} D^2(k) &= ||\Delta\vec{p}(k)||^2 \\ &= \textstyle\sum_{j=1}^{n}[\Delta s_j + k(\Delta e_j - \Delta s_j)]^2 \ . \end{aligned}$$

Differentiating with respect to k:

$$\frac{\partial D^2(k)}{\partial k} = \sum_{j=1}^{n} 2[\Delta s_j + k(\Delta e_j - \Delta s_j)](\Delta e_j - \Delta s_j) \ .$$

At the minimum, the derivative is 0, so:

$$\sum_{j=1}^{n} k(\Delta e_j - \Delta s_j)(\Delta e_j - \Delta s_j) = -\sum_{j=1}^{n} \Delta s_j(\Delta e_j - \Delta s_j)$$

which gives k for the closest approach as:

$$k = \frac{\sum_{j=1}^{n} \Delta s_j(\Delta s_j - \Delta e_j)}{\sum_{j=1}^{n}(\Delta s_j - \Delta e_j)^2} = \frac{\Delta\vec{s}.(\Delta\vec{s} - \Delta\vec{e})}{||\Delta\vec{s} - \Delta\vec{e}||^2}$$

where $\vec{u}.\vec{v} = \sum_{j=1}^{n} u_j v_j$ is the inner product of vectors $\vec{u}$ and $\vec{v}$.

Note that $||\Delta\vec{s} - \Delta\vec{e}||^2$ is the squared length of $\Delta\vec{s} - \Delta\vec{e}$. If this is 0, then the difference in the starting positions is the same as the difference in the ending positions; i.e., the objects are moving in parallel. In this case, k is undefined by the above equation — the distance is the same for all k ($0 \leq k \leq 1$). If only the distance of closest approach is required (and not the positions or time), then the distance can be computed from, say, the starting positions or the ending positions.

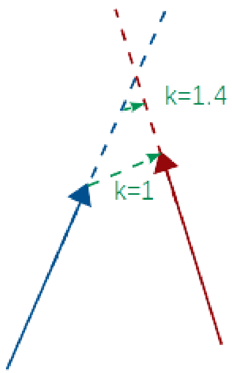

Figure 3: Closest approach is outside the time interval

Also note that the above equation may give $k < 0$ or $k > 1$. In such cases, if the objects' straight-line motions were to be extended before t_s or after t_e, respectively, then the closest approach would occur in the extension. For example, Figure 3 shows two motions for which, if they were extended, $k \approx 1.4$ would give the closest approach.

If the equation gives $k < 0$ then the closest approach within the time period $[t_s, t_e]$ occurs at t_s; i.e., at $k = 0$. Likewise, if $k > 1$, the closest approach within $[t_s, t_e]$ occurs at t_e; i.e., at $k = 1$.

Thus, the closest approach within $[t_s, t_e]$ is given by:

$$k = \max\left(0, \min\left(1, \frac{\Delta\vec{s}.(\Delta\vec{s} - \Delta\vec{e})}{||\Delta\vec{s} - \Delta\vec{e}||^2}\right)\right) \quad . \tag{3}$$

Once k has been determined for the closest approach, the position of each object at closest approach can be determined using Equation 1, and thence the inter-object distance at closest approach using Equation 2.

3 Closest Approaches along Piece-wise Linear Routes

Equation 3 applies to two motions, each of which is a single straight-line segment; the two motions must have a common start time and a common end time.

A route comprises a sequence of straight-line segments, so, given two routes, R_1 and R_2, Equation 3 potentially could be applied to the first segment in R_1 and the first segment in R_2, then to the second segment in R_1 and the second segment in R_2, and so on, to determine the closest approaches segment by segment.

However, in general, the j^{th} segment in R_1 will not have the same start time, nor the same end time, as the j^{th} segment in R_2. Indeed, in general, R_1 and R_2 will not have the same number of segments. For example, Figure 2 shows two routes, one with way points at times 0, 10, 20 & 60; the other with way points at times 0, 20 & 40.

Time	Route 1 Position
0	(0, 0)
10	(2, 2)
20	(4, 2)
60	(0, 0)

Time	Route 2 Position
0	(2, 8)
20	(0, 6)
40	(4, 2)

Table 2: Two routes with unaligned times

Time	Route 1 Position	Route 2 Position
0	(0, 0)	(2, 8)
10	(2, 2)	**(1, 7)**
20	(4, 2)	(0, 6)
40	**(2, 1)**	(4, 2)
60	(0, 0)	

Table 3: Two routes with aligned times

To apply Equation 3, the routes need to be *aligned* in time, so that: (1) they have the same number of way points; and (2) the j^{th} way points in each have the same time.

For example, Figure 3 shows the same two routes after alignment:

- For Route 1, the straight line segment from position (4, 2) at time 20 to position (0, 0) a time 60, was split into two segments by inserting a way point at time 40 at position (2, 1), determined by interpolating the segment to time 40.
- Likewise, for Route 2, the segment from (2, 8) at time 0 to (0, 6) at time 20 was split by adding a new way point at time 10, with position (1, 7).

In both cases, the position of the new way point was determined by interpolating the original route to the appropriate time — thus, the insertion of the way point does not change the motion of the object following the route.

Note that a way point at time 60 is *not* inserted into Route 2 since Route 2 finishes at time 40. The final way point in Route 1 is discarded for the purpose of determining closest approaches.

The following section provides methods for aligning the routes and determining the closest approaches. Note that, in theory, there may be multiple closest approaches, with the same distance.

3.1 Aligning Times and Determining Closest Approaches

First note that a straight-line movement from a way point with starting position $\vec{s}$ at time t_s and ending position $\vec{e}$ at time t_e can be broken into two consecutive movements by introducing an intermediate way point at any time τ between the starting and end times, with corresponding position given by interpolation between the starting and ending positions.

That is, the movement $(t_s, \vec{s}) \to (t_e, \vec{e})$ can be broken into the movements $(t_s, \vec{s}) \to (\tau, \vec{p}(\tau))$ and $(\tau, \vec{p}(\tau)) \to (t_e, \vec{e})$, where $t_s < \tau < t_e$ and $p(\tau) = \vec{s} + k(\vec{e} - \vec{s})$ where $k = (\tau - t_s)/(t_e - t_s)$.

Now assume initially that the two routes start at the same time and end at the same time. Then a sequence of increasing times τ_l, $1 \leq l \leq m$ for some m, needs to be determined such that τ_1 is the common start time, τ_m is the common end time, and for each successive pair of times τ_l and τ_{l+1}, $1 \leq l < m$, each object has a uniform velocity over the period $[\tau_l, \tau_{l+1}]$. Then Equation 3 can be used to determine the closest approach that occurs within each $[\tau_l, \tau_{l+1}]$, and the overall closest approaches determined by finding those that have the smallest separation.

		Route 1		Route 2		Closest Approach			
t_s	t_e	s_1	e_1	s_2	e_2	Time	Route 1	Route 2	Distance
0	10	(0, 0)	(2, 2)	(2, 8)	(1, 7)	10	(2.0, 2.0)	(1.0, 7.0)	5.10
10	20	(2, 2)	(4, 2)	(1, 7)	(0, 6)	12	(2.4, 2.0)	(0.8, 6.8)	5.06
20	40	(4, 2)	(2, 1)	(0, 6)	(4, 2)	36	(2.4, 1.2)	(3.2, 2.8)	1.79

Table 4: Closest approaches for corresponding segments of aligned routes

For example, Table 4 shows corresponding segments of aligned routes, with the closest approach for each segment. The rightmost columns of the table show the time at which the closest approach occurs in the segment, the way point in Route 1 for closest approach, the way point in Route 2 for closest approach, and the separation at closest approach. The overall closest approach thus occurs in the final segment (between times 20 and 40).

If the objects' routes do not start at the same time, then any motion prior to the later start time is ignored. Likewise, any motion after the earlier end time is ignored. In other words, the objects' separation is only considered over the common period of their two routes. If there is no common period (i.e., one route ends before the other begins), then there is no closest approach.

The algorithm for aligning two routes in time is detailed in Figure 4. Determining the closest approaches between two routes is detailed in Figure 5.

4 Conclusion

This technical note details an algorithm for determining the times, positions and distance of closest approaches between two piece-wise linear routes.

1. Given routes R_1 and R_2.
2. Determine the common time period $[s, e]$ for the two routes: $s = \max(s_1, s_2)$ where s_i is the start time of Route R_i; $e = \min(e_1, e_2)$ where e_i is the end time of Route R_i.
3. If there is no common period (i.e., $e < s$), then there is no closest approach and the algorithm is complete. The aligned routes are both empty. (Note that if one route is empty, so is the other.)
4. Otherwise, let T_i be the set of times of R_i's way points that fall within $[s, e]$.
5. Let T be the sequence of times that contains, in ascending order, the time s, every time in T_1, every time in T_2, and the time e, with each time occurring at most once (i.e., $T = \mathrm{sort}(\{s\} \cup T_1 \cup T_2 \cup \{e\})$).
6. Let R'_i be the route formed by interpolating R_i at each time in T. That is, the j^{th} way point in R'_i has time T_j, and position given by interpolating R_i to time T_j.
7. The aligned routes are R'_1 and R'_2.

Figure 4: Algorithm for aligning two routes in time

1. Given two routes R_1 and R_2.
2. Determine time-aligned routes R'_1 and R'_2.
3. If both aligned routes are empty, there is no closest approach. The algorithm is complete.
4. Otherwise, if either aligned route has exactly one way point, then that way point determines the unique closest approach. (This can happen only when one route's end time is the same as the other route's start time.)

   Let t be the time of that way point — note that the other route must also have a way point at time t, since the routes are aligned in time. The positions of these two way points, at time t, are the positions of the closest approach, and their separation gives the distance of closest approach. The algorithm is complete.
5. Otherwise, both aligned routes have at least two way points. Let T denote the sequence of common times of the aligned routes, in ascending order.
6. Determine a sequence C_l of candidate closest approaches by considering each consecutive pair of times in T, as follows:
   (a) Let t_s be the earlier of the pair of times. Let t_e be the later of the pair of times.
   (b) Let s_i be the position in Route R'_i at time t_s.
   (c) Let e_i be the position in Route R'_i at time t_e.
   (d) Determine the closest approach for the straight-line movements given by t_s, s_i, t_e and e_i, using Equation 3. If the movements are parallel (or effectively so given finite precision computations), then take the closest approach as occurring at the midpoint; i.e., at $t = (t_s + t_e)/2$.
7. Determine the candidate closest approaches that have the smallest inter-object separation. These are the closest approaches between the two routes.

Figure 5: Algorithm for determining closest approaches