Two Axle Vehicle

A.J. de Graaf - Art in Control

June 23, 2017

Abstract

Essential and convenient formulae that relate quantities for this type of vehicle are introduced and explained. Several topics are introduced to be integrated in the analysis document in a later stage.

Contents

1 Introduction

It is desired to obtain a set of formulae that can be used to relate the quantities of the configuration space of this particular vehicle to the pose and twist of a vehicle attached coordinate system. Moreover, a definition of the scalar speed of the vehicle is introduced which has a useful sensation of both translation and rotation velocity.

A set of distinct points with associated body-attached coordinate systems is used to establish relations with the vehicle configuration space. These points are listed in the next table: These points will be used to define how the vehicle

point	frame	description
V	Ψ_v	vehicle origin with body-attached coordinate system
F	Ψ_f	centre of front steering axle with associated frame
A	Ψ_a	centre of aft steering axle with associated frame
P	Ψ_p	virtual point for parallel steering coordinate system
S	Ψ_s	virtual point for fictive steering wheel frame

performs movements and how it follows a trajectory. Some of these points (V, F, A) will be used to define associated points on the trajectory. These points will be primed (V', F', A').

It should be noted that in practice only an estimate is known of these points and associated frames. Proper symbols will be introduced to highlight this in the used notation.

2 Definitions

A few essential definitions are required to introduce the vehicle configuration space, vehicle pose and twist[1]. The vehicle configuration space is defined as:

$$q = (x_v, y_v, \phi_v, \delta_f, \delta_a)^T = ({}^0x_v, {}^0y_v, {}^0\phi_v, {}^v\phi_f, {}^v\phi_a)^T \qquad (1)$$

where the next coordinate systems or frames are used: Ψ_0, Ψ_v, Ψ_f, Ψ_a.

[1] see also: Two-Dimensional Robotics, A.J. de Graaf, July, 2012

The pose of the vehicle is defined as:

$$p = {}^0p_v = (x_v, y_v, \phi_v)^T \tag{2}$$

which is a subset of the vehicle configuration space. The steer angles are combined in vector $r = (\delta_f, \delta_a)^T$ resulting in the next short expression for the configuration space:

$$q = \begin{pmatrix} p \\ r \end{pmatrix} \tag{3}$$

The twist of the vehicle is defined as:

$$\dot{p} = t_v = {}^0\left(\frac{d^0p_v}{dt}\right) = {}^0\left({}^0\dot{p}_v\right) = (\dot{x}_v, \dot{y}_v, \dot{\phi}_v)^T \tag{4}$$

which represents the combination of translation velocity (vector) and rotation velocity (scalar).

In several occasions the twist of the vehicle will be represented in vehicle coordinates as its use is more convenient:

$${}^v\dot{p} = {}^vt_v = ({}^v\dot{x}_v, {}^v\dot{y}_v, {}^v\dot{\phi}_v)^T \tag{5}$$

The speed of the vehicle is defined implicitly by the next relation:

$$\dot{s}^2 = \dot{x_v}^2 + \dot{y_v}^2 + (d \cdot \dot{\phi_v})^2 \tag{6}$$

or in its equivalent form using vehicle coordinates:

$$\dot{s}^2 = {}^v\dot{x_v}^2 + {}^v\dot{y_v}^2 + (d \cdot {}^v\dot{\phi_v})^2 \tag{7}$$

where d is a scaling parameter to increase the sensation of rotation.

2.1 Shorthands

The shorthands x, y and ϕ will be used for x_v, y_v and ϕ_v respectively whenever it is convenient and unambiguous in its context.

The shorthands $\dot{x}$, $\dot{y}$ and $\dot{\phi}$ will be used for ${}^v\dot{x}_v$, ${}^v\dot{y}_v$ and ${}^v\dot{\phi}_v$ respectively whenever it is convenient and unambiguous in its context. This simplifies the definition of vehicle speed based on vehicle coordinates:

$$\dot{s}^2 = \dot{x}^2 + \dot{y}^2 + (d\dot{\phi})^2 \tag{8}$$

3 Kinematic relations

The relations between vehicle twist, speed and steer angles are needed to support the change of representation. These relations are derived for kinematic steering conditions (i.e. no weel slip angles)

First, the speed of the center of the two steering bodies is introduced:

$$\dot{x} = \dot{s}_f \cos\delta_f = \dot{s}_a \cos\delta_a \tag{9}$$

$$\dot{y} = \frac{\dot{s}_f \sin\delta_f + \dot{s}_a \sin\delta_a}{2} \tag{10}$$

$$\dot{\phi} = \frac{\dot{s}_f \sin\delta_f - \dot{s}_a \sin\delta_a}{2d} \tag{11}$$

where $\dot{s}_f$ and $\dot{s}_a$ correspond to ${}^f\dot{x}_f$ and ${}^a\dot{x}_a$ respectively. In a similar fashion, $\dot{s}_p$ and $\dot{s}_s$ correspond to ${}^p\dot{x}_p$ and ${}^s\dot{x}_s$ associated to the frames Ψ_p and Ψ_s.

Analogous to the scalar vehicle speed, implicit relations can be derived for the scalar speed of front and rear steering axle:

$$\dot{s}_f^2 = \dot{x}^2 + (\dot{y} + {}^v x_f \dot{\phi})^2 \tag{12}$$
$$\dot{s}_a^2 = \dot{x}^2 + (\dot{y} + {}^v x_a \dot{\phi})^2 \tag{13}$$

If the parameter d is chosen to satisfy the next constraint:

$$d = {}^v x_f = -{}^v x_a \tag{14}$$

and the steer angles comply to the next relations:

$$\delta_f = \arctan \frac{\dot{y} + {}^v x_f \dot{\phi}}{\dot{x}} \tag{15}$$
$$\delta_a = \arctan \frac{\dot{y} + {}^v x_a \dot{\phi}}{\dot{x}} \tag{16}$$

then we obtain the next quadratic expression to relate scalar vehicle speed and the speed of the steering axles:

$$2\dot{s}^2 = \dot{s}_f^2 + \dot{s}_a^2 = \dot{x}^2 \left(\frac{1}{\cos^2 \delta_f} + \frac{1}{\cos^2 \delta_a} \right) \tag{17}$$

Combining equations 17 with 9 through 11 renders the explicit relations between scalar speed and twist:

$$\dot{x} = \dot{s} \frac{\cos \delta_f \cos \delta_a \sqrt{2}}{\sqrt{\cos^2 \delta_f + \cos^2 \delta_a}} \tag{18}$$
$$\dot{y} = \dot{x} \frac{\tan \delta_f + \tan \delta_a}{2} \tag{19}$$
$$\dot{\phi} = \dot{x} \frac{\tan \delta_f - \tan \delta_a}{2d} \tag{20}$$

4 Series expansion

It would be convenient if a small change of vehicle twist could be related to small change of scalar speed and steer angles. Different parametrisations can be adopted for this purpose:

- use a perturbation Δr to the nominal steer angles r
- use a perturbation $\Delta r'$ to the nominal value of r'
- use a perturbation Δu to the nominal steering pole u

where the second parametrisation r' is a linear operation to r:

$$r' = Rr = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} r \tag{21}$$

representing the sum and the difference of steer angles.

4.1 Perturbation to steering angles

Therefore equations 18 through 20 are put in matrix form to obtain:

$$
{}^{v}t_v = \begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\phi} \end{pmatrix} = \dot{s}P = \dot{s}\eta P' = \dot{s}\frac{\cos\delta_f \cos\delta_r \sqrt{2}}{\sqrt{\cos^2\delta_f + \cos^2\delta_r}} \begin{bmatrix} 1 \\ \frac{\tan\delta_f + \tan\delta_r}{2} \\ \frac{\tan\delta_f - \tan\delta_r}{2d} \end{bmatrix} \tag{22}
$$

where η appears as a scale factor and P' a column matrix that relates the components of vehicle twist and both depend on the two steer angles δ_f and δ_a.

A small change of vehicle twist $\Delta^v t_v$ is modelled as a first order series expansion about the nominal values of δ_f, δ_a and $\dot{s}$:

$$
\Delta^v t_v = \begin{pmatrix} \Delta\dot{x} \\ \Delta\dot{y} \\ \Delta\dot{\phi} \end{pmatrix} = \begin{bmatrix} \dot{s}Q & P \end{bmatrix} \begin{pmatrix} \Delta\delta_f \\ \Delta\delta_a \\ \Delta\dot{s} \end{pmatrix} = \begin{bmatrix} \dot{s}(\eta Q' + P'\nu) & \eta P' \end{bmatrix} \begin{pmatrix} \Delta\delta_f \\ \Delta\delta_a \\ \Delta\dot{s} \end{pmatrix} \tag{23}
$$

where Q' is a matrix with partial derivatives of the vector P' with respect to the components of r (i.e. δ_f and δ_a):

$$
Q' = \frac{\partial P'}{\partial r} = \begin{bmatrix} 0 & 0 \\ \frac{1}{2\cos^2\delta_f} & \frac{1}{2\cos^2\delta_a} \\ \frac{1}{2d\cos^2\delta_f} & \frac{-1}{2d\cos^2\delta_a} \end{bmatrix} \tag{24}
$$

and where ν is a row matrix with partial derivatives of the scalar η with respect to r:

$$
\nu = \frac{\partial\eta}{\partial r} = \sqrt{2} \begin{bmatrix} \frac{-\sqrt{\cos^2\delta_f + \cos^2\delta_a}\sin\delta_f\cos\delta_a - \cos\delta_f\cos\delta_a \frac{\sin\delta_f\cos\delta_f}{\sqrt{\cos^2\delta_f + \cos^2\delta_a}}}{\cos^2\delta_f + \cos^2\delta_a} \\ \frac{-\sqrt{\cos^2\delta_f + \cos^2\delta_a}\cos\delta_f\sin\delta_a - \cos\delta_f\cos\delta_a \frac{\sin\delta_a\cos\delta_a}{\sqrt{\cos^2\delta_f + \cos^2\delta_a}}}{\cos^2\delta_f + \cos^2\delta_a} \end{bmatrix}^{\top} \tag{25}
$$

4.2 Change to common and differential steering angles

A small change of vehicle twist $\Delta^v t_v$ is modelled as a first order series expansion about the nominal values of δ_f, δ_a and $\dot{s}$:

$$
\Delta^v t_v = \begin{pmatrix} \Delta\dot{x} \\ \Delta\dot{y} \\ \Delta\dot{\phi} \end{pmatrix} = \begin{bmatrix} \dot{s}QR^{-1} & P \end{bmatrix} \begin{pmatrix} \Delta\delta_f + \Delta\delta_a \\ \Delta\delta_f - \Delta\delta_a \\ \Delta\dot{s} \end{pmatrix} \tag{26}
$$

using R as defined in (21) and $Q = \eta Q' + P'\nu$.

4.3 Special cases for parallel and anti-parallel steering

There are two special cases corresponding to parallel and anti-parallel steering. In the case of parallel steering are both steer angles equal. In the case of anti-parallel steering both steer angles are opposite.

Due to the structure of η composed of cosine terms this results in an interesting reduction:

$$
\eta = \frac{\cos\delta_f \cos\delta_r \sqrt{2}}{\sqrt{\cos^2\delta_f + \cos^2\delta_r}} = \frac{\cos^2\delta\sqrt{2}}{\sqrt{2\cos^2\delta}} = \cos\delta \tag{27}
$$

In case of parallel steering ($\delta_p = \delta_f = \delta_a$), the expression for vehicle twist, equation (22). simplifies to:

$$ {}^v t_v = \dot{s}\eta P' = \dot{s}\cos\delta_p \begin{bmatrix} 1 \\ \frac{2\tan\delta_p}{2} \\ \frac{0}{2d} \end{bmatrix} = \dot{s} \begin{bmatrix} \cos\delta_p \\ \sin\delta_p \\ 0 \end{bmatrix} \tag{28} $$

In case of anti-parallel steering ($\delta_s = \delta_f = -\delta_a$), the expression for vehicle twist (equation 22) simplifies to:

$$ {}^v t_v = \dot{s}\eta P' = \dot{s}\cos\delta_s \begin{bmatrix} 1 \\ \frac{0}{2} \\ \frac{2\tan\delta_s}{2d} \end{bmatrix} = \dot{s} \begin{bmatrix} \cos\delta_s \\ 0 \\ \frac{\sin\delta_s}{d} \end{bmatrix} \tag{29} $$

4.4 Perturbation to steering pole

This time the vehicle twist is expressed in the angles of the steering pole (i.e. δ_p and δ_s) and the scalar speed $\dot{s}$:

$$ {}^v t_v = \begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\phi} \end{pmatrix} = \dot{s}P = \dot{s} \begin{bmatrix} \cos\delta_s\cos\delta_p \\ \cos\delta_s\sin\delta_p \\ \frac{\sin\delta_s}{d} \end{bmatrix} \tag{30} $$

Note the similarity to (22). Also the special cases for parallel steering (28) and anti-parallel steering (29) follow for $\delta_s = 0$ and $\delta_p = 0$ respectively.

Again, a small change of vehicle twist $\Delta^v t_v$ is modelled as a first order series expansion about the nominal values of δ_s, δ_p and $\dot{s}$:

$$ \Delta^v t_v = \begin{pmatrix} \Delta\dot{x} \\ \Delta\dot{y} \\ \Delta\dot{\phi} \end{pmatrix} = \begin{bmatrix} \dot{s}U & P \end{bmatrix} \begin{pmatrix} \Delta\delta_s \\ \Delta\delta_p \\ \Delta\dot{s} \end{pmatrix} \tag{31} $$

where U is a matrix with partial derivatives of the vector P with respect to the components of u (i.e. δ_s and δ_p):

$$ U = \frac{\partial P}{\partial u} = \begin{bmatrix} -\sin\delta_s\cos\delta_p & -\cos\delta_s\sin\delta_p \\ -\sin\delta_s\sin\delta_p & \cos\delta_s\cos\delta_p \\ \frac{\cos\delta_s}{d} & 0 \end{bmatrix} \tag{32} $$

The inverse of this relation can be determined by a least squares inverse:

$$ \begin{pmatrix} \Delta\delta_s \\ \Delta\delta_p \\ \Delta\dot{s} \end{pmatrix} \approx \left(\begin{bmatrix} \dot{s}U & P \end{bmatrix}^\top \begin{bmatrix} \dot{s}U & P \end{bmatrix} \right)^{-1} \begin{bmatrix} \dot{s}U & P \end{bmatrix}^\top \Delta^v t_v \tag{33} $$

where:

$$ \begin{bmatrix} \dot{s}U & P \end{bmatrix}^\top = \begin{bmatrix} -\dot{s}\sin\delta_s\cos\delta_p & -\dot{s}\sin\delta_s\sin\delta_p & \dot{s}\frac{\cos\delta_s}{d} \\ -\dot{s}\cos\delta_s\sin\delta_p & +\dot{s}\cos\delta_s\cos\delta_p & 0 \\ \cos\delta_s\cos\delta_p & \cos\delta_s\sin\delta_p & \frac{\sin\delta_s}{d} \end{bmatrix} \tag{34} $$

and:

$$ \begin{bmatrix} \dot{s}U & P \end{bmatrix}^\top \begin{bmatrix} \dot{s}U & P \end{bmatrix} = \begin{bmatrix} \dot{s}^2\left(\frac{\cos^2\delta_s}{d^2} + \sin^2\delta_s\right) & 0 & \dot{s}\cos\delta_s\sin\delta_s\left(\frac{1}{d} - 1\right) \\ 0 & \dot{s}^2\cos^2\delta_s & 0 \\ \dot{s}\cos\delta_s\sin\delta_s\left(\frac{1}{d} - 1\right) & 0 & \frac{\sin^2\delta_s}{d^2} + \cos^2\delta_s \end{bmatrix} \tag{35} $$

This last matrix expression becomes a diagonal matrix when $d = 1$ which provides convenient expressions for the perturbations without matric calculation. However, this matrix depends on $\dot{s}$ and therefore needs to be stabilized when the nominal speed approaches to zero.

Closer examination of the least squares expression could reveal that the perturbations of $\dot{s}$ and u can be seen as a set of cross products and inner products of the feedforward and feedback components of the desired twist seen from different coordinate frames (i.e. V and S) sometimes with and sometimes without the rotation component. These products are normalised on the feedforward component of the twist or some component present in one fo the coordinate frames.

5 Path representation

A path is considered as a cascade of two functions returning a pose for each valid argument in the domain of the function:

$$f_s : t \to s \quad t \in [t_0, t_1] = [f_s^{-1}(s_0), f_s^{-1}(s_1)] \tag{36}$$

$$g_p : s \to p \quad s \in [s_0, s_1] = [f_s(t_0), f_s(t_1)] \tag{37}$$

The domain variable of g_p is of type progress which represents both change of position and change of orientation. This path defines the pose for one or two points of the vehicle. If two points are used, then it is also possible two use two different functions g_{p1} and g_{p2} for F and A respectively.

A representation of a path is made by a series of poses and an interpolation method between each pair of successive poses.

5.1 Sampled path representation

A path can be represented by a sequence of points $\{R\}$ with associated poses $\{p_r\}$.

5.2 Path interpolation

Interpolation between to successive poses $p_{r,i}$ and $p_{r,i+1}$ will be modelled using an interpolation function $g_{p,i}$ defined on an interval $[0, 1]$:

$$g_p(s) = p_{r,i} + g_{p,i}(\lambda) = p_{r,i} + g_{p,i}(\frac{s - s_i}{\Delta s_i}) \tag{38}$$

where

$$\lambda = \frac{s - s_i}{s_{i+1} - s_i} = \frac{s - s_i}{\Delta s_i} \tag{39}$$

and $\Delta s_i = s_{i+1} - s_i$.

5.2.1 Line interpolation

The linear interpolation function can be described as:

$$p_r(\lambda) = p_{r,i} + g_{p,i}(\lambda) = p_{r,i} + \lambda \begin{pmatrix} x_{r,i+1} - x_{r,i} \\ y_{r,i+1} - y_{r,i} \\ \phi_{r,i+1} - \phi_{r,i} \end{pmatrix} = p_{r,i} + \lambda \frac{\partial p_r}{\partial \lambda} \tag{40}$$

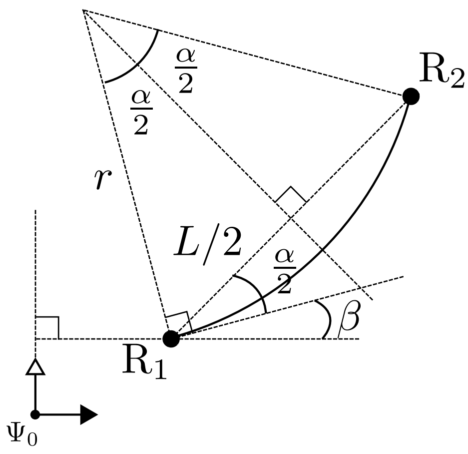

Figure 1: Arc interpolation as a rotation from R1 to R2

where $\lambda \in [0, 1]$. Where linear interpolation has been applied independently for position and orientation.

5.2.2 Arc interpolation

Alternatively, an arc interpolation function can be used:

$$p_r(\lambda) = p_{r,i} + \begin{pmatrix} r(\cos\beta \sin\gamma - \sin\beta(1-\cos\gamma)) \\ r(\sin\beta \sin\gamma + \cos\beta(1-\cos\gamma)) \\ \gamma \end{pmatrix} \tag{41}$$

where α, β, l, r and γ have been defined as (see also figure 1):

$$\alpha = \phi_{r,i+1} - \phi_{r,i} \tag{42}$$

$$\beta = \arctan\left(\frac{y_{r,i+1} - y_{r,i}}{x_{r,i+1} - x_{r,i}}\right) - \frac{\alpha}{2} \tag{43}$$

$$l = \sqrt{(x_{r,i+1} - x_{r,i})^2 + (y_{r,i+1} - y_{r,i})^2} \tag{44}$$

$$r = \frac{l}{2\sin\left(\frac{\alpha}{2}\right)} \tag{45}$$

$$\gamma = \lambda\alpha \tag{46}$$

6 Path association

It is required to find the associated points on the path segment to belong to distinct points or frames attached to the vehicle. The points are the vehicle origin V and the centres of the two axles F and A. The associated points are the path projections V', F' and A'.

Different methods can be used to find the associated points for F and A. This is based on knowing that there is a fixed distance $2d$ between these two points. Three methods can be considered:

1. independent projection of F and A (no distance constraint on F' and A')

2. projection for F and intersection for A (or vice versa) using distance constraint on F' and A'

3. projection of F and A under the distance constraint on F' and A'

These methods can be adopted irrelevant what paths will be used for F and A. For priority reasons, it is assumed that the first method will be chosen for design and implementation.

sym	meaning
p	pose of vehicle origin in reference coordinates
$\hat{p}$	estimated pose of vehicle
$\bar{p}$	projected pose of vehicle
$\bar{s}$	projected progress of vehicle
$\bar{t}$	projected time of vehicle
Δp	path tracking error of vehicle
Δs	progress error of vehicle
Δt	time error of vehicle

Table 1: Vehicle pose related symbols

The associated and the original points are used to determine both the path deviation and the nominal motion to follow the path segments. The nominal motion can be considered as the feed-forward action and the path deviation is input to determine the feedback action. Combining these actions renders the motion setpoint as output of the position control layer.

The path projections can be determined as the local minima of the norm of the path deviations of the corresponding points of the vehicle. If only the geometry of the path segments is considered then the projected progress is a point in the domain of the path geometry function:

$$\bar{s}_i = \arg\min \|g_p(s) - \hat{p}_i\| \tag{47}$$

where the L2-norm is derived from the definition of progress:

$$\|\Delta p\|^2 = \Delta x^2 + \Delta y^2 + (d\Delta\phi)^2 \tag{48}$$

If the deviation of orientation should be neglected, then the parameter d should be set equal to zero.

The path projections are determined by evaluation of the path geometry using the projected progress:

$$\bar{p}_i = g_p(\bar{s}_i) \tag{49}$$

Combining equations 47 and 49 provides the path deviation at the path projection:

$$\Delta p = \bar{p} - \hat{p} \tag{50}$$

It should be noted that this path deviation is only an estimate of the real but unknown path deviation:

$$\Delta p^{\mathrm{real}} = p^{\mathrm{ref}} - p \tag{51}$$

It is also worthwhile to represent the path deviation in a different coordinate system. Candidates are the current vehicle coordinate system or the associated projected pose on the path segment. Taking the current vehicle pose as coordinate system results in longitudinal, lateral and orientation errors:

$$ {}^{v}\Delta p = {}^{v}p_{v'} \tag{52} $$

where v indicates the current pose and v' indicates the projected pose. The calculations will need to use the estimated pose $\hat{p}$ and its associated projection $\bar{p}$ instead.

6.1 Nearest path sample

The projection of a point to this path would result in the index $\bar{i}$ of the pose with minimum projection error according to (48):

$$ \bar{i} = \arg\min \|p_{r,i} - \hat{p}\| \tag{53} $$

where $\hat{p}$ represents the point for which the projection needs to be determined. The corresponding path progress $\bar{s_i}$ for point $\bar{i}$ depends on the interpolation method and will be defined in the next subsections.

6.2 Path projection

It is required to determine the interpolated point that is considered as the projection when the distance between the points of the sampled representation of the path is too big.

This is done by minimizing the norm and solving the linearised equation:

$$ \bar{\lambda} = \arg\min \|\Delta p(\lambda)\|^2 = \arg\min \left(\Delta p(\lambda)^\top D^2 \Delta p(\lambda)\right) \tag{54} $$

which is the zero of the partial derivative with respect to λ:

$$ \frac{\partial \|\Delta p(\lambda)\|^2}{\partial \lambda} = 2\Delta p(\lambda)^\top D^2 \frac{\partial p_r}{\partial \lambda} = 0 \tag{55} $$

where:

$$ \Delta p(\lambda) = p_r(\lambda) - \hat{p} \tag{56} $$

and:

$$ D = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & d \end{bmatrix} \tag{57} $$

It is further assumed that the increase of path progress Δs over the primitive section obeys the next relation:

$$ \Delta s^2 = \left\| p_r(\lambda)\big|_{\lambda=0}^{\lambda=1} \right\|^2 = \|p_r(1) - p_r(0)\|^2 \tag{58} $$

where the L2-norm depends on the choice of parameter d (a value of zero is preferred when points A and F are following the path segments; see section 7).

6.2.1 Line projection

If the minimum projection distance is on a line segment then partial derivative of the pose p_r with respect to λ can be used to solve λ:

$$\bar{\lambda} = \frac{\frac{\partial p_r}{\partial \lambda}^\top D^2(\hat{p} - p_{r,i})}{2\frac{\partial p_r}{\partial \lambda}^\top D^2 \frac{\partial p_r}{\partial \lambda}} \tag{59}$$

where $\hat{p}$ is the pose to be projected and $p_{r,i}$ is the begin of the line segment and where the partial derivative is defined as:

$$\frac{\partial p_r}{\partial \lambda} = \begin{pmatrix} x_{r,i+1} - x_{r,i} \\ y_{r,i+1} - y_{r,i} \\ \phi_{r,i+1} - \phi_{r,i} \end{pmatrix} = p_{r,i+1} - p_{r,i} \tag{60}$$

which can be interpreted as the direction vector of a line in three dimensional space. This means that λ can be calculated as the improduct of the vector from $p_{t,i}$ to $\hat{p}$ with this direction vector divided by the squared length of this vector:

$$\bar{\lambda} = \frac{(p_{r,i+1} - p_{r,i})^\top D^2(\hat{p} - p_{r,i})}{2\left\|p_{r,i+1} - p_{r,i}\right\|^2} \tag{61}$$

This value can be used to determine the corresponding path progress based on the two successive points $p_{r,i}$, $p_{r,i+1}$ and the linear interpolation method. This gives the increment of path progress between the two path points:

$$\Delta s_i^2 = (s_{i+1} - s_i)^2 = l^2 + d^2\alpha^2 \tag{62}$$

which is equal to the L_2-norm over the interval and where α and l are:

$$\alpha = \phi_{r,i+1} - \phi_{r,i} \tag{63}$$

$$l = \sqrt{(x_{r,i+1} - x_{r,i})^2 + (y_{r,i+1} - y_{r,i})^2} \tag{64}$$

Thus the interpolated progress will be:

$$\bar{s} = \bar{s_i} + \bar{\lambda}\Delta s_i \tag{65}$$

where $\bar{s_i}$ corresponds to the progress of $p_{r,i}$.

6.2.2 Arc projection

If the minimum projection distance is on an arc segment then the next partial derivative needs to be used:

$$\frac{\partial p_r}{\partial \lambda} = \begin{pmatrix} r\alpha(\cos\beta\cos\gamma - \sin\beta\sin\gamma) \\ r\alpha(\sin\beta\cos\gamma + \cos\beta\sin\gamma) \\ \alpha \end{pmatrix} \tag{66}$$

which depends on λ by the sine and cosine of γ. This implies that we may need to iterate the projection rule to obtain the optimal λ:

$$\lambda_{n+1} = \lambda_n + \frac{\frac{\partial p_r}{\partial \lambda}^\top D^2(\hat{p} - p_r(\lambda_n))}{2\frac{\partial p_r}{\partial \lambda}^\top D^2 \frac{\partial p_r}{\partial \lambda}} \tag{67}$$

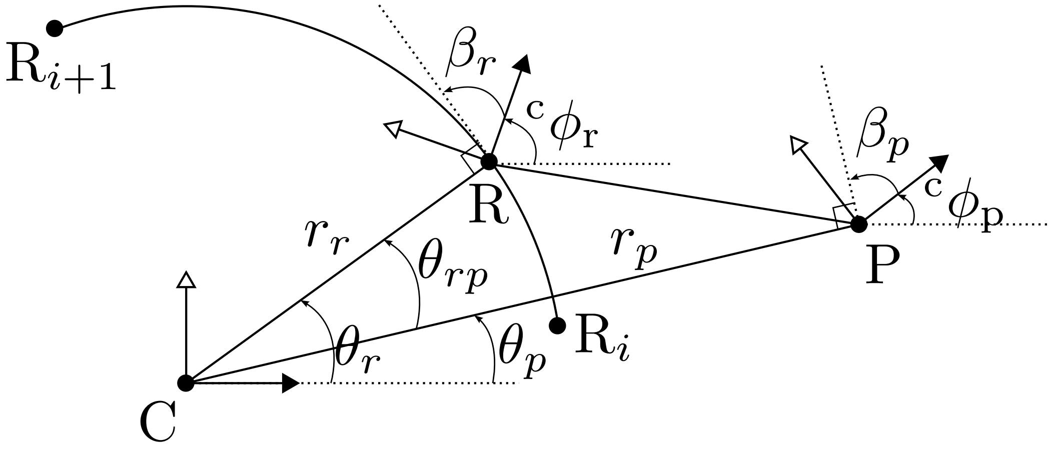

Figure 2: Arc projection in polar coordinates

where the denominator can be simplified to:

$$r^2\alpha^2\left((\cos\beta\cos\gamma-\sin\beta\sin\gamma)^2+(\sin\beta\cos\gamma+\cos\beta\sin\gamma)^2\right)+d^2\alpha^2=(r^2+d^2)\alpha^2 \tag{68}$$

The path progress corresponding to λ can be found in a way equivalent to the linear interpolation method (65). The only difference is the calculation of Δs_i which is according to:

$$\Delta s_i^2=(s_{i+1}-s_i)^2=r^2\alpha^2+d^2\alpha^2 \tag{69}$$

It can be seen that $r\alpha$ reduces to l for small values of α and therefore arc interpolation reduces to line interpolation.

6.2.3 Arc projection - alternative method

An alternative method for finding the projection of a point P with pose $p_p = (x_p, y_p, \phi_p)$ to a corresponding point R with pose $p_r = (x_r, y_r, \phi_r)$ on an arc is using a different representation. We use polar coordinates (r_p, θ_p) and (r_r, θ_r) for P and R, taking the centre of the arc as reference (see also figure 2). This representation can be used to provide an expression for the L2 norm that should be minimised to find the projection:

$$\|\Delta p\|^2=\|p_p-p_r\|^2=d^2(\theta_{rp}-\beta_{rp})^2+r_p^2+r_r^2-2r_pr_r\cos\theta_{rp} \tag{70}$$

where $\theta_{rp}=\theta_r-\theta_p$ and $\beta_{rp}=\beta_r-\beta_p$. This is based on $\phi_r=\theta_r-\beta_r+\frac{\pi}{2}$ and $\phi_p=\theta_p-\beta_p+\frac{\pi}{2}$ to relate the definition of arc interpolation, see (41), to the L2-norm. The minimum of this norm is found when the partial derivative with respect to θ_{rp} is equal to zero:

$$\frac{\partial\|\Delta p\|^2}{\partial\theta_{rp}}=2d^2(\theta_{rp}-\beta_{rp})+2r_pr_r\sin\theta_{rp}=0 \tag{71}$$

It can be observed that when $\beta_{rp}=0$, then also θ_{rp} should be zero to minimise the L2-norm. This corresponds to R and P are on the same line through the centre of the arc.

If $\beta_{rp} \neq 0$, then the zero of the non-linear function should be found. Again, this could be done by using a linear series expansion near a current estimate of the zero:

$$2d^2(\theta_{rp,i} + \Delta\theta - \beta_{rp}) + 2r_p r_r(\sin\theta_{rp,i} + \cos\theta_{rp,i}\Delta\theta) = 0 \tag{72}$$

solving $\Delta\theta$ gives the next recursive expression:

$$\theta_{rp,i+1} = \theta_{rp,i} - \frac{d^2(\theta_{rp,i} - \beta_{rp}) + r_p r_r \sin\theta_{rp,i}}{d^2 + r_p r_r \cos\theta_{rp,i}} \tag{73}$$

for which the right term converges to $-\theta_{rp,i}$ when $\beta_{rp} = 0$ as explained before. Zero is de best initial guess for θ_{rp}, where we take R on the line through C and P.

It is noteworthy to say that the denominator is very robust. If $r_p r_r \approx d^2$ (which corresponds to an arc with small radius) then P would have to be at the opposite side of the centre of the arc. If $r_p r_r$ large then P has to be very far from R to obtain a negative $\cos\theta_{rp}$.

Alternatively, it is also possible to find the zero directly by evaluating (71) using Regula Falsi.

6.3 Path intersection

Apart from a path projection other points on a path can be needed at a given distance from a known point (e.g. laying on the path). This can be done by finding the intersection of a circle and the path segments.

6.3.1 Line intersection

This case corresponds to finding the intersections between a circle and a line and selecting the intersection that is on the line segment. Reuse the first two components of (40) and find the solution that has a given distance $2d$ to a given point (x_2, y_2) gives the next vector v between the point and the intersection:

$$v = \begin{pmatrix} x_2 - (x_0 + \lambda(x_1 - x_0)) \\ y_2 - (y_0 + \lambda(y_1 - y_0)) \end{pmatrix} = \begin{pmatrix} x_{20} - \lambda x_{10} \\ y_{20} - \lambda y_{10} \end{pmatrix} \tag{74}$$

If the length of this vector is equal to 2d, then the next quadratic expression can be used to solve λ:

$$x_{20}^2 + 2\lambda x_{20}x_{10} + \lambda^2 x_{10}^2 + y_{20}^2 + 2\lambda y_{20}y_{10} + \lambda^2 y_{10}^2 = (2d)^2 \tag{75}$$

which can be simplified to:

$$l_{20}^2 + 2\lambda(v_{20} \cdot v_{10}) + \lambda^2 l_{10}^2 = (2d)^2 \tag{76}$$

where v_{20} and v_{10} and the connecting vectors between the points and where l_{20} and l_{10} are the lengths of these vectors. The roots of this equation in the interval between zero and unity are valid solutions.

6.3.2 Arc intersection

This case corresponds to finding the intersections between two circles and selecting the intersection that is on the arc segment. Reusing the first two components of (41) and finding the solution that has a given distance $2d$ to a given point (x_2, y_2) gives the next vector v between the point and the intersection:

$$v = \begin{pmatrix} x_2 - (x_0 - r\sin\beta + r(\cos\beta\sin\gamma + \sin\beta\cos\gamma)) \\ y_2 - (y_0 + r\cos\beta + r(\sin\beta\sin\gamma - \cos\beta\cos\gamma)) \end{pmatrix} = \begin{pmatrix} \Delta x - r\sin(\beta+\gamma) \\ \Delta y + r\cos(\beta+\gamma) \end{pmatrix} \tag{77}$$

where (x_0, y_0) is equal to the first two components of $p_{r,i}$. If the length of this vector is equal to $2d$, then the next quadratic expression can be used to solve γ:

$$\Delta x^2 + \Delta y^2 + r^2 - 2r\Delta x\sin(\beta+\gamma) + 2r\Delta y\cos(\beta+\gamma) = (2d)^2 \tag{78}$$

which appears to be equal to finding the angle for a cross-product of two vectors such that it is equal to a quadric term:

$$2r(\Delta x\sin(\beta+\gamma) - \Delta y\cos(\beta+\gamma)) = \Delta x^2 + \Delta y^2 + r^2 - (2d)^2 \tag{79}$$

Changing from Cartesian $(\Delta x, \Delta y)$ to polar coordinates $(l\cos\phi, l\sin\phi)$ gives the next expression:

$$2rl(\cos\phi\sin(\beta+\gamma) - \sin\phi\cos(\beta+\gamma)) = l^2 + r^2 - (2d)^2 \tag{80}$$

which can be further reduced to:

$$2rl\sin(\beta + \gamma + \frac{\pi}{2} - \phi) = l^2 + r^2 - (2d)^2 \tag{81}$$

where l is the distance between the fixed point and the centre of the circle. If this fixed point equals the centre of the circle ($l = 0$), then there is either no solution (when $2d \neq l$) or many solutions, for any value of γ on the arc (when $2d = r$).

6.4 Point association options

If the associated points need to be determined for two of more points with fixed mutual distance, then different options are available. Three different methods can be used:

1. project one point and intersect the other points with the given distance between the points
2. project all points and ignore the mutual distance between the points
3. project all points under the constraint of the mutual distances

The first two methods can be based on the previous sections. The last method requires a different algorithm.

7 Path tracking

The expressions used in this section represent the basic linearised controllers that may only be applicable to the nominal situation where projection errors and progress error are small.

All constraints that should take into accounts limits to the steering and drive system should be added in a later stage. Substantial deviation from the linear response in the nominal situation should be expected when actuators meet their saturation values.

7.1 Guidance principles

The principles for path projection and path deviation as defined in the previous section can be used to explain three different path tracking principles including their convenient en less convenient characteristics.

7.1.1 Single reference and single control point

In this case, the reference $p_{v,\mathrm{ref}}$ is the projected pose $\bar{p}_v$ which is based on the estimated pose $\hat{p}_v$. The path deviation ${}^v\Delta p_v$ is used for the feedback law and the partial derivative of the reference against progress is used for the feed-forward part of the controller.

$$\dot{p}_{v,\mathrm{out}} = \dot{p}_{v,\mathrm{ff}} + \dot{p}_{v,\mathrm{fb}} = \frac{\mathrm{d}p_{v,\mathrm{ref}}}{\mathrm{d}t} + \frac{{}^v\Delta p_v}{t_p} = \left.\frac{\partial g_p}{\partial s}\right|_{s=\bar{s}_v} \dot{s} + \frac{{}^v\Delta p_v}{t_p} \tag{82}$$

where $\bar{s}_v$ is the projected progress of V on the reference trajectory. The feed-forward term is based on the path interpolation method as:

$$\left.\frac{\partial g_p}{\partial s}\right|_{s=\bar{s}_v} = \frac{\partial p_r}{\partial \lambda}\frac{1}{\Delta s_i} \tag{83}$$

where $\frac{\partial p_r}{\partial \lambda}$ and Δs_i are taken from section 5.2 for either linear or arc interpolation.

Here the effect of path deviation on the projected progress $\bar{s}$ and its time-derivative is ignored. The corresponding steer angles r could be represented by $g_r(\bar{s})$.

A series expansion of $\dot{p}_{v,\mathrm{out}}$ as a perturbation by the feedback term to the feedforward motion expressed in steering pole u and progress rate $\dot{s}$ according to section 4.4 can be used to obtain a suitable algorithm:

$$\dot{p}_{v,\mathrm{out}} = \dot{p}_{v,\mathrm{ff}} + \dot{p}_{v,\mathrm{fb}} \approx P\dot{s} + \begin{bmatrix} \dot{s}U & P \end{bmatrix} \begin{pmatrix} \Delta\delta_s \\ \Delta\delta_p \\ \Delta\dot{s} \end{pmatrix} \tag{84}$$

where P and U are as defined in 4.4.

7.1.2 Double reference and double control point

In this case, we could state that we simply have two references $p_{f,\mathrm{ref}}$ and $p_{a,\mathrm{ref}}$ based on their associated points $\bar{p}_f$ and $\bar{p}_a$, based on their estimated poses $\hat{p}_f$

and $\hat{p}_a$ respectively. These associated points can be determined in different ways as described in section 6.4.

The path deviations ${}^f\Delta p_f$ and ${}^a\Delta p_a$ however, do not provide an analogous solution for the feedback law. Also the feed-forward part of the controller needs special attention. This is the consequence of having two reference trajectories that are mutually dependent. A unique pair of associated points on both reference trajectories should be found that have equal distance as the control points F and A of the vehicle.

The feed-forward part of the controller could be expressed by the next equations for front and aft control point:

$$\dot{p}_{f,\text{ff}} = \left.\frac{\partial g_{p1}}{\partial s}\right|_{s=\bar{s}_f} \dot{s}_f \tag{85}$$

$$\dot{p}_{a,\text{ff}} = \left.\frac{\partial g_{p2}}{\partial s}\right|_{s=\bar{s}_a} \dot{s}_a \tag{86}$$

where $\bar{s}_f$ and $\bar{s}_a$ are the projected progress of F and A on the first and second reference trajectories respectively.

By definition $\dot{s}_f$ and $\dot{s}_a$ are the speed components along the x-axis of frames F and A assuming that these points follow the trajectory with the x-axis aligned to the tangent of the path. This implies that only the linear motion component can be included in the definition of progress:

$$\dot{s}_f^2 = \dot{x_f}^2 + \dot{y_f}^2 \tag{87}$$

$$\dot{s}_a^2 = \dot{x_a}^2 + \dot{y_a}^2 \tag{88}$$

which is distinct from (6). However, consistent with (12) and (13) under the conditions of kinematic steer angles. This will affect the choice of d to compute the increment of Δs_i on the primitive path sections as defined in 6.2

Steer angle setpoints can be computed from associated points F' and A'.

$$\delta_f = \delta_{f,\text{ff}} + \delta_{f,\text{fb}} = {}^v\phi_{f'} - \frac{{}^{f'}y_f}{\tau_y \dot{s}_f} \tag{89}$$

$$\delta_a = \delta_{a,\text{ff}} + \delta_{a,\text{fb}} = {}^v\phi_{a'} - \frac{{}^{a'}y_a}{\tau_y \dot{s}_a} \tag{90}$$

where ${}^v\phi_{f'}$ and ${}^v\phi_{a'}$ are the rotation components of the poses linked to F' and A' expressed in vehicle coordinates. Here ${}^{f'}y_f$ and ${}^{a'}y_a$ express the lateral component of the path deviations Δp_f and Δp_a.

The speed setpoint should be such that both $\dot{s}_f$ and $\dot{s}_a$ obey the maximum speed constraints imposed by the path. If the speed setpoint $\dot{s}_f$ for the front axle is known, then the vehicle speed setpoint can be computed as:

$$2\dot{s}^2 = \dot{s}_f^2 \left(1 + \frac{\cos^2 \delta_f}{\cos^2 \delta_a}\right) \tag{91}$$

Taking into account that $\dot{s}$ and $\dot{s}_f$ have equal sign gives the next linear relation:

$$\dot{s} = \dot{s}_f \sqrt{\frac{\cos^2 \delta_f + \cos^2 \delta_a}{2\cos^2 \delta_a}} \tag{92}$$

Small steer angles and small differences between the absolute values of the steer angles have only a minor effect on the scale factor between $\dot{s}$ and $\dot{s}_f$.

7.1.3 Single reference and double control point

This principle can be considered as a special case of the previous, where both references are equal. The feed-forward part of the controller could be expressed by the next equations for front and aft control point:

$$\dot{p}_{f,\text{ff}} = \left.\frac{\partial g_p}{\partial s}\right|_{s=\bar{s}_f} \dot{s}_f \tag{93}$$

$$\dot{p}_{a,\text{ff}} = \left.\frac{\partial g_p}{\partial s}\right|_{s=\bar{s}_a} \dot{s}_a \tag{94}$$

where $\bar{s}_f$ and $\bar{s}_a$ are the projected progress of F and A on the common reference trajectory. Note the similarity to the feed-forward part of the two reference trajectories.

Here $\dot{s}_f$ and $\dot{s}_a$ have to satisfy (17).

7.2 Progress control

The control of vehicle pose along the path is essentially a one-dimensional problem where progress and its time-derivative as the scalar speed quantity can be used ignoring the geometric aspects of the trajectory.

The reference is defined by the previously introduced function f_s in (36). This function and its time-derivatives will be used to define the reference values:

$$s_{\text{ref}} = f_s(t) \tag{95}$$

$$\dot{s}_{\text{ref}} = f_s'(t) \tag{96}$$

$$\ddot{s}_{\text{ref}} = f_s''(t) \tag{97}$$

where f_s is monotonous so that the inverse function f_s^{-1} can be evaluated. There may be intervals with a constant value. In these cases, the largest value for t is most appropriate.

Synthesis and manipulation of the reference is further explained in the first subsection. Next, two different control modes (time-based and progress-based) are treated in the following subsections, both relying on the reference as defined above. Finally, the impact of motion control errors to the controllers is assessed.

7.2.1 Reference generation

The actual used reference for longitudinal control will be based on the synthesis and manipulation of previously generated segments assuming that path segments are defined in the design stage.

Therefore, the impact of scheduling and dispatching of path segments to reference generation will be analysed in this section. This will be done regardless of the used control mode as explained in the next subsections.

Reference generation is assumed to be adhering to a common reference of time to enable synchronous motion of multiple vehicles and designing a choreography of movements for a cluster of vehicles in certain zones (e.g. to enhance merging two flows of vehicles).

This common reference of time is a master clock providing a time reference t_m which needs to be distributed over the fleet of vehicles and the fixed infrastructure with relative time errors that have ignorable effects. The master clock will be derived from a real-time clock which provides a time reference t_r.

The actual time reference for dispatching of path segments t_b is constructed by adding an additional offset Δt to t_m. This is done to handle time gaps that are caused by scheduling, dispatch and execution delays and intentional waiting intervals.

Evaluation of (36) will provide the reference progress value s_b for a specific vehicle. A chain of function mappings provides the reference for each vehicle:

$$t_r \to t_m \to t_b \to s_m \tag{98}$$

where t_b is the time-reference equivalent to t in the next subsections.

The first mapping ($t_r \to t_m$) can be used to implement a global slowdown and suspend of the synchronous process maintaining the choreography. This can be communicated ahead of time by providing the parameters for sections that will be evaluated soon to obtain the master time reference.

The second mapping ($t_m \to t_b$) can be used to implement waiting time and to accept incurred delays caused by disturbances (e.g. waiting for s leading vehicle).

The last mapping ($t_b \to s_m$) should be used to execute predefined speed profiles and may require some manipulation to implement speed transitions caused by speed limits.

7.2.2 Time-based control mode

In the first mode a path will be executed starting at t_0, i.e. the start time and time is originating from a real-time or virtual clock. The start time needs to be chosen to match the current time or the end time of the previous path segment.

The controller output forwards the feed-forward values $\dot{s}_{\text{ref}}$ and $\ddot{s}_{\text{ref}}$ adding a simple feedback term based on the progress error:

$$\dot{s}_{\text{set}} = f_s'(t) + \frac{\Delta s}{\tau_s} = \dot{s}_{\text{ref}} + \frac{\Delta s}{\tau_s} \tag{99}$$

$$\ddot{s}_{\text{set}} = f_s''(t) = \ddot{s}_{\text{ref}} \tag{100}$$

$$\Delta s = s_{\text{ref}} - \bar{s} \tag{101}$$

where τ_s is a constant to implement a proportional controller. It could be extended to implement a PI controller.

When the vehicle has to reduce speed caused by external factors limiting $\dot{s}_{\text{set}}$ then Δs will increase and the controller output will increase to catch-up the delay that caused the progress error.

This can be prevented by calculating the time offset Δt when the limiting factor ends (at t_e)and applying this offset for the reference:

$$s_{\text{ref}} = f_s(t - \Delta t|_{t=t_e}) \tag{102}$$

$$\Delta t = f_s^{-1}(\bar{s}) - t \tag{103}$$

7.2.3 Progress-based control mode

In the second mode a path will be executed starting at s_0, i.e. the start progress that corresponds to the end of the last path segment. The derivatives have to be determined indirectly by using the inverse of f_s:

$$\bar{t} = f_s^{-1}(\bar{s}) \tag{104}$$

$$\bar{\dot{s}} = f'_s(\bar{t}) = f'_s(f_s^{-1}(\bar{s})) \tag{105}$$

$$\bar{\ddot{s}} = f''_s(\bar{t}) = f''_s(f_s^{-1}(\bar{s})) \tag{106}$$

this results in the next progress-based control algorithm:

$$\dot{s}_{\text{set}} = f'_s(\bar{t}) + \frac{\Delta s}{\tau_s} = \bar{\dot{s}} + \frac{\Delta s}{\tau_s} \tag{107}$$

$$\ddot{s}_{\text{set}} = f''_s(\bar{t}) = \bar{\ddot{s}} \tag{108}$$

$$\Delta s = s_{\text{ref}} - \bar{s} \tag{109}$$

where τ_s is a constant to implement a proportional controller. Notice the similarity with the time-based equations.

Integrator wind-up can be prevented by adjusting the time offset Δt as explained in the previous subsection. It can also be done by disabling the feedback term of (107).

7.2.4 Impact of motion control errors

Motion control will only be able to follow the progress control setpoints under ideal circumstances. Normally, there will be control errors:

$$\Delta\dot{s} = \dot{s}_{\text{set}} - \hat{\dot{s}} \tag{110}$$

$$\Delta\ddot{s} = \ddot{s}_{\text{set}} - \hat{\ddot{s}} \tag{111}$$

where the second equation cannot be used directly as $\ddot{s}$ is hard to be measured or estimated.

The impact of $\Delta\dot{s}$ will be analysed in this subsection for the next motion model:

$$\hat{\dot{s}} = \eta\dot{s}_{\text{set}} - \epsilon\text{sign}(\dot{s}) = \eta\dot{s}_{\text{set}} - \dot{s}_{\text{set,min}}\text{sign}(\dot{s}) \tag{112}$$

with $\dot{s}_{\text{set,min}} = \frac{\epsilon}{\eta}$ and gives the next motion error:

$$\Delta\dot{s} = (1-\eta)\dot{s}_{\text{set}} + \epsilon \tag{113}$$

under the restriction that $\dot{s} > 0$ and progress error:

$$\Delta s = \int_0^t \Delta\dot{s}dt = \int_0^t ((1-\eta)\dot{s}_{\text{set}} + \epsilon)dt \tag{114}$$

Although this is a very basic model, it nevertheless reveals the main issues that occur with longitudinal control (i.e. the issues at the start and end of a trajectory). These issues have also been investigated by a modelling and simulation study where the motion model includes friction and a proportional error.

There are four different situations that need to be further analysed:

- acceleration: $\dot{s}\ddot{s} > 0$ in time-based mode
- acceleration: $\dot{s}\ddot{s} > 0$ in progress-based mode
- deceleration: $\dot{s}\ddot{s} < 0$ in time-based mode
- deceleration: $\dot{s}\ddot{s} < 0$ in progress-based mode

7.2.5 Motion control errors - acceleration case

A specific reference for acceleration ($t \in [0, 1]$ and $s \in [0, \frac{a}{2}]$) is:

$$s_{\text{set}} = \frac{at^2}{2} \tag{115}$$

$$\dot{s}_{\text{set}} = at \tag{116}$$

$$\ddot{s}_{\text{set}} = a \tag{117}$$

In time-based mode, the movement starts at:

$$t_{\min} = \frac{\dot{s}_{\text{set},\min}}{a} = \frac{\epsilon}{a\eta} \tag{118}$$

$$s_{\text{set}}\big|_{t_{\min}} = \frac{1}{2a}(\frac{\epsilon}{\eta})^2 \tag{119}$$

and results in the next errors ($t > t_{\min}$):

$$\Delta s = \left[\frac{a(1-\eta)t^2}{2} + \epsilon t\right]_{t_{\min}}^{t} + \frac{1}{2a}(\frac{\epsilon}{\eta})^2 \tag{120}$$

$$\Delta \dot{s} = a(1-\eta)t + \epsilon \tag{121}$$

$$\Delta \ddot{s} = a(1-\eta) \tag{122}$$

which results in the next position error at $t = 1$:

$$\Delta s\big|_{t=1} = \frac{a(1-\eta)}{2} - \frac{\eta}{2a}(\frac{\epsilon}{\eta})^2 + \epsilon \tag{123}$$

In progress-based mode, the reference is based on the actual progress value:

$$s_{\text{set}} = \frac{at^2}{2} \tag{124}$$

$$\dot{s}_{\text{set}} = \sqrt{2a\bar{s}} \tag{125}$$

$$\ddot{s}_{\text{set}} = a \tag{126}$$

where the vehicle needs to start at a minimum position $\bar{s}_{\min}$:

$$\bar{s}_{\min} = \frac{\dot{s}^2_{\text{set},\min}}{2a} = \frac{1}{2a}(\frac{\epsilon}{\eta})^2 \tag{127}$$

and

$$\Delta s = \int \Delta \dot{s} ds \tag{128}$$

$$\Delta \dot{s} = (1-\eta)\sqrt{2a\bar{s}} + \epsilon \tag{129}$$

$$\Delta \ddot{s} = (1-\eta)a \tag{130}$$

7.2.6 Motion control errors - deceleration case

A specific reference for deceleration ($t \in [-1, 0]$ and $s \in [\frac{-a}{2}, 0]$) is:

$$s_{\text{set}} = \frac{-at^2}{2} \tag{131}$$

$$\dot{s}_{\text{set}} = -at \tag{132}$$

$$\ddot{s}_{\text{set}} = -a \tag{133}$$

In time-based mode, the movement stops at:

$$t_{\max} = \frac{\dot{s}_{\text{set,min}}}{-a} = -\frac{\epsilon}{a\eta} \tag{134}$$

$$s_{\text{set}}\big|_{t_{\max}} = -\frac{1}{2a}(\frac{\epsilon}{\eta})^2 \tag{135}$$

and results in the next errors:

$$\Delta s = -a(1-\eta)\frac{(t^2-1)}{2} + \epsilon(t+1) \tag{136}$$

$$\Delta \dot{s} = -a(1-\eta)t + \epsilon \tag{137}$$

$$\Delta \ddot{s} = -a(1-\eta) \tag{138}$$

starting with an initial error that is zero $\Delta s\big|_{t=-1} = 0$ which results in the next position error at $t_{\max}$:

$$\Delta s\big|_{t=t_{\max}} = \frac{a(1-\eta)}{2} - \frac{1}{2a}(\frac{\epsilon}{\eta})^2 - \frac{\eta}{2a}(\frac{\epsilon}{\eta})^2 + \epsilon \tag{139}$$

At $t = 0$, the total position error is:

$$\Delta s\big|_{t=0} = \Delta s\big|_{t=t_{\max}} - s_{\text{set}}\big|_{t_{\max}} = \frac{a(1-\eta)}{2} - \frac{\eta}{2a}(\frac{\epsilon}{\eta})^2 + \epsilon \tag{140}$$

In progress-based mode, the reference is based on the actual progress value:

$$s_{\text{set}} = \frac{-at^2}{2} \tag{141}$$

$$\dot{s}_{\text{set}} = \sqrt{-2a\bar{s}} \tag{142}$$

$$\ddot{s}_{\text{set}} = -a \tag{143}$$

where the vehicle will stop at a minimum position $\bar{s}_{\max}$:

$$\bar{s}_{\max} = -\frac{\dot{s}^2_{\text{set,min}}}{2a} = -\frac{1}{2a}(\frac{\epsilon}{\eta})^2 \tag{144}$$

and

$$\Delta s = \int \Delta \dot{s} ds \tag{145}$$

$$\Delta \dot{s} = (1-\eta)\sqrt{-2a\bar{s}} + \epsilon \tag{146}$$

$$\Delta \ddot{s} = (1-\eta)a \tag{147}$$

7.2.7 Motion control errors - summary and comparison

If we consider the increase of the progress error Δs over the interval when the vehicle moves in time-based mode, then we can notice that the increase of the error is equal:

$$\Delta s\big|^{\text{accel}}_{t=1} - \Delta s\big|^{\text{accel}}_{t_{min}} = \Delta s\big|^{\text{decel}}_{t_{max}} - \Delta s\big|^{\text{decel}}_{t=-1} = \frac{a(1-\eta)}{2} - \frac{\epsilon^2}{2a\eta^2} - \frac{\epsilon^2}{2a\eta} + \epsilon \tag{148}$$

acceleration		deceleration	
t	Δs	t	Δs
0	0	-1	0
$\frac{\epsilon}{a\eta}$	$\frac{1}{2a}(\frac{\epsilon}{\eta})^2$	$-\frac{\epsilon}{a\eta}$	$\frac{a(1-\eta)}{2} - \frac{1}{2a}(\frac{\epsilon}{\eta})^2 - \frac{\eta}{2a}(\frac{\epsilon}{\eta})^2 + \epsilon$
1	$\frac{a(1-\eta)}{2} - \frac{\eta}{2a}(\frac{\epsilon}{\eta})^2 + \epsilon$	0	$\frac{a(1-\eta)}{2} - \frac{\eta}{2a}(\frac{\epsilon}{\eta})^2 + \epsilon$

Table 2: Time-based motion errors during acceleration and deceleration

Table 2 shows the progress errors that occur during time-based acceleration and deceleration according to the progress reference of the previous sections. The total progress error is equal over the time window as could be expected from symmetry considerations.

Table 3 shows that the progress offset from the start during acceleration is equal to the progress offset to the end during deceleration as could be expected from symmetry considerations.

There is no data about the duration of the acceleration and deceleration intervals as the time could be arbitrarily long on the sections where the setpoint velocity is below the minimum value.

	acceleration	deceleration
s	$s_{min} = \frac{1}{2a}(\frac{\epsilon}{\eta})^2$	$s_{max} = -s_{min}$

Table 3: Progress-based motion errors during acceleration and deceleration

7.3 Steering control

Control of vehicle pose by limiting the path deviation is assumed to be based on limiting the path deviations of F and A. The simplified control-law is given in (89) and (90).

Additional analysis is required to take care of the special cases when the vehicle speed is low and in particular near the zero crossing. A corrective action is speed dependent and could result in too aggressive steer actions.