Following trajectory

Idea of algorithm:
The algorithm gets n points and a current position as input and provides the position of the next point, relative to the current position as output, which the vehicle is supposed to drive. For this purpose the algorithm calculates the nearest point of the trajectory. If the calculated point is closer than a certain distance δ then the vehicle is either close enough to the planned trajectory or it has reached a node of the trajectory, so it drives to the next point in the trajectory.

Derivation:
In: $(x, y)^n$ is the trajectory,
In: $(c_x.c_y)$ is the current position of the vehicle,
Out: (r_x, r_y) is the relative position of the next point to be driven.

Calculate (d_{ix}, d_{iy}) $\forall i \in \{1, ..., n-1\}$, where (d_{ix}, d_{iy}) describes the closest point of the straight line between the points (x_i, y_i) and (x_{i+1}, y_{i+1}) and is calculated as follows:

Let g be the straight line that goes through (x_i, y_i) and (x_{i+1}, y_{i+1}) which is:

$$g(t) = \begin{pmatrix} x_i \\ y_i \end{pmatrix} + t \cdot \begin{pmatrix} x_{i+1} - x_i \\ y_{i+1} - y_i \end{pmatrix} \text{ with } t \in \mathbb{R}$$

Now calculate the normal vector $\begin{pmatrix} b_x \\ b_y \end{pmatrix}$ to this straight line:

$(x_{i+1} - x_i) \cdot b_x + (y_{i+1} - y_i) \cdot b_y = 0$. Therefore if $x_{i+1} - x_i \neq 0$:
$b_x = -\frac{y_{i+1} - y_i}{x_{i+1} - x_i} \cdot b_y$ and we can choose b_y freely except 0, for example 1.

Now we need to find the intersection between the straight line defined by the current position and the normal vector, and g:

$$\begin{pmatrix} c_x \\ c_y \end{pmatrix} + s \cdot \begin{pmatrix} b_x \\ b_y \end{pmatrix} = \begin{pmatrix} x_i \\ y_i \end{pmatrix} + t \cdot \begin{pmatrix} x_{i+1} - x_i \\ y_{i+1} - y_i \end{pmatrix}, \text{ with } s, t \in \mathbb{R}.$$

Therefore $c_x + s \cdot b_x = x_i + t \cdot (x_{i+1} - x_i)$ gives us $t = \frac{c_x + sb_x - x_i}{x_{i+1} - x_i}$.

$\Rightarrow c_y + sb_y = y_i + \frac{y_{i+1} - y_i}{x_{i+1} - x_1} \cdot (c_x + sb_x - x_i)$
$\Rightarrow s \cdot b_y - s \cdot b_x \frac{y_{i+1} - y_i}{x_{i+1} - x_i} = y_i + \frac{y_{i+1} - y_i}{x_{i+1} - x_i} \cdot (c_x - x_i) - c_y$
$\Rightarrow s = \frac{y_i + \frac{y_{i+1} - y_i}{x_{i+1} - x_i} \cdot (c_x - x_i) - c_y}{b_y - b_x \cdot \frac{y_{i+1} - y_i}{x_{i+1} - x_i}}$
And with s, t can be obtained.
If $x_{i+1} - x_i = 0$ we can assume that $y_{i+1} - y_i \neq 0$ since we always expect different points from the trajectory. We then get:
$b_y = \frac{-(x_{i+1} - x_i) \cdot b_x}{y_{i+1} - y_i} = 0$ and we can chose b_x freely.
We then get $s = -c_x + x_i$ and therefore $t = \frac{c_y - y_i}{y_{i+1} - y_i}$

If $t \in [0, 1]$ then $(d_{ix}, d_{iy}) = g(t)$ with above calculated t.
Otherwise the closest calculated point is out of scope and the closest point within scope is either (x_i, y_i) or (x_{i+1}, y_{i+1}), therefore (d_{ix}, d_{iy}) is the one that is closer to the current position (c_x, c_y).

Let (d_{ax}, d_{ay}) be the closest to (c_x, c_y) out of all (d_{ix}, d_{iy}) with $i \in \{1, ..., n-1\}$.
If $\|(d_{ax}, d_{ay}) - (c_x, c_y)\| > \delta$ then $(r_x, r_y) = (d_{ax}, d_{ay}) - (c_x, c_y)$ where δ is a constant which represents the maximum distance from a point so that it is still counted as reached. Otherwise $(r_x, r_y) = (x_{a+i}, y_{a+i}) - (c_x, c_y)$ with $i = \arg\min_{i \in \{a+1, \ldots, n\}} \{(x_{a+1}, y_{a+1}) - (c_x, c_y) > \delta\}$.