Luca Baglivo

Autonomous Vehicles Navigation

Trajectory Planning and Control

Notes for the "Space Robotics" course
in Aerospace Engineering

PREFACE

The purpose of this lecture is to provide an introductory overview of the basic theories and applications on trajectory planning and control of mobile robots in particular of autonomous vehicles. Treaty, there is an inevitable involvement of the study of non-holonomic systems, and indeed precisely because of this feature of mobile robots it has been created and is still developing a line of research that aims to provide general methods for navigation of non holonomic robots like those provided by the more mature study of the problems inherent holonomic robots. It is natural, therefore, refer to the manipulators robot as holonomic systems to understand the differences in application and complexity of the theories in one and other field.

I wish to thank prof. Mariolino De Cecco and prof. Francesco Angrilli for important ideas, valuable advices and for the stimulus to the preparation of this booklet.

TABLE OF CONTENTS

After a needed introduction on non-holonomic systems, some of the main kinematic models of autonomous wheeled vehicles are described, the construction of which is closely related to the imposition of no wheel spin constraint. The kinematic model of the robot affects, in an urgent way, planning and motion control tasks. Below are some methods of motion planning that exploit the nilpotent property of many non-holonomic systems. A planning method for continuous curvature paths which take into account non-holonomic constraints and mechanical and dynamic constraints of the robot is described. For the car model type a method for the planning of open loop checks is described. To the scheduled tests it is necessary to add the feedback inspections thus completing the task correctly even in the presence of the inevitable measurement disruptions, deviations from the initial conditions provided for and deviations from the ideal model. There is an introduction to the feedback inspections both with theoretical approach (linearised static feedback), and with heuristic approach (tracking heuristic algorithms).

TABLE OF CONTENTS

1. INTRODUCTION

Mobile robotics deals with the study of robotic systems which are able to move independently in a certain type of environment in order to accomplish a given task and nowadays it is used in various fields, from civil to industrial sectors, from military to aerospace. Within mobile robotics the study of autonomous navigation techniques is essential as it deals with the main problem, to enable the robot to locate itself, plan its movements and control its execution.

The problem of *location* sees the need to use the position measuring instruments that are suitable to the environment in which the robot operates and to integrate more sensors in a localization system. To obtain a single position measurement that integrates in a great way the individual measures of the various tools, appropriate techniques are used which are being studied as the more general problem of *sensor fusion* [1].

Another fundamental problem of autonomous navigation is the *planning* of motion, on two levels: a high level of global strategy which is about achieving the final goal, and, functional to it, a lower level of local planning.

Because of unavoidable disturbance factors, it is necessary, as for manipulating robots, with greater reason also for mobile robots, a *feedback inspection* on the planned motion.

A part from a few exceptions, from a kinematic point of view, wheeled mobile robots are *non-holonomic* systems. The non-holonomic systems are characterized by equations of constraint on the speed of the variables that describe the system. These equations *are* not *integrable* and typically they appear when the system has a lower number of controls to the number of degrees of freedom. For example, a car type robot has two controls, the linear and angular velocity, while the variables that describe the evolution of the system are three, for example, the Cartesian position of a point and its alignment. The consequence of non-holonomic constraints is that all trajectories in the space of the generalized variables of the robot are acceptable. For this the techniques of classical geometry developed for the motion of holonomic systems such as manipulators, are not applicable to the non-holonomic ones.

1.1. Structured, semi-structured, unstructured environments

The overall strategy of planning and motion control strongly depends on the type of environment in which the robot moves. Three macro-categories of environments based on the information of the environment available can be classified:

- Structured *environments* : You have a complete knowledge of the topology of the free spaces layout and the location of obstacles in addition to the position of the target. A typical example is that of an industrial logistics environment (covered goods warehouse, palletizing machines, etc.). The path can be fully planned off-line in advance and unforeseen situations cannot be handled by the robot to complete its task
- *Semi-structured* environments: knowledge of the environment is not complete because, for example, you do not have sufficient information beforehand on the target or because there is a flexible management of the path locally and in real time in order to avoid obstacles, which were not known beforehand. The robot can plan its movement in a space surrounding it, by changing the path which was already planned off-line comprehensively.
- Unstructured *environments* : there is not enough information to comprehensively plan the movements in advance. The robot must be able to locate itself in real time with respect to the environment and plan the motion in progress in order to complete the task (problem SLAM, Simultaneous Localization and Map Building).

2. AGV KINEMATIC MAIN MODELS

This chapter summarizes the main kinematic models of autonomous vehicles used in mobile robotics. These models are used to describe the time evolution of the system variables of interest, ie what is termed the *state* of the system in the space of its possible configurations defined by the vector of *generalized coordinates* or *configuration variables*. These coordinates are the same number as the degrees of freedom of the system and may or may not have physical meaning. The construction of the kinematic models starts from a basic assumption that unites them, the existence of the constraint of no wheel spin. This constraint belongs to the class of *non-holonomic* or *anholonomic* constraints, what mathematically, in the case of wheeled mobile robots, results in a restriction of the possible values of magnitude and direction of the speed and, consequently, of the possible trajectories achievable starting from a certain configuration. The models that meet this constraint are ideal; in reality there are always some deviations from the model due to the effective non-ideality of the constraints.

In this discussion the attention is focused on the system control model considered ideal. Moreover, even assuming that they are respected, constraints introduce complications in planning and trajectory control of non-holonomic vehicles.

2.1. Non-holonomic systems

Consider the mechanical system the configuration of which can be described completely by the vector of *generalized coordinates* $q \in Q$, belonging to the real vector space Q of the size n. Typically the motion of the system is subject to constraints caused by the very structure of the system or by the way in which it is implemented or controlled. These constraints may be bilateral or unilateral (respectively expressed by equations or inequalities) and may depend or not on the weather. Limited to bilateral constraints not dependent on time, the mathematical relationships that express the constraint can be equations in the generalized coordinates and / or in their derived (in this case the constraint is called differential).

The constraints called *holonomic* can have the shape:

$$h_i(q) = 0, \quad i = 1,..,k < n \qquad (2\text{-}1)$$

Functions $h_i : Q \to \mathbb{R}$ are assumed continuous and independent and the system subject to such constraints is also called holonomic. Most manipulators provide a typical example of holonomic constraints

A system subject to such constraints is forced to restrict the totality of its possible configurations in a subspace of Q size $n-k$ .. The system configuration can be described no longer by n but by $n-k$ new generalised coordinates that represent the degrees of freedom of the system. For example, in the case of a fixed-base kinematic chain composed of two elements bound with two revolute pairs (or prismatic), one on the frame and the other one between the two elements, the constraints-free system has $n = 2 \cdot m = 12$ degrees of freedom ($m = 6$ for every free body in space), while the holonomic constraints of revolute pairs (each requires 5 single bonds) take away $k = 2 \cdot 5 = 10$ df for which the configuration of the system can be completely described by $n - k = 2$ dof

The restraints called *kinematic* include the generalized coordinates and speed. In the linear form with respect to the generalized speed (called of *Pfaffian*)*:*

$$a_i^T(q)\dot{q} = 0, \quad i = 1,...,k < n, \qquad \text{or}$$

$$A^T(q)\dot{q} = 0 \qquad (2\text{-}2)$$

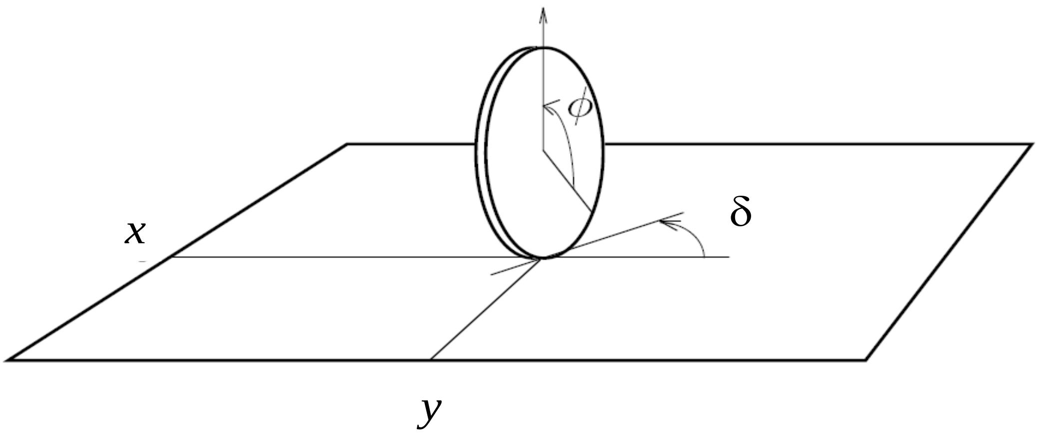
The matrix A^T is composed by k independent vectors.
These constraints limit the possible movements of the system by restricting the set of generalized speed that can be carried out in a given configuration, the typical example is that of a wheel with the no sliding constraint reported in the following paragraph.

Naturally the existence of holonomic constraints implies the existence of kinematic constraints obtained by derivation with respect to time:

$$\frac{\partial h_i}{\partial q}\dot{q} = 0, \quad i = 1,\dots,k$$

Instead the opposite is not necessarily true. In fact it happens that the kinematic constraints cannot be integrated, meaning they cannot be put in the form (2-1). For example, in the case of only one kinematic constraint, if the constraint is integrable then there is a function $h(q)$ so that $\partial h / \partial q = a^T(q)$ it is $h(q_0) = c$, where c there is an integration constraint linked to the initial conditions q_0. If the constraints cannot be integrated, the constraints and the mechanical system are called **non-holonomic** or **anholonomic**. A typical example of a non-holonomic system is the car. The speeds of a car are constrained because the wheels cannot move in a lateral direction. Therefore, the same car cannot move sideways nor rotate on the spot. Despite this, we know that it is possible to park a vehicle in any point and in any direction, compatible with the presence of obstacles.

2.2. Unicycle model



Picture 1: Unicycle model with four generalized coordinates.

Consider a disk that can roll without sliding on a flat surface (Picture 1), maintaining its mid flat surface in a vertical position. The configuration is fully described by four variables: two Cartesian coordinates (x , y) of the point of contact with the ground with respect to a fixed system; the corner δ defines the orientation of the disc with respect to the axis x; the corner ϕ between a radial axis fixed on the disc and the vertical axis. Because of the non-slip constraint, the generalized speeds of the system cannot have arbitrary values. In fact, by indicating with R the radius of the disc they must respect the constraints

$$\dot{x} - R\dot{\phi}\cos(\delta) = 0$$

$\dot{y} - R\dot{\phi}\sin(\delta) = 0$,

or in matrix form:

$$A^T(q)\dot{q} = \begin{bmatrix} 1 & 0 & -R\cos\delta \\ 0 & 1 & -R\sin\delta \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\phi} \end{bmatrix} = 0$$

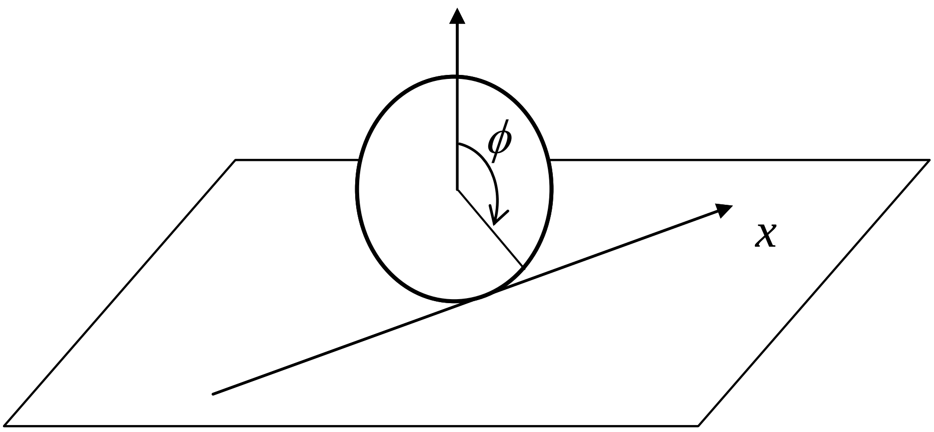
which clearly express the condition that the speed of the centre of the disc is maintained in the mid flat surface of the disk, i.e. is always tangent to the trajectory. From these differential constraints it is not possible to obtain a report on the generalized coordinates, the number of degrees of freedom of the system's speeds is reduced to two (4 minus 2) but the generalized coordinates cannot be reduced. In other words the constraints remain only on the speeds and cannot be integrated and, as a consequence, there is no limitation on the configurations which can be accessed from the disk. In fact, you can move the disk from one configuration $[x_1, y_1, \delta_1, \phi_1]$ to a configuration $[x_2, y_2, \delta_2, \phi_2]$ through the following motion sequence:

1. The disk must be rolled taking the point of contact from (x_1, y_1) to (x_2, y_2) along any length curve $R(\phi_2 - \phi_1 + 2k\pi)$, with k positive whole number
2. The disk is rotated around its own vertical axis from δ_1 to δ_2.

Clearly, the potential curves are endless. From the constraint equations the unicycle kinematic model can be seen immediately which forms the basis for any type of model obtained from non-slip constraints:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\delta} \end{bmatrix} = \begin{bmatrix} \cos\delta \\ \sin\delta \\ 0 \end{bmatrix} v + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} w \quad (2\text{-}3)$$

Two controls are introduced in the model v and w, respectively, the speed of the centre of the disc and the angular speed of the disc around the vertical axis, which are equal in number to the degrees of freedom on the speeds.



Picture 2: Disc constrained on straight rail.

Different, although almost identical, is the case of a disc which rolls without sliding on a vertical straight guide which, clearly, generates an integrable constraint, thus holonomic, and this reduces the number of generalized coordinates from two to one as in this case the straight coordinate x and the angle of rotation ϕ are linked by a completed relationship of proportionality through the radius of the disc. Mathematically:

$$\dot{x} = \dot{\phi} R$$

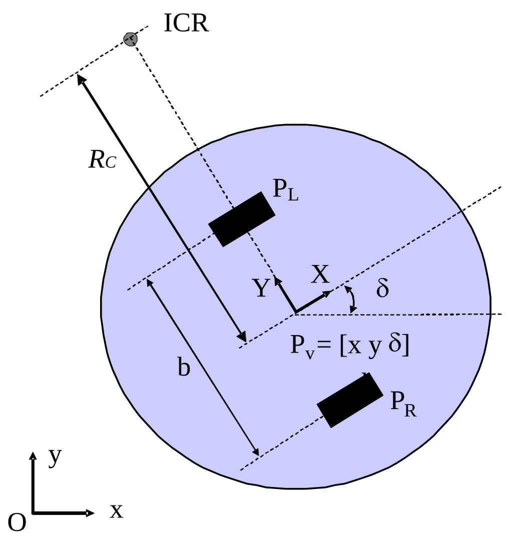
Therefore, by incorporating:

$$x = \phi R + c$$

where c it is a constant that depends on the initial conditions. In this example, the single constraint on speeds is written in the Pfaffian form with $a^T(q) = [1 \quad -R]$ and you can get the solution of $\partial h / \partial q = a^T(q)$, which is $h(q) = x - \phi R = c$.

It also appears evident how, in the presence of speed constraints, it is important to establish whether they can be integrated or not in order to analyze what is the configuration space accessible by the system, ie if this space is restricted by the constraints or not and in this case the constraints are non-holonomic.

2.3. Differential guide model



Picture 3: The differential drive model diagram.

A classic transport system for mobile robots is constituted by two parallel driving wheels, controlled in speed or acceleration by two independent motors. Since two wheels are not sufficient for a stable support, some idler wheels are added to keep the vehicle with a constant inclination with respect to the ground.

Taking as a reference the midpoint between the driving wheels, P_v, the system's configuration is defined by three coordinates: the vector $[x, y]$ which identifies the position of P_v and the trim angle δ which defines the instantaneous travelling direction of the vehicle and coincides with the orientation of the driving wheels. Given b the distance between the wheels, v_R, v_L, v respectively the speed of the right wheel and the left one and that of the reference point, the kinematic model can be written as

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\delta} \end{bmatrix} = \begin{bmatrix} \frac{v_R + v_L}{2}\cos\delta \\ \frac{v_R + v_L}{2}\sin\delta \\ \frac{v_R - v_L}{b} \end{bmatrix} = \begin{bmatrix} v\cos\delta \\ v\sin\delta \\ \frac{v_R - v_L}{b} \end{bmatrix} \text{(2-4)}$$

The model is obtained by simply considering the rotation of the vehicle around its instantaneous centre of rotation (ICR) and expressing the speed v of the reference point as a function of the speeds of the right and left wheels. Below are calculations for further clarity. In the mobile reference system integral with the robot, the non null components (along the axis *x*) of the speed vectors are:

$$v = \left(\frac{dP_v}{dt}\right)_x = \dot{\delta}R_c$$

$$v_R = \left(\frac{dP_R}{dt}\right)_x = \dot{\delta}(R_c + b/2)$$

$$v_L = \left(\frac{dP_L}{dt}\right)_x = \dot{\delta}(R_c - b/2)$$

By adding and subtracting, member to member, the last two equations you get:

$$v = \frac{v_R + v_L}{2}$$

$$v_R - v_L = \dot{\delta}b$$

The kinematic model changes depending on the generalized coordinates and on the controls that are chosen, for the latter there are two examples of models of the same system. The first is a dynamic model controlled directly by using accelerations u_R and u_L of the wheels:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\delta} \\ \dot{v}_R \\ \dot{v}_L \end{bmatrix} = \begin{bmatrix} \frac{v_R + v_L}{2}\cos\delta \\ \frac{v_R + v_L}{2}\sin\delta \\ \frac{v_R - v_L}{b} \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} u_R \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} u_L \text{(2-5)}$$

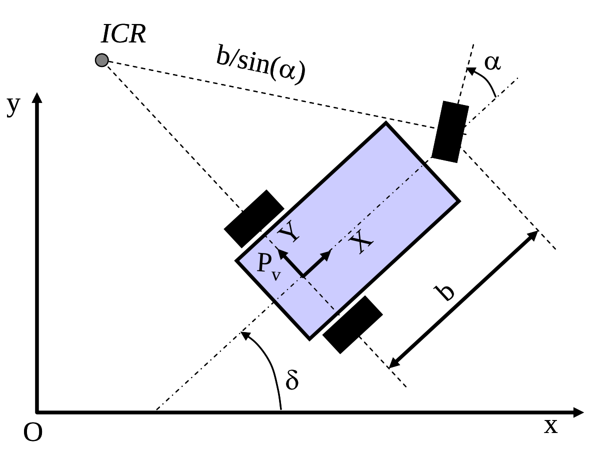
The second has as controls the speed of the middle point between the driving wheels and the angular velocity of the vehicle, therefore:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\delta} \end{bmatrix} = \begin{bmatrix} \cos\delta \\ \sin\delta \\ 0 \end{bmatrix} v + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} w \text{(2-6)}$$

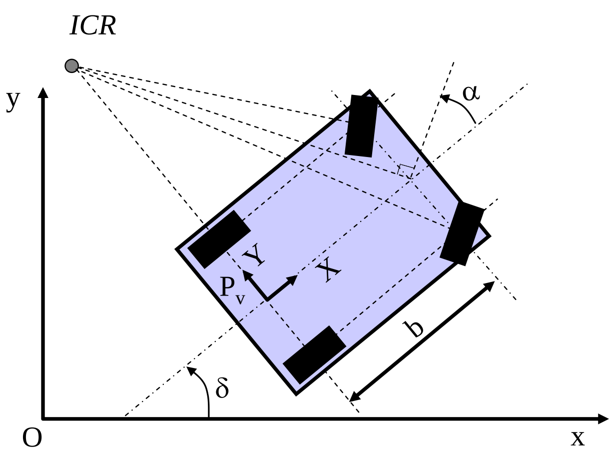
Model (2-6) is formally identical to that obtained for the unicycle.

As you will see in the dedicated section, the characteristics of controllability and the same constraints of the system change according to the model that is taken into consideration.

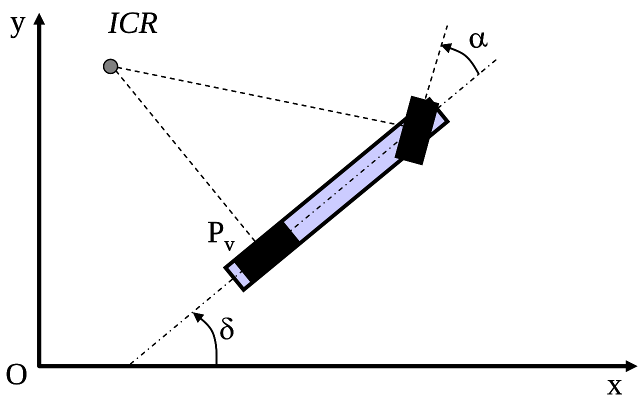
2.4. Three wheel and car-like model



Picture 4: Three-wheeled model



Picture 5: Car model (car-like)



Picture 6: Bicycle model

One wants to build a model of a three-wheeled vehicle with fixed axle rear wheels and steerable front wheel (Picture 4). Using the generalized coordinates $[x, y, \delta, \alpha]$, where (x, y) are the Cartesian coordinates of the mean rear axle point (P_v), δ is the angle that measures the balance of the vehicle with respect to the axis x, α is the steering angle of the front wheel measured in relation to the axis of longitudinal symmetry of the vehicle.

First, we identify the constraints of the system, it is clear that they are two non-slip constraints: one for the front wheel, the other constraint is unique for all the rear wheels (which are parallel), and can be imagined imposed on a fictitious wheel located at point P_v, in fact any wheel which is found with its rolling axis aligned with the rear axle has to follow the same rigid motion in an integral manner to that of the axis and according to an orthogonal speed to it. Decided (x_a, y_a) the coordinates of the front wheel can be written:

1° non-holonomic constraint for the front wheel

$$\dot{x}_a \sin(\delta + \alpha) - \dot{y}_a \cos(\delta + \alpha) = 0$$

2° non-holonomic constraint for the rear wheel

$$\dot{x}\sin\delta - \dot{y}\cos\delta = 0$$

As

$$x_a = x + b\cos\delta$$
$$y_a = y + b\sin\delta$$

The first constraint becomes:

$$(\dot{x} - \dot{\delta}b\sin\delta)\sin(\delta+\alpha) - (\dot{y} + \dot{\delta}b\cos\delta)\cos(\delta+\alpha) =$$
$$= \dot{x}\sin(\delta+\alpha) - \dot{y}\cos(\delta+\alpha) - \dot{\delta}b\cos\alpha = 0$$

Therefore the constraint matrix is

$$A^T(q) = \begin{bmatrix} \sin(\delta+\alpha) & -\cos(\delta+\alpha) & -bcos\alpha & 0 \\ \sin\delta & -\cos\delta & 0 & 0 \end{bmatrix}$$

The rank of the matrix is constant[1] for any value of δ and α and it is worth 2. This implies that the space of solutions of the system consisting of the two constraints, i.e. the generalized eligible speeds, has dimension 2: we can control the system with two degrees of freedom on speed.

There are several ways to obtain the kinematic model of a system subject to kinematic constraints, one of them is to start from the equation of constraints $A^T\dot{q} = 0$, evaluate the degree of the matrix A^T, and calculate the core associated with the matrix. However the result of this operation does not lead to the writing of a model with immediate physical significance, therefore, it is preferable to obtain the model from kinematic methods of mechanics.

Consider the rotation of the vehicle around its centre of instantaneous rotation (ICR); called P_a the position vector of the front wheel with respect to the fixed reference system, R_{ca} is $R_{c\text{v}}$ respectively the radius of curvature in correspondence of the front wheel and the pain of reference P_{v}:

$$v_a = \dot{\delta}R_{ca} = \dot{\delta}\frac{b}{\sin\alpha}$$
$$v = \dot{\delta}R_{c\text{v}} = \dot{\delta}\frac{b}{\tan\alpha}$$

The magnitudes obtained above respectively express the speed of the front wheel and the point of reference and their signs depend on the conventions established for signs of $\dot{\delta}$ and of *b*.

From the first equation:

$$\dot{\delta} = \frac{v_a\sin\alpha}{b} \quad (\text{2-7})$$

The expression of the curvature also results from the second equation κ

[1]The determining of the minor format from the first two rows and the first two columns worth $\sin\alpha$, for $\alpha = 0$ the two determinants of minor formats from the third column and respectively from the first and second column will never vanish simultaneously for each value of δ.

$$\dot{\delta} = \frac{1}{R_{cv}} v = \kappa v \qquad \Rightarrow \qquad \kappa = \frac{\tan\alpha}{b} \quad (2\text{-}8)$$

Therefore, to express speeds $\dot{x}$ and $\dot{y}$ as a function of the speed of the point of reference, of the balance and of the steering angle:

$$\begin{aligned} \dot{x} &= v\cos\delta = \dot{\delta}\frac{b}{\tan\alpha}\cos\delta = v_a\cos\alpha\cos\delta \\ \dot{y} &= v\sin\delta = \dot{\delta}\frac{b}{\tan\alpha}\sin\delta = v_a\cos\alpha\sin\delta \end{aligned} \quad (2\text{-}9)$$

In this way the kinematic model of the system, in the form of the control system, is the following:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\delta} \\ \dot{\alpha} \end{bmatrix} = \begin{bmatrix} \cos\alpha\cos\delta \\ \cos\alpha\sin\delta \\ \dfrac{\sin\alpha}{b} \\ 0 \end{bmatrix} u_1 + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} u_2 \quad (2\text{-}10)$$

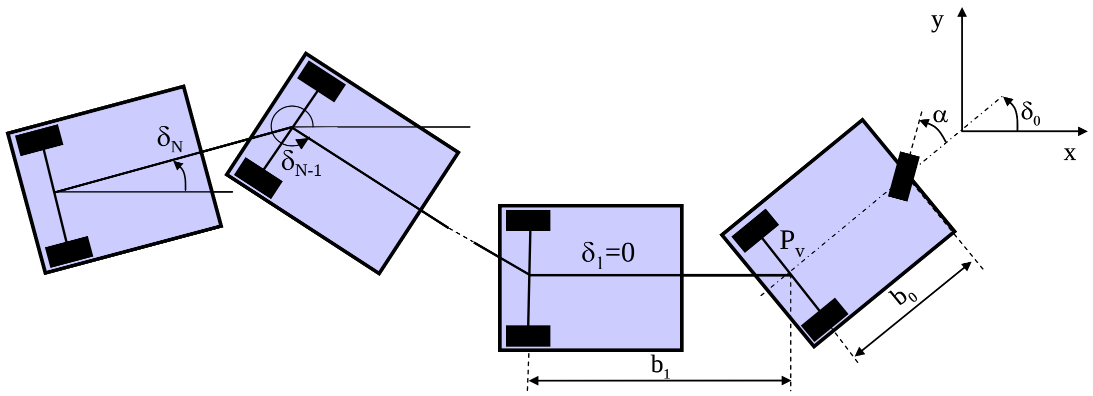
Where the controls chosen in this case, the most commonly used, are the speed of *traction* of the *front wheel* and the steering speed. If instead the system is controlled via the speed of *traction* of the rear *axis* it is enough to replace to the u_1 control $u_1' = u_1\cos\alpha$ and the model becomes:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\delta} \\ \dot{\alpha} \end{bmatrix} = \begin{bmatrix} \cos\delta \\ \sin\delta \\ \dfrac{\tan\alpha}{b} \\ 0 \end{bmatrix} u_1' + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} u_2 \quad (2\text{-}11)$$

As expected, the first two equations of the model (2-11) are those of the unicycle but in this case the speed of balance configuration is dependent on the steering speed. It is also important to notice that the latter model has a singularity in $\alpha = \pm\pi/2$ which corresponds to the situation the steering wheel is arranged perpendicular to the longitudinal axis. This uniqueness however does not appear in the model (2-10) with front wheel drive.

Another important consideration is that the resulting models, when we take as reference the average point of the rear wheels, are valid not only for the three-wheeled vehicle but also for the quadricycle system (car type). In fact both can be summarized in a system in which the wheels, both the front steering ones, and the rear ones, coincide with only two virtual wheels, a steering front one and another rear one, placed on the axis of longitudinal symmetry, i.e. the bicycle system (Picture 6). In the casa of the quad system the steering wheels must always be adjusted according to the steering angle α of the virtual steering wheel, so as to identify a single centre of instantaneous rotation of the vehicle (Picture 5).

2.5. Car model with N trailers



Picture 7: Car model with N trailers

A more complex model than the one just seen is obtained by adding N trailers to a tricycle robot or car model with rear-wheel drive. The kinematically easiest model predicts that each trailer is hooked in the middle of the rear axle of the preceding one. One can choose as a vector of generalized coordinates the one of the car model to which the trim of each trailer angle is added, measured between its own axis of symmetry and the axis x (Picture 7):

$$q = [x, y, \alpha, \delta_0, \delta_1, ..., \delta_{N-1}, \delta_N]^T$$

The size of the space of generalised coordinates, the number of degrees of freedom, is therefore N+4.
The non-holonomic constraints are two for the motor and other N constraints for the rear axle of the i-th trailer, the coordinates of which are obtained by:

$$\begin{cases} x_i = x - \sum_{j=1}^{i} b_j \cos\delta_j \\ y_i = y - \sum_{j=1}^{i} b_j \sin\delta_j \end{cases} \qquad i = 1,.., N$$

The N+2 constraints are:

constraint for the front axle of the motor

$$\dot{x}_a \sin(\delta_0 + \alpha) - \dot{y}_a \cos(\delta_0 + \alpha) = 0$$

constraint for the rear axle of the motor

$$\dot{x} \sin\delta - \dot{y} \cos\delta = 0$$

constraints for the rear axle of the i-th trailer

$$\dot{x}_i \sin\delta_i - \dot{y}_i \cos\delta_i = 0 \qquad i = 1,.., N$$

Expressing the constraints only in function of the generalized coordinates we obtain:

$$\begin{aligned}
&\dot{x}\sin\delta_0 - \dot{y}\cos\delta_0 = 0 \\
&\dot{x}\sin(\delta_0+\alpha) - \dot{y}\cos(\delta_0+\alpha) - \dot{\delta}_0 b_0\cos\alpha = 0 \\
&\dot{x}\sin\delta_i - \dot{y}\cos\delta_i + \sum_{j=1}^{i}\dot{\delta}_j b_j\cos(\delta_i-\delta_j) = 0 \qquad i=1,..N
\end{aligned} \tag{2-12}$$

Rather than finding the core associated with the constraint matrix to obtain the kinematic model of the control system, you can directly write the kinematic equations:

$$\begin{aligned}
&\dot{x} = v\cos\delta_0 \\
&\dot{y} = v\sin\delta_0 \\
&\dot{\delta}_o = \frac{v\tan\alpha}{b_0} \\
&\dot{\delta}_i = -\frac{1}{b_i}v_{i-1}\sin(\delta_i-\delta_{i-1}) \\
&v_i = v_{i-1}\cos(\delta_i-\delta_{i-1}) \qquad i=1,..,N
\end{aligned}$$

Where v_i It is the linear speed of the rear axle (midpoint) of the i-th trailer. Finally, the kinematic model:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\alpha} \\ \dot{\delta}_0 \\ \dot{\delta}_1 \\ \dot{\delta}_2 \\ \vdots \\ \dot{\delta}_i \\ \vdots \\ \dot{\delta}_N \end{bmatrix} = \begin{bmatrix} \cos\delta_0 \\ \sin\delta_0 \\ 0 \\ \tan\alpha/b_0 \\ -\sin(\delta_1-\delta_0)/b_1 \\ -\cos(\delta_1-\delta_0)\sin(\delta_2-\delta_1)/b_1 \\ \vdots \\ -\left(\prod_{j=1}^{i-1}\cos(\delta_j-\delta_{j-1})\right)\sin(\delta_i-\delta_{i-1})/b_i \\ \vdots \\ -\left(\prod_{j=1}^{N-1}\cos(\delta_N-\delta_{N-1})\right)\sin(\delta_N-\delta_{N-1})/b_N \end{bmatrix} u_1 + \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} u_2 \tag{2-13}$$

The controls are still two as in the case of the car model, the rear driving wheel speed u_1 and the steering speed u_2.

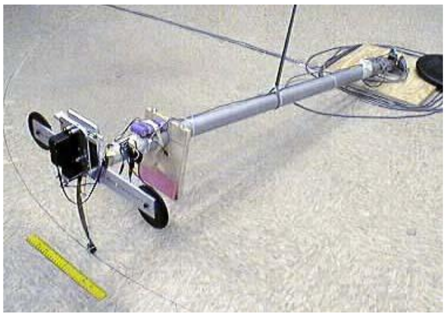
2.6. Holonomic and non-holonomic systems

HOLONOMIC SYSTEMS	NON-HOLONOMIC SYSTEMS
Example*: Manipulators with revolute and/or prismatic pairs*	Example*: vehicles on wheels with pure rolling*
Kind of constraints and number of applicable controls	
N bodies+ *p* simple geometric constraints (eg. motion on the plane)	*N* bodies+*p* simple geometric constraints
$\Downarrow$	$\Downarrow$
$N' = 6 * Np$ degrees of freedom (df)	$N' = 6 * Np$ dof
$\Downarrow$	$\Downarrow$
N' independent speeds	*N'* independent speeds
N' df+*k* kinematic constraints **which can be integrated**	*N* bodies+ *k* kinematic constraints **which cannot be integrated**
$\Downarrow$	$\Downarrow$
$n = N'\text{-}k$ *df*	$n = N'$*configuration variables* (Vdc)
$\Downarrow$	$\Downarrow$
q_i, $i = 1,..n$ *of coupling variables* (or generalised coordinates)	q_i, $i = 1,..n$*configuration variables* (Or generalised coordinates)
$\Downarrow$	
n independent speeds	$m = n\text{-}k$ independent speeds
$\Downarrow$	$\Downarrow$
$m = n$ controls *non differential*	$m < n$ controls *differential* (Kinematic, starting from speeds)
Motion planning	
Desired Cartesian trajectory $\underline{x}_d(t)$	Desired Cartesian trajectory $\underline{x}_d(t)$
$\Downarrow$	$\Downarrow$
Calculating the trajectory in the coupling space*:* $\underline{q}(t) = f(\underline{x}_d(t))$	Calculating the *m* desired control speeds $v_d(t) = \underline{v}_d = g(\dot{\underline{x}}_d(t))$

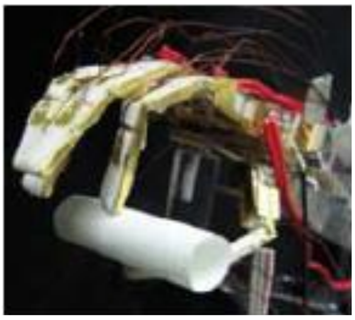
2.7. Other examples of non-holonomic systems

The non-holonomic constraints can be caused by several factors, of a physical nature or related to the type of control, some of them are:

1. *Surfaces that roll without sliding*
   Examples:
   - in addition to wheeled mobile robots, the manipulation of objects with robotic fingers (Picture 9)
2. *Conservation of angular momentum in multibody systems*
   Examples:
   - floating manipulators in space without external actuators
   - dynamically balanced jumping robots(Picture 8)
   - divers and astronauts in mid air
   - satellites with balance stabilisation



Picture 8: Robot hopping: Jumping robot



Picture 9: Robotic hand with five fingers

3. *Special control tasks*
    - redundant robots (eg with 3 degrees of freedom on the plan but target with no specifications on the balance)
    - underwater robots (6 generalised coordinates, 4 speed inputs)

3. PLANNING AND PATH CONTROL

The planning objectives of motion differ depending on whether it is high-level or low level.

The low-level goals are the motion planning strategies for achieving the high level primary objectives

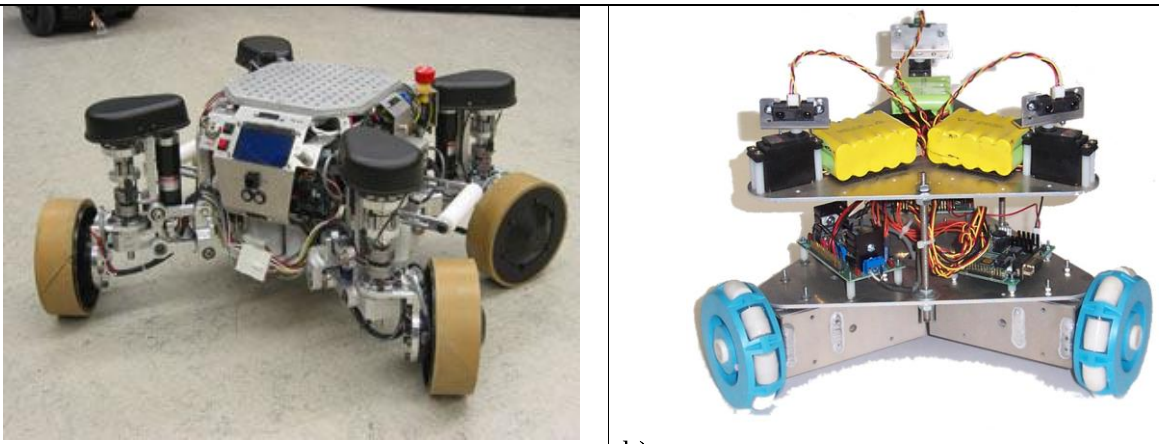
Low-level motion strategies can be:

- Geometric paths
- Trajectories, ie parameterised paths over time
- Sequences of movement commands based on information of the sensors which the robot has

While the primary objectives are:

- Move toward the target without colliding with obstacles (any environment)
- Build a map of the environment (unstructured)
- Finding an object (in a non-structured or semi-structured environment)

The most difficult task is a low-level planning of a compatible path (permissible) with non-holonomic kinematic constraints. While for an omnidirectional robot (Picture 10) any path is admissible, for non-holonomic robot a certain degree of continuity in the functions that geometrically define the path is required. For example, the excellent routes for the vehicle of Reeds and Shepp [21] provide a discontinuity in the curvature (on which a greater limit is imposed because of the limit on the angle of steering) and may be carried out from the rear axle of a vehicle which will only turn on condition that the vehicle stops in the points of discontinuity. For this type of vehicle, of course, the paths with discontinuity in the tangent are not allowed or, if planned, cannot be carried out even with ideally zero error. It is also important to analyze the feasibility of the planned trajectories and the controllability of the vehicle, the choice of the reference point for which the path and trajectory are planned.



a) b)

Picture 10: a) robot with four steering and driving wheels. 8 degrees of freedom of control. b) robot with three wheels, each can move transversely without sliding.

3.1. Admissible paths and trajectories

The space of a mobile robot configurations is defined by the minimum number of parameters (generalised coordinates) that allow to locate the entire system in its environment (for example the vector $[x, y, \delta, \alpha]$ in the case of the model treated in §2.4). As for the manipulators it is common to distinguish between Cartesian trajectories and trajectories in the coupling space, for mobile robots *a trajectory is defined* as a given continuous vector function in time which consists of m as many scalar functions, as the system checks (eg speed of traction and steering angle). A trajectory is *admissible* if and only if, given the initial and final conditions of the configuration, it is a solution of the system of differential equations which constitute the kinematic model of the robot. A *route* is what in the space of configurations corresponds to a given trajectory. In other words, if you apply a given law of motion (trajectory) to the robot, what results in Cartesian space is a path. An admissible path corresponds to an admissible trajectory.

The mathematical relationships between path and trajectory are not trivial for a non-holonomic system because of the non-integrability of the differential equations system of the kinematic model. In fact, for example, for the car model in which the vector of configuration variables is $[x, y, \delta, \kappa]$ and it consists of the Cartesian coordinates (x, y) the average point of the rear axle, of the balance δ and of the curvature κ, you have:

$$\begin{cases} \dot{x}(t) = v(t)\cos\delta(t) \\ \dot{y}(t) = v(t)\sin\delta(t) \\ \dot{\delta}(t) = \kappa(t)v(t) \\ \dot{\kappa}(t) = u(t) \end{cases} \Leftrightarrow \begin{bmatrix} \dot{x}(t) \\ \dot{y}(t) \\ \dot{\delta}(t) \\ \dot{\kappa}(t) \end{bmatrix} = \begin{bmatrix} \cos\delta(t) \\ \sin\delta(t) \\ \kappa(t) \\ 0 \end{bmatrix} v(t) + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} u(t) \quad (3\text{-}1)$$

The controls are the speed of the rear axle, v, and the change of curvature with respect to time, u. Wanting to move from the temporal variable to t the curvilinear coordinate s just keep in mind that:

$$v(t) = \frac{ds}{dt}$$

$$\kappa(s) = \frac{d\delta(s)}{ds} \Rightarrow \dot{\delta}(t) = \kappa(t)\frac{ds}{dt} = \kappa(t)v(t)$$

So the model becomes:

$$\begin{cases} \dot{x}(s) = \cos\delta(s) \\ \dot{y}(s) = \sin\delta(s) \\ \dot{\delta}(s) = \kappa(s) \\ \dot{\kappa}(s) = u(s) \end{cases}$$

If finally the equations of the model are integrated you obtain

$$\begin{aligned} x(s) &= \int_0^s \cos\delta(s)ds \quad ; \quad \delta(s) = \int_0^s \kappa(s)ds \\ y(s) &= \int_0^s \sin\delta(s)ds \quad ; \quad \kappa(s) = \int_0^s u(s)ds \end{aligned} \tag{3-2}$$

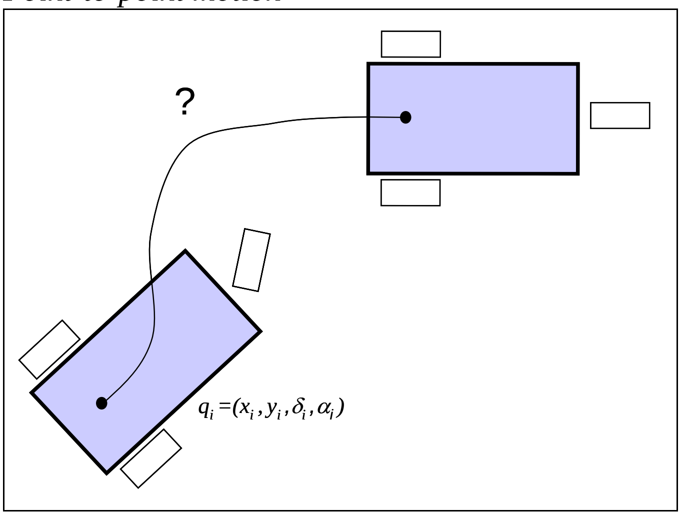
These equations, while being integrated in numerical form and used as a template for the odometry reconstruction (applied controls) are not a solution of the system of differential equations in the classical sense because the balance, which is one of the configuration variables (status) of the system, appears within the integrals; in other words to obtain the solution of the system, given the initial and final conditions, one of the variables should be integrated.

3.2. Path following and Trajectory tracking

A further definition of trajectory is the route to which the temporal law with which the path is followed by the robot is associated. This meaning is used to distinguish between three classes of problems based on the task assigned to the robot:

- *Point-to-point motion*
- *Route tracking (path following)*
- *Trajectory tracking (trajectory tracking)*

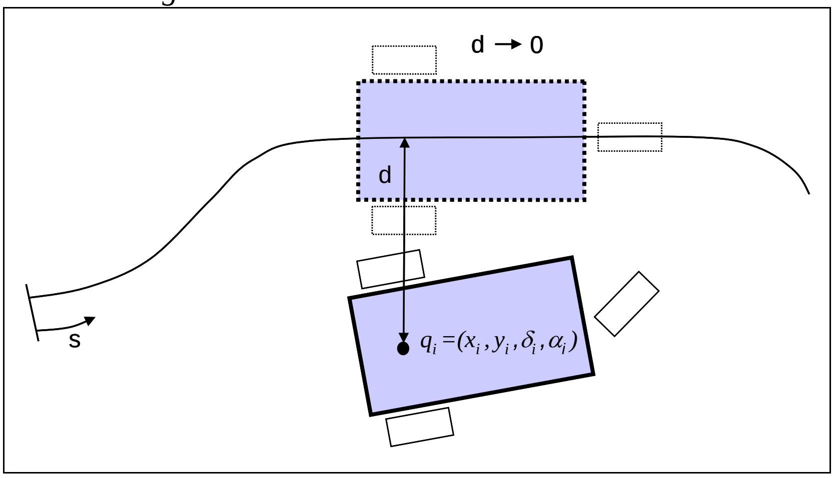
Point-to-point motion



Picture 11: Point-to-point motion. There are no specifications on the trajectory but only on the initial and final configurations

The robot must reach a final position (position and balance) starting from a given initial position. In general it must reach a final configuration given an initial configuration in the space of generalised coordinates. The problem, from the control point of view, is to stabilize the robot in a point of equilibrium in the configuration variables space. For a car type vehicle only two controls are available, for example speed of front wheel drive and steering speed, in order to change the four configuration variables (x, y, δ, α). The situation remains the same even for a vehicle with N trailers in which you have two inputs and N+4 configuration variables. Any feedback on the control uses as error the difference between the current configuration and the desired one.

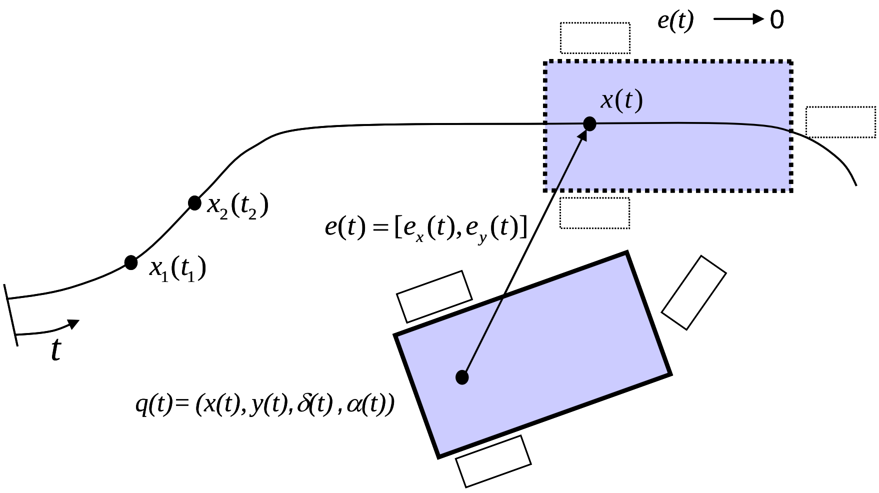
Path tracking



Picture12: Path tracking. The objective is to cancel the distance from the trajectory regardless of time

The robot must reach and follow a path in Cartesian space starting from a given initial configuration inside or outside the path. The control algorithm is based on the geometrical description of the Cartesian trajectory, usually parameterised according to the curvilinear coordinate s. The control temporal law is not specified since the main goal is to approach as much as possible the vehicle to the trajectory reducing the distance from it. Therefore, the evolution of the parameter s over time can be chosen arbitrarily and the two inputs can be deducted with respect to time without changing the path. A choice may be to maintain the speed input of the traction constant or to vary it according to any temporal law, and to carry out the tracking control by acting only on the angle of steering.

Trajectory tracking



Picture 13: Trajectory tracking The task is to track the path according to an imposed temporal law

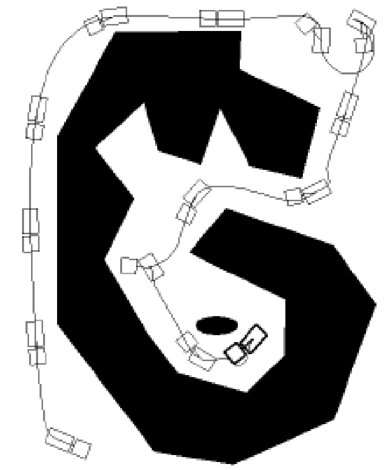
In this case the robot has to reach and follow a trajectory in Cartesian space starting from a given initial configuration in or out of the path, or a path with an imposed temporal law. One may think that the robot has to chase and reach a virtual robot that moves along the desired path with variable speed according to a desired law. From the control point of view, the objective is to minimize the Cartesian error $e(t)$ between the actual position of the robot and the one provided for each moment of time t.

3.3. Path planning and motion planning

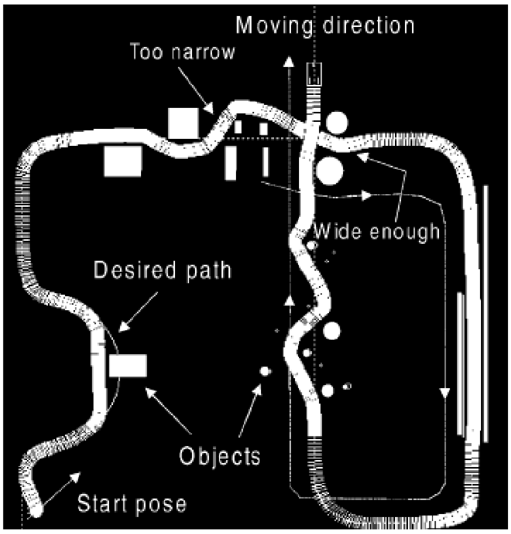
As said, the non-holonomic schedule of motion corresponds to the planning of admissible trajectories and may be assimilated to the problem of the open-loop controls planning to obtain a given path, that in the case of the manipulators corresponds to an inverse kinematics problem. The problem, upstream, is to plan a route compatible with both non-holonomic constraints and with the presence of obstacles potentially interposed between the starting and landing point. This problem which applies to autonomous vehicles is called of *steering*.

In a structured environment the route and motion planning can be calculated beforehand at global *level*, i.e. including the complete task (Picture 14). However there is the need for real-time plan of the route at local level (In a certain surrounding of the robot) to allow the robot to work around unexpected obstacles (Picture 15) Or pitchfork a pallet in a position not known beforehand.

For a point-to-point motion, the problem can be solved directly only by calculating a law of motion that allows to guide the robot from the starting configuration to the arrival one. In this way, the path appears to be an output of the planning algorithm and can not be set up as input, there are no physical obstacles as constraints but however virtual obstacles must be taken into account, such as the limitation on the angle of the steering angle ($|\alpha| < \alpha_{max} \leq \pi/2$). The transformation of generalized coordinates in coordinates "*chained form*"(See §3.4.1), when the system is convertible, allows the problem to be easily fixed.



Picture 14: Example of route planning for a vehicle with trailer.



Picture 15: Route planning and bypass of unforeseen obstacles.

If it is required that the robot follows a given path (admissible, free of obstacles, optimized for minimum travel time, minimum angular acceleration, etc.) one of the following strategies must be followed:

A. Tracking the path or the Cartesian trajectory with heuristic algorithms (see §3.5.4)
B. Plan the notion with theoretical and mathematical methods (see §3.4) Calculating the desired law of motion

C. Integrating method B. with a feedback control algorithm of the law of motion planned to take into account the possible tracking errors and reduce them (see §3.5.2).

3.4. Methods of motion planning for non-holonomic systems

3.4.1. "Chained form" systems.

There are canonical forms for kinematic models that are functional for the development of both open loop and feedback control strategies. The most used canonical form is the *chain shaped* or "chained form". A chain system is in the following form:

$$\begin{cases} \dot{x}_1 = u \\ \dot{x}_2 = f_2(x_1)u \\ \dot{x}_3 = f_3(x_1, x_2)u \\ \vdots \\ \dot{x}_p = f_p(x_1, x_2, .., x_{p-1})u \end{cases}$$

Where $x_i \in \mathbb{R}^{m_i}$ the system configuration variables and $\Sigma_i m_i = n$, the sum of their sizes is equal to the number of degrees of freedom.

For a two input control system the canonical form is called *chain shaped single* $(2, n)$:

$$\begin{cases} \dot{x}_1 = u_1 \\ \dot{x}_2 = u_2 \\ \dot{x}_3 = x_2 u_1 \quad (3\text{-}3) \\ \vdots \\ \dot{x}_n = x_{n-1} u_1 \end{cases}$$

Where it is assumed that all x_i are deducted sizes. The input u_1 is called generator input and the variables x_1 and x_2 are called basic variables. Note that if u_1 is constant (or constant in intervals), the system is linear (or sometimes linear). It is shown that the chain systems of form $(2, n)$ are completely controllable. They are characterized by a particular property called *nilpotent* [14].

There are necessary and sufficient conditions for a two-input system to be transformed into a chained form by:

1. a change of coordinates $x = \phi(q)$
2. a reversible transformation of inputs $v = \beta(q)u$.

It is shown that a non-holonomic system with $m = 2$ input and $n = 3,4$ generalized coordinates can always be put into a chain form. Furthermore, the models of vehicles with trailers N, if with each trailer hinged in the midpoint of the rear axle of the previous one, may be transformed in chained form. In the case where there is an offset in the hinge, only models of vehicles with less than two trailers can be transformed into chained form.

For example, consider the kinematic model (2-3) of the unicycle. Using the coordinate transformation

$$
\begin{aligned}
x_1 &= -\delta \\
x_2 &= x\cos\delta + y\sin\delta \\
x_3 &= -x\sin\delta + y\cos\delta
\end{aligned}
$$

and the input transformation

$$
\begin{aligned}
v_1 &= x_3 u_1 + u_2 = (-x\sin\delta + y\cos\delta)u_1 + u_2 \\
v_2 &= -u_1
\end{aligned}
$$

is quick to verify, remembering that $\dot{x} = v_1\cos\delta$ and $\dot{y} = v_1\sin\delta$ that the transformed system results into a chain canonical form. It is worth noting that the new variables x_2 and x_3 are simply the Cartesian coordinates of the unicycle expressed in the coordinate system integral to the robot and oriented with the axis x_2 aligned with the axis with respect to which the balance of the vehicle is measured.

As regards to the tricycle model or, equivalently, car, the following change of coordinates can be used:

$$
\begin{aligned}
x_1 &= x \\
x_2 &= \frac{\tan\alpha}{b\cos^3\delta} \\
x_3 &= \tan\delta \\
x_4 &= y
\end{aligned}
\tag{3-4}
$$

and the input transformation

$$
\begin{aligned}
v_1 &= \frac{u_1}{\cos\delta} \\
v_2 &= -3\sin\delta\sin^2\alpha\,\frac{u_1}{b\cos^2\delta} + b\cos^3\delta\cos^2\alpha\,u_2
\end{aligned}
\tag{3-5}
$$

for which the system assumes the chain form

$$
\begin{cases}
\dot{x}_1 = u_1 \\
\dot{x}_2 = u_2 \\
\dot{x}_3 = x_2 u_1 \\
\dot{x}_4 = x_3 u_1
\end{cases}
\tag{3-6}
$$

In this case the transformation of coordinates, and therefore the chain shape, is defined for $\delta \neq \pi/2 \pm k\pi,\ k \in \mathbb{Z}$.

It occurs that, as expected, also the single chain system is non-holonomic but in this case *some procedures* can be used which are appropriate to obtain classes of solutions to the problem of point-to-point motion planning, or trajectories (motion laws) that drive the robot from an assigned starting configuration to an assigned landing configuration. It is possible to use different types of inputs ($u(t)$):

- inputs *sinusoidal*
- inputs *constant in intervals*

- inputs *polynomial*

Sinusoidal inputs

Choosing for the two inputs

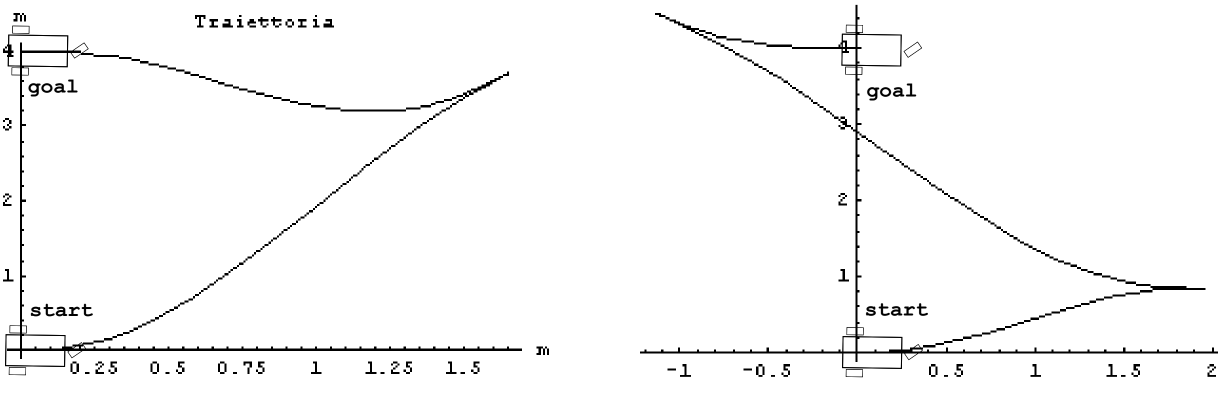
$$\begin{aligned} u_1 &= a_0 + a_1 \sin(\omega t) \\ u_2 &= b_0 + b_1 \cos(\omega t) + b_2 \cos(2\omega t) + ... + b_{n-2} \cos[(n-2)\omega t] \end{aligned} \quad (3\text{-}7)$$

$n+1$ variables to change n the coordinates generalised from a given initial state to a given final condition in a given time T are generated By integrating (from the first to the last cascade equation) the differential equations of the model (3-3) in which the inputs in the sinusoidal shape (3-7) are substituted, a linear system which has as variables the $n+1$ coefficients is obtained. It is shown [27] that, if $a_1 \neq 0$, the system is invertible.

In the specific case of pattern (3-6) there are four coordinates and five parameters (a_0, a_1, b_0, b_1, b_2). A possible termination procedure [18] is as follows

- the initial and final transformed coordinates are calculated
- the cascade equations starting from the first in the time interval are integrated $T = 2\pi / \omega$
- a value of is chosen a_1
- the system is solved by calculating the other four coefficients (a_0, b_0, b_1, b_2).

When the imposed time T and the parameter change a_1, the geometry of the planned route will change. In Picture16 there is an example of application (using the symbolic computation Mathematica code®) With two different values of a_1. Furthermore, the geometry of the path is not invariant to rotation, i.e. it depends on the initial and final balance δ_0 and δ_f and not only on the difference $\delta_0 - \delta_f$.



Picture16: Motion planning with the same initial and final conditions but with different values of a_1

Inputs constant in intervals

The total motion time is split T into subintervals of length Δ in each of which the two inputs are constant

$$\begin{aligned} u_1(\tau) &= u_{1k} \\ u_2(\tau) &= u_{2k} \end{aligned}$$

with $\tau \in [(k-1)\Delta, k\Delta]$

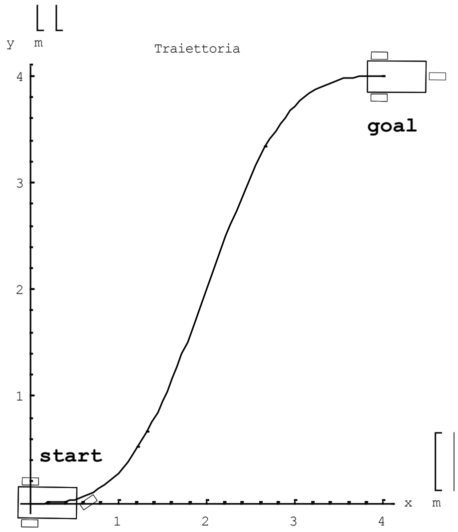
If it is u_1 always kept constant and $n-1$ subintervals are taken ($k = 1,..,n$ -1) So that

$$(n-1)\Delta = T$$

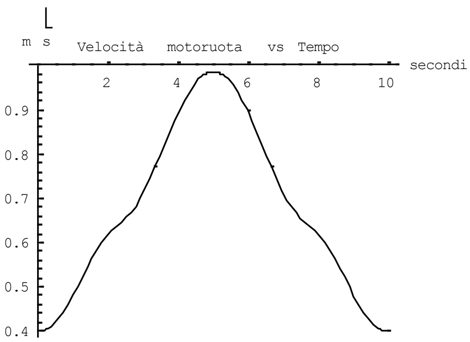
and

$$u_1 = \frac{x_{f1} - x_{01}}{T}$$

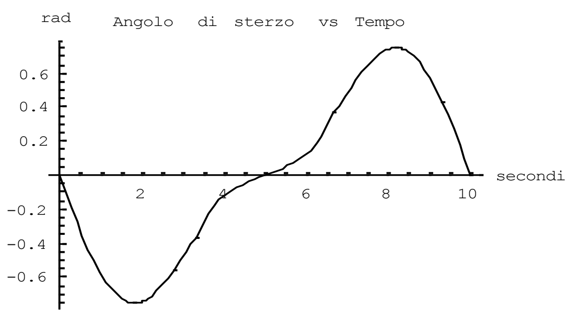
where n is the number of generalized coordinates, x_{f1} and x_{01} they are respectively the final and the initial value of the variable x_1, the $n-1$ constant values ($u_{2,1}, u_{2,2}, .., u_{2,n-1}$) of the input u_2 are obtained by solving a triangular linear system that results from the integration of the equations of the chain model. In the event that the horizontal axis of the starting and end point are the same, an intermediate point needs adding, and it is not possible to make automatic manoeuvres with speed inversion.



Picture 17: Trajectory obtained with inputs which are constant in intervals



Picture 18: Speed input of the motor wheel



Picture 19: Steering angle control. Does not have a constant speed (tangent to the graph)

In Picture 18 and in Picture 19 it is evident how the inputs of the traction speed and of the steering angle are continuous functions but the initial and final conditions on v and α are free because they are not imposed in the decisive equations. Furthermore, the steering speed (the derivative of the graph of Picture 19) may have discontinuities.

Polynomial inputs

An algorithm similar to that of constant in intervals input, but with better performance in the level of continuity ("smoothness") of the input functions, it is to use polynomial in the input.

Choosing for the two inputs:

$$u_1 = sign(x_{f1} - x_{01})$$
$$u_2 = c_0 + c_1 t + .. + c_{n-2} t^{n-2}$$

As the total time you can choose

$$T = x_{f1} - x_{01}$$

You will get a linear system in the variables ($c_0, c_1, .., c_{n-2}$) obtained by integrating the equations of the chain model, of the type

$$A(T)\begin{pmatrix} c_0 \\ c_1 \\ \vdots \\ c_{n-2} \end{pmatrix} + f(u_i, T) = \begin{pmatrix} u_{f2} \\ u_{f3} \\ \vdots \\ u_{fn} \end{pmatrix}$$

Where the matrix $A(T)$ is invertible for $T \neq 0$

In the event that the horizontal axis of the starting and landing point are the same, an intermediate point needs adding, and it is not possible to make automatic manoeuvres with speed inversion.

3.4.2. Planning for open-loop trajectories

The methods seen so far allow the calculation of an admissible path and the relevant controls to bring a vehicle from a starting configuration to a landing one (point to point motion). By contrast, you cannot predict exactly what the planned route will be, which makes even more complex the task of moving the robot in an environment full of obstacles. It is therefore preferable, if the environment is structured, to plan a path free from obstacles and that it is also admissible, not only as regards to the kinematic constraints but also for what concerns the limits on the controls (maximum steering angle, maximum traction and steering speed, accelerations, etc.).

For the car model with rear-wheel drive **(**2-11**)** it is possible to calculate the traction and steering speed inputs (input trajectory) as a function of the desired Cartesian path, assuming that this is a admissible trajectory and with a certain degree of continuity [5]. It is sufficient to set the wanted Cartesian coordinates $\left(x_d(t), y_d(t)\right)$ in function of time starting from a certain initial moment t_0. The remaining desired configuration variables can be obtained from the Cartesian coordinates (balance and steering angle) and the associated input commands. This is possible because due to non-holonomic constraints the vehicle has to have speed and balance tangent to the Cartesian trajectory, and moreover the steering angle is geometrically related to the curvature of the path and the steering speed is linked to the desired angular acceleration. All these variables can be obtained mathematically from geometrical considerations and from derivatives in relation to the time of the Cartesian trajectory.

The desired Cartesian trajectory (geometric path in function of time) is admissible when it can be obtained as a result of evolution over time of the system configuration according to the equations of the kinematic model of the car:

$$\dot{x}_d = \cos\delta_d v_{d1} \tag{3-8}$$

$$\dot{y}_d = \sin\delta_d v_{d1} \tag{3-9}$$

$$\dot{\delta}_d = \frac{\tan\alpha_d v_{d1}}{b} \tag{3-10}$$

$$\dot{\alpha}_d = v_{d2} \tag{3-11}$$

with input $v_{d1}(t)$ and $v_{d2}(t)$ at least continuous in intervals.
v_{d1} It can be obtained from (3-8) and (3-9)

$$v_{d1}(t) = \pm\sqrt{\dot{x}_d(t) + \dot{y}_d(t)}$$

the sign of which depends on the choice of motion direction, forward or backward.
Dividing (3-9) by (3-8), the desired balance is obtained

$$\delta_d(t) = ATAN2\left(\frac{\dot{x}_d(t)}{v_{d1}(t)}, \frac{\dot{y}_d(t)}{v_{d1}(t)}\right) \tag{3-12}$$

Deriving the (3-8) and (3-9) with respect to time and multiplying the first by $\dot{y}_d$, the second by $\dot{x}_d$ and subtracting member by member one obtains

$$\dot{\delta}_d(t) = \frac{\dot{x}(t)\ddot{y}(t) - \ddot{x}(t)\dot{y}(t)}{v_{d1}^2(t)} \tag{3-13}$$

The result just obtained can be substituted in (3-10) and thus it is possible to obtain the steering angle the sign of which depends on the sign of the speed of traction

$$\alpha_d(t) = \arctan\left(b\frac{\dot{x}(t)\ddot{y}(t) - \ddot{x}(t)\dot{y}(t)}{v_{d1}^3(t)}\right) \tag{3-14}$$

Finally to obtain the input v_{d2} It is sufficient to deduce, with respect to time, the steering angle just obtained:

$$v_{d2}(t) = \frac{bv_{d1}[(\dddot{y}_d\dot{x}_d - \dddot{x}_d\dot{y}_d)v_{d1}^2 - 3(\ddot{y}_d\dot{x}_d - \ddot{x}_d\dot{y}_d)(\dot{x}_d\ddot{x}_d + \dot{y}_d\ddot{y}_d)]}{v_{d1}^6 + b^2(\ddot{y}_d\dot{x}_d - \ddot{x}_d\dot{y}_d)^2}$$

The equations just obtained biunivocally bind the Cartesian trajectory to the input trajectory, i.e. they allow to obtain the controls in function of the open loop time following which the robot follows the desired path with the imposed temporal law. The open loop control requires that the robot is placed exactly in the expected initial configuration ($x_d(t_0), y_d(t_0), \delta_d(t_0), \alpha_d(t_0)$) so that the output trajectory from the control system is effectively the desired one. The equations show that the inputs depend only on the desired output Cartesian trajectory and its derivatives up to the third order. Therefore to ensure the exact reproducibility of the desired trajectory it is necessary, in addition to the above condition on the initial configuration, that the trajectory is differentiable three times with respect to time.

It is important to take into consideration that the inputs contain a singularity, they are not defined in case $v_{d1}(\bar{t}) = 0$ for some $\bar{t} \geq t_0$. Because of this inconvenience, parameterised Cartesian trajectories are used as a function of a geometric parameter of the path, thereby the geometric description of the path is separate from the temporal one. If you indicate with σ the path parameter (the most used is the curvilinear coordinate s), the scheduling of Cartesian trajectory will imply a temporal law $\sigma(t)$ which can be introduced in the equation of the trajectory expressing the Cartesian coordinates in function of σ:

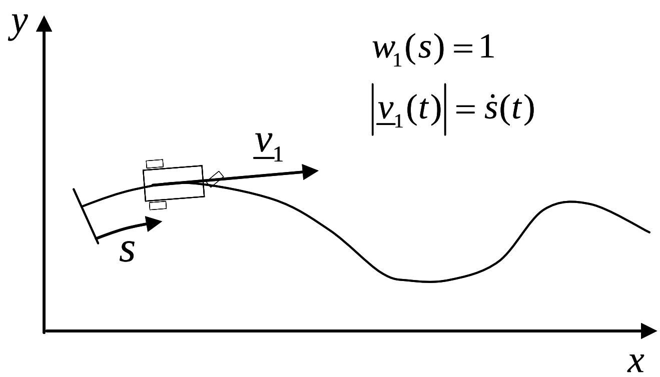
$$\begin{aligned} \dot{x}_d(t) &= \frac{d}{dt} x_d(\sigma(t)) = \frac{d}{d\sigma} x_d(\sigma(t))\dot{\sigma}(t) = x'_d(\sigma)\dot{\sigma}(t) \\ \dot{y}_d(t) &= y'_d(\sigma)\dot{\sigma}(t) \end{aligned} \qquad (3\text{-}15)$$

(the apex is used to distinguish the derivative with respect to the path parameter from the one with respect to time).
The linear velocity of the trajectory input in parameterized coordinates is called *pseudo-speed* and in a given point corresponding to a given σ is worth

$$w_{d1}(\sigma) = \pm\sqrt{x'^2_d(\sigma) + y'^2_d(\sigma)} \qquad (3\text{-}16)$$

while the actual speed command (reminding being the linear velocity of the reference point of the car model, i.e. the rear axle midpoint) is worth

$$v_{d1}(t) = w_{d1}(\sigma(t)\dot{\sigma}(t) \qquad (3\text{-}17)$$



Picture 20: Traction speed in function of the curvilinear coordinate. The pseudo speed is the module of the unit vector tangent to the trajectory, so it is always unitary.

If chosen $\sigma(t) = s(t)$, derivatives $x'(s) = dx(s)/ds$ and $y' = dy(s)/ds$ are the Cartesian components of the unit vector tangent to the trajectory at the point corresponding to s and therefore the pseudo-velocity, that is the module of this unit vector, will have unit value, therefore always different from zero.
The condition for which $v_{d1}(\bar{t}) = 0$ is obtained when $\dot{\sigma}(\bar{t}) = 0$ while $w_{d1}(\sigma) \neq 0$.

The desired balance expression in function of the path parameter is always defined without singular points:

$$\delta_d(\sigma) = ATAN2\left(\frac{\dot{x}_d(\sigma)}{w_{d1}(\sigma)}, \frac{\dot{y}_d(\sigma)}{w_{d1}(\sigma)}\right)$$

Due to the separation between time dependence and space dependence also the formulas $\alpha_d(\sigma)$ and $w_{d2}(\sigma)$ (with $v_{d2}(t) = w_{d2}(\sigma)\dot{\sigma}(t)$) have no singularity because there is only geometic information related to the curvature of the path and to the higher order derivatives with respect to the parameter σ. Indeed

$$\alpha_d(\sigma) = \arctan\left(b\frac{x'(\sigma)y''(\sigma) - x''(\sigma)y'(\sigma)}{w_{d1}^3(\sigma)} \right)$$

$$w_{d2}(\sigma) = \frac{bw_{d1}[(y_d'''x_d' - x_d'''y_d')w_{d1}^2 - 3(y_d''x'_d - x_d''y_d')(x_d'x_d'' + y_d'y_d'')]}{w_{d1}^6 + b^2(y_d''x_d' - x_d''y_d')^2}$$

The expression of the inputs in function of Cartesian trajectory of the rear-wheel drive car robot model (for the front wheel one the input transformation is immediate and is function only of the steering angle) can also be obtained for the chain model (2,4). The time evolution of the desired trajectory is described by the kinematic model in chained form

$$\begin{cases} \dot{x}_{d1} = u_{d1} \\ \dot{x}_{d2} = u_{d2} \\ \dot{x}_{d3} = x_{d2}u_{d1} \\ \dot{x}_{d4} = x_{d3}u_{d1} \end{cases} \quad (3\text{-}18)$$

From the desired trajectory in Cartesian coordinates $\left(x_d(t), y_d(t)\right)$ and from the change of coordinates (3-4) the desired trajectory in the variables transformed into a function of the Cartesian ones and their derivatives can be obtained

$$\begin{aligned} x_{d1}(t) &= x_d(t) \\ x_{d2}(t) &= \frac{\ddot{y}_d(t)\dot{x}_d(t) - \dot{y}_d(t)\ddot{x}(t)}{\dot{x}_d^3(t)} \\ x_{d3}(t) &= \frac{\dot{y}_d(t)}{\dot{x}_d(t)} \\ x_{d4}(t) &= y_d(t) \end{aligned}$$

and also the expression of the transformed inputs

$$\begin{aligned} u_{d1}(t) &= \dot{x}_d(t) \\ u_{d2}(t) &= \frac{\dddot{y}_d\dot{x}_d^2 - \dddot{x}_d\dot{y}_d\dot{x}_d - 3(\ddot{y}_d\dot{x}_d\ddot{x}_d - \ddot{x}_d^2\dot{y}_d)}{\dot{x}_d^4} \end{aligned}$$

3.4.3. Continuous curvature paths Clothoids.

It is well known, since the studies by Dubin and Reeds & Shepp, that excellent paths in terms of length, for a simplified car model (three configuration variables controlled in angular velocity) are made up by segments and circular arcs. One significant disadvantage of this class of paths is the presence of discontinuities of the curvature. These discontinuities occur in every transition among segments and arcs as the curvature jumps from zero to a non-null value. Since the curvature is linked to the steering angle by the relation

$$\kappa = \frac{\tan\alpha}{b}$$

if a vehicle carried out a path tracking, in order to perform exactly without errors the planned route it should stop at each discontinuity of curvature to be able to steer the front wheels (the front wheel in the case of tricycle model).

It is preferable, therefore, that the path is characterized by the continuity of curvature. It is then necessary, due to the mechanical limits on the angle of steering, that the curvature is limited. Furthermore, since the angular acceleration, i.e. the rate of change of the curvature, is linked to the vehicle's steering speed, it is preferable that the derivative of the curvature is limited on the upper part to ensure that the vehicle performs the trajectory with a certain pulling speed (proportional to the limit on the derivative of the curvature.

The problem of planning continues curvatures paths is part of the class of problems called "path smoothing", in which it is attempted to "smooth out" the discontinuity starting from a nominal path, such as a polygonal line, or by a sequence of path configurations to obtain a smooth curve.

The curves used can be divided into two categories:.

1. with coordinates having a closed-form expression, for example:
    a. B-spline
    b. Fifth degree polynomials
    c. Polar spline
2. parametric curves in which the curvature is a function of the length of the arc (curvilinear coordinate s):
    d. Clothoids
    e. Cubic spirals
    f. Spline G^2
    g. Intrinsic spline

Clothoids are parametric curves which have interesting features. They derive from the choice of a class of paths in which the curvature varies linearly with the length of the arc.

If it becomes necessary that the curvature follows the linear law,

$$\kappa = \frac{d\delta}{ds} = \lambda s + \kappa(0) \quad (3\text{-}19)$$

with λ constant and $\kappa(0) = c$ the initial curvature, for the Cartesian path is obtained

$$\begin{aligned}
\delta(s) &= \delta_0 + \int_0^s \kappa(\xi)d\xi = \delta_0 + \int_0^s (\lambda\xi + c)d\xi = \delta_0 + \frac{1}{2}\lambda s^2 + cs \\
x(s) &= x_0 + \int_0^s \cos\delta(\xi)d\xi = x_0 + \int_0^s \cos\left(\frac{1}{2}\lambda\xi^2 + c\xi\right)d\xi \\
y(s) &= y_0 + \int_0^s \sin\delta(\xi)d\xi = y_0 + \int_0^s \sin\left(\frac{1}{2}\lambda\xi^2 + c\xi\right)d\xi
\end{aligned} \quad (3\text{-}20)$$

Assuming the initial conditions null, in order to learn the Cartesian trajectory it is necessary to calculate the integrals

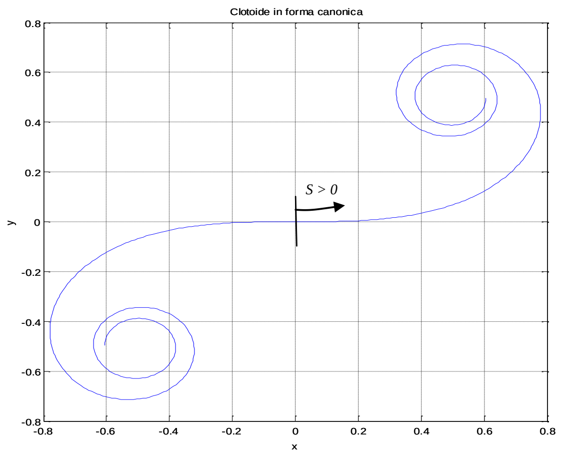
$$x(s)=\int_0^s \cos\left(\frac{1}{2}\lambda\xi^2\right)d\xi$$

$$y(s)=\int_0^s \sin\left(\frac{1}{2}\lambda\xi^2\right)d\xi$$

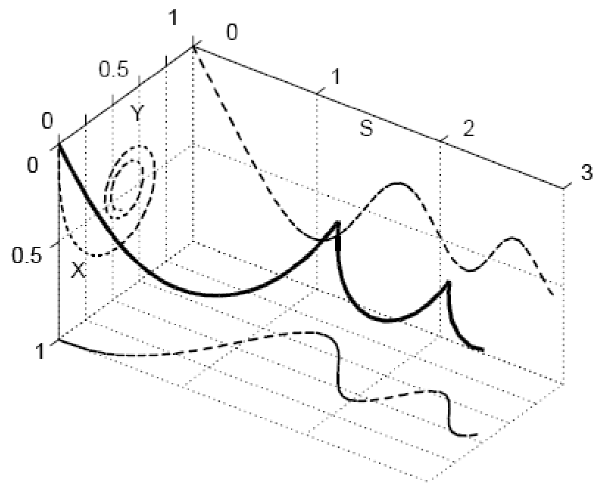
This type of integral has no closed form solution ($y=g(x)$), they are known as *Fresnel integrals*. The canonical form, respectively, of the cosine and the Fresnel sine is

$$CF(x)=\int_0^x \cos\left(\frac{\pi}{2}t^2\right)dt$$

$$SF(x)=\int_0^x \sin\left(\frac{\pi}{2}t^2\right)dt \qquad (3\text{-}21)$$



Picture 21: Chart of the clothoid in canonical form, with negative parameter (third quadrant) and positive (first quadrant). The graph is obtained from a linear variation of the curvature in *s*.



Picture 22: The parametric graph in space (x, y, s) is known as Cornu spiral and its projection in the plane (x, y) generates a clothed arc.

Note that the constant parameter λ is the angular acceleration expressed in geometrical terms, that is the derivative of the curvature with respect to the arc length.
The Fresnel integrals can be calculated by approximation with development in the power series, or by numerical methods.

As an application of route planning via clothoids the model (3-1) is to be considered. The system has two controls: the driving speed of the rear wheels, v; the angular acceleration γ which is directly linked to the steering speed $\dot{\alpha}$. If the speed is v constant the following relation is valid

$$\gamma(t)=\frac{d^2\delta}{dt^2}=\dot{\kappa}v=\frac{d}{dt}\left(\frac{\tan\alpha}{b}\right)v=\frac{\dot{\alpha}v}{b\cos^2\alpha} \qquad (3\text{-}22)$$

Instead in function of the curvilinear coordinate:

$$\gamma(s) = \frac{d^2\delta}{ds^2} = \frac{d\kappa}{ds} = \kappa'(s)$$

An upper limit to the curvature and its derivative is introduced

$$|\kappa| \le \kappa_{\max}\,;\ |\gamma| \le \gamma_{\max}$$

The limit on the curve is tied to the mechanical limit on the angle of steering, $\alpha_{\max}$, through the relation

$$\kappa_{\max} = \frac{\tan\alpha_{\max}}{b}$$

while the limit on the angular acceleration is tied to the limit on the steering speed and must be deducted according to the module of the speed v, in fact

$$\frac{\gamma_{\max}}{v} = \frac{\dot{\alpha}_{\max}}{b\cos^2\alpha_{\max}}$$

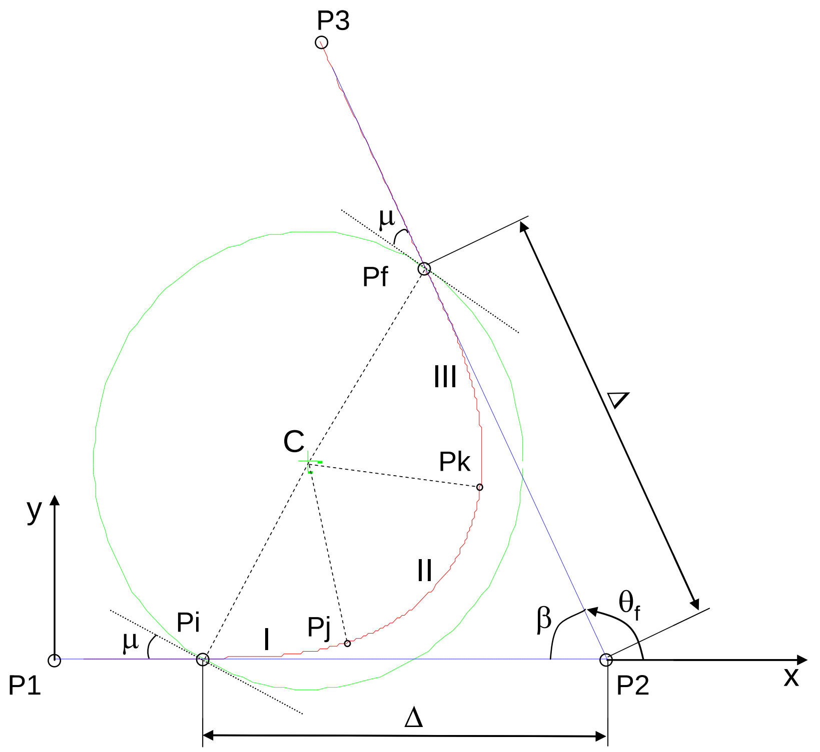
A value of $\gamma_{\max}$ valid for $|v| = 1$ can be calculated and, if a different value is chosen $v' \ne 1$, it is appropriate (necessary if $|v'| > 1$) to impose a new value $\gamma'_{\max} = \gamma_{\max}/|v'|$. In this way, by imposing a maximum value for the angular acceleration, the constraint on the maximum steering speed is still also respected, since $\dot{\alpha} \le \dot{\alpha}/\cos\alpha$, for any value of α. If the linear trend described by (3-19) is imposed to the bend the reports are valid:

$$\gamma = \dot{k}v = \frac{dk}{ds}v^2 = \lambda v^2 = \frac{\dot{\alpha}v}{b\cos^2\alpha} \quad \Rightarrow \quad \lambda = \frac{1}{v}\frac{\dot{\alpha}}{b\cos^2\alpha}$$

From which you can derive the maximum value of the curvature coefficient, it indicates how quickly the curvature of the clothoid increases with the increase of the arc length s

$$\lambda_{\max} = \frac{\gamma_{\max}}{v^2}$$

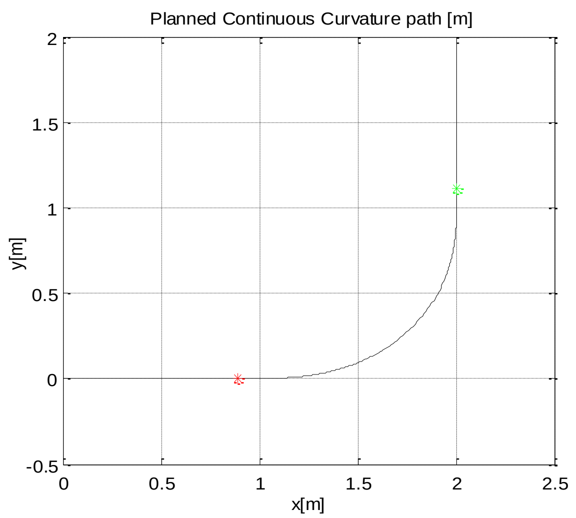
The clothoids allow to manage directly the curvature of the Cartesian trajectory and are suitable to be united with geometric continuity to circular arcs and straight line segments.



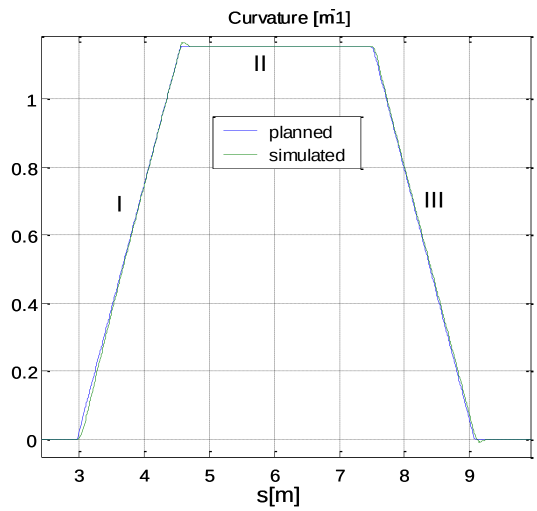
Picture 23: Building elements of continuous curvature trajectory. In red (continuous line) the curve, in green (dash-dot) the reference circumference of the curve. I and III are the traits of clothoid, II is any part of a circle.

A method [8] is described to build a continuous curvature path made up of segments, of clothoid arcs and circumference arcs (Picture 23).

Assume that the robot has to carry out a change of position passing from segment P1P2 to segment P2P3 which forms a certain angle with the first θ_f. This angle is also the overall change in the balance of the robot. The initial and the final curvature are both zero since the curve is preceded and followed by two segments. If the segments were linked only with an arc of circumference (which had $k \le k_{\max}$) the curvature would have two discontinuous jumps passing from zero (the initial segment P1P2) to a non-zero constant value (on the whole arc of circumference) and back to zero (on the final segment P2P3), this would imply an instantaneous change of the steering angle, which would be not feasible unless the vehicle did not stop in the points of discontinuity. Therefore two stretches of clothoid are introduced which are necessary to bring the curvature, in a linear way with the arc length *s* from the value zero to the maximum value (the first clothoid) and then, after a possible circular arc, from the maximum value to the zero value (the second clothoid). A characteristic trend of the curvature is reported in Picture 25.



Picture 24: Continous bending connection curve. Simulated path



Picture 25: Curvature trend of the path in function of the arc length. Comparison between planning and simulation

The circular arcs have a radius $k_{\max}^{-1}$ while clothoids have coefficient of curvature $|\lambda| \le \lambda_{\max}$.
The curve begins from point Pi, to which the configuration variables correspond ($x_s, y_s, \delta_i, 0$), and ends at point Pf, to which the configuration variables correspond ($x_f, y_f, \delta_i + \theta_f, 0$). Without losing general information it is convenient to use a relative reference system in such a way that $Pi \equiv (0,0,0,0)$ and assume that the curve is traversed in a counter clockwise direction.

The first portion of the curve consists of a clothoid arc with curvature and length $\lambda_{\max}$ coefficient $k_{\max} / \lambda_{\max}$ it allows to go from zero curvature in Pi, to curvature $k_{\max}$, up to Pj. The Pj configuration variables are

$$Pj = \begin{cases} x_j = \sqrt{\pi / \lambda_{\max}}\, CF\left(\sqrt{k_{\max}^2 / (\pi \lambda_{\max})}\right) \\ y_j = \sqrt{\pi / \lambda_{\max}}\, SF\left(\sqrt{k_{\max}^2 / (\pi \lambda_{\max})}\right) \\ \delta_j = k_{\max}^2 / (2\lambda_{\max}) \\ \kappa_j = k_{\max} \end{cases}$$

The first two equations are obtained from a simple change of variable in Fresnel's canonical integrals (3-21).

The second portion of the curve consists of a circular arc of radius $k_{\max}^{-1}$ ending in the Pk configuration = ($x_k, y_k, \delta_k, k_{\max}$). The centre C of the circle of which the arc is part of is located at a distance $k_{\max}^{-1}$ from Pk in a direction orthogonal to δ_k, and it has coordinates

$$\begin{cases} x_C = x_j - k_{\max}^{-1} \sin \delta_j \\ y_C = y_j + k_{\max}^{-1} \cos \delta_j \end{cases}$$

The third and last part of the curve consists of a clothoid arc with curvature $-\lambda_{\max}$ and length coefficient equal to that of the first section, it allows to pass from curvature $k_{\max}$, in Pk, to zero curvature, until reaching the final configuration Pf.

While the clothoid arcs have constant length as always characterized by a change of curvature $k_{\max}$ with curvature coefficient $\lambda_{\max}$, the arc of circumference can vary its length and it determines the final position of the whole curve. Its length is reduced to zero when the total deflection is equal to twice the balance variation which is obtained with each of the two sections of the clothoid and that is when θ_f

$$\theta_f = \theta_{\min} = \frac{k_{\max}^2}{\lambda_{\max}}$$

It is observed that the place of accessible configurations is a circumference with a radius

$$r = \sqrt{x_C^2 + y_C^2}$$

and that the angle between the balance direction of the final configuration Pf and the tangent to the circumference Pf is constant and valid

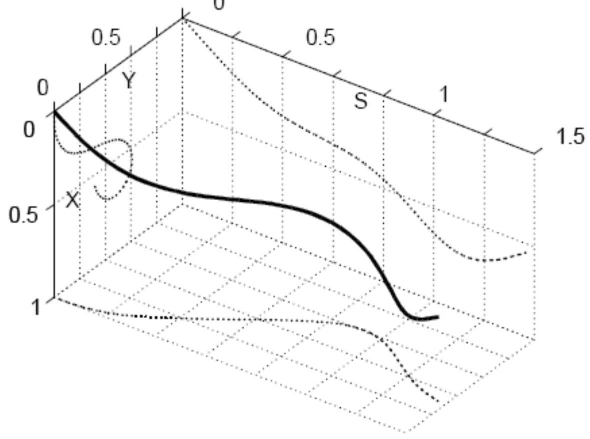
$$\mu = \arctan(\frac{x_C}{y_C})$$

If it is required to reach a final position with alignment $\delta_f < \theta_{\min}$, The curve will be devoid of an arc of circumference and will consist of two clothoids with curvature coefficient calculated so as to obtain the desired alignment variation.

This method can be used, in combination with other suitable algorithms, for the local route planning and allows to generate continuous curvature admissible paths taking into account the constraints on the maximum steering angle and the desired maximum angular acceleration. The feature of continuous curvature demonstrates the following advantages:

- respect for non-holonomic constraints and the maximum steering speed is to minimize any slippage of the wheels resulting in increased accuracy for the measurement with encoder
- the continuity of the curvature and the upper limit on the angular acceleration improve and make the performance of any laser triangulation sensor more stable, the quality of the measurement of which falls in the presence of abrupt balance changes.

3.4.4. Polynomial curvature paths



Picture 26: Polynomial Spiral (or generalized Cornu spiral) in the space (x, y, s). Its projection in the plane (x, y) represents the Cartesian path. In this example, the curvature is a third degree polynomial in s. The curvature changes sign at about mid path, which devolves toward the origin.

In the clothoid arc the curvature varies linearly with the length of the arc s

$$k(s) = a + bs$$

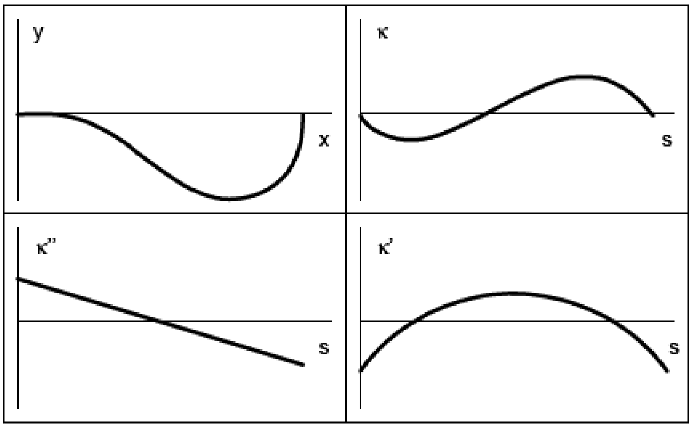
Please refer to the model (3-2) without the fourth equation, in which the curvature is used as an input in function of the arc length. This model has the advantage of not having to consider the dependence on time.
To connect two configurations, starting and ending:

$$q_0 = (x_0, y_0, \delta_0, \kappa_0)$$
$$q_f = (x_f, y_f, \delta_f, \kappa_f)$$

if only the relative changes of position are considered (x, y) and balance (in a reference system with the origin coinciding with the midpoint of the rear axle and the abscissa axis oriented as the structure of the vehicle, the initial configuration is $(0,0,0,\kappa_0)$) *five constraints* should be imposed, namely $[\kappa_0, x_f, y_f, \delta_f, \kappa_f]$.
With a clothoid arc only three constraints can be imposed, ie $[a, b, s_f]$ the two coefficients of the linear expression of the curvature and the total length of the path s_f. So with a clothoid arc it is possible to reach the desired final position starting from the initial curvature set but no constraints on balance and final curvature (or end position). Rather than using structured methods such as the one seen previously in §3.4.3, You can increase the degree of polynomial of the curvature [12], thus increasing the number of unknown coefficients with the possibility to respect a greater number of constraints on the boundary conditions.

To obtain a class of continuous curvature admissible paths that connect q_i with q_f simply use a third-degree polynomial

$$k(s) = a + bs + cs^2 + ds^3$$



Picture 27: The second derivative of the curvature with respect to s shows that the path in the upper left is obtained from a third degree polynomial spiral.

The obtainable paths belong to the class of so-called *polynomial spirals.*
The vector of parameters to be calculated will therefore be

$$\underline{p} = [a\, b\, c\, d\; s_f]$$

The coeffient *to* is calculated simply from the initial curvature: $a = \kappa(0)$. For the rest of the coefficients it is necessary, starting from the boundary conditions, to solve a nonlinear system. You can decouple the system by replacing the closed-form expression of the balance (analytically obtained from the first two equations) in the integrals of the two position equations *x* is *y*. You get the system

$$\begin{cases} k(s) = a + bs + cs^2 + ds^3 + ... \\ \delta(s) = as + \dfrac{bs^2}{2} + \dfrac{cs^3}{3} + \dfrac{ds^4}{4} + ... \\ x(s) = \displaystyle\int_0^s \cos(as + \dfrac{bs^2}{2} + \dfrac{cs^3}{3} + \dfrac{ds^4}{4} + ...)\, ds \\ y(s) = \displaystyle\int_0^s \sin(as + \dfrac{bs^2}{2} + \dfrac{cs^3}{3} + \dfrac{ds^4}{4} + ...)\, ds \end{cases}$$

in which the integrals are called *generalised Fresnel integrals*. These integrals are transcendental and the calculation should be performed with numerical methods.

An advantage of the polynomial spirals is that they allow to obtain the desired degree of continuity (for example on the continuity of the steering speed but also on the steering torque) depending on the degree of the polynomial. Also they simplify the solution of the system through the decoupling of the equations on the curvature and balance.

In choosing the optimal path between those calculated it is possible to minimize a desired cost function, an example can be the functional

$$J_k(\underline{p}) = \frac{1}{2}\int_0^{s_f} [\kappa(\underline{p})]^2 ds$$

3.5. Control of the trajectory

The planning of controls in open-loop is not robust with respect to deviations from the ideal system specifications (disturbance on the measurement, errors on the initial conditions, the model accuracy)

The feedback on the planned trajectory improves system performance by allowing to carry out the task assigned to the robot even in the presence of disturbances and deviations from the expected initial conditions. To estimate the state of the system, that is the installation of the robot, measurements from various sensors in real-time are used. In order to calculate the corrective input of the law of feedback, an estimate of the current configuration of the robot is necessary at all times. For this purpose, typically you have proprioceptive tools (encoders, gyroscope) and eteroceptive (laser, ultrasound) the measurements of which are combined together by a *sensor fusion algorithm* that performs, in every available, moment the best estimate (according to a given criterion) of the installation of the robot.

In a typical control architecture, a high-level planner manages the planning of routes free from obstacles and provides a set of goals to lower-level control. In this way the control of the trajectory task takes care of transforming the ideal path in a law of motion executable by the actuators. Also, if the planned route is not admissible with respect to non-holonomic constraints (discontinuity in the tangent, in the curvature, etc.), the feedback algorithm allows to recover the transient errors caused by geometric discontinuities.

From the control point of view, planning is equivalent to the open loop control or *feed forward,* and it constitutes a part of the feedback control. In fact, a closed-loop control algorithm is formed by the union of the action of feedback to a feed forward term. The latter is obtained from a priori knowledge of the environment and the objective of the task, while the feedback action depends on the current measured position of the robot and is calculated in real time based on the measurements of the sensors the robot is equipped with. However, the separation between the open-loop and closed-loop control strategies is not so clear: you can choose to either get a check in feedback via a real-time re-planning of the route based on the current location provided by the sensors, that is, taking into account in real time deviations from nominal conditions, tending to restore them

For mobile robots typically the planning phase ends with the calculation of a path which is kinematically admissible and free from obstacles, to which the open loop control inputs are associated. The admissibility is guaranteed if one takes into account the constraints of the non-holonomic system. It is also possible, in the planning stage, to choose a path that meets an excellent criterion together with constraints on the controls. One possible optimal criterion is the minimisation of a cost function, for example the determination of the path free from obstacles that makes minimum the length of the path or the maximum angular acceleration or a combination of the two. Among the constraints on the controls the maximum steering angle, the maximum linear and angular velocity must be taken into consideration. In any case, at this stage the controls are calculated off-line. So if you experience unexpected events such as wheel slippage or incorrect initial localization, these disrupt the proper implementation of point-to-point motion or tracking of path or trajectory

The solution is to resort to an estimated error feedback control and thus obtain a certain degree of strength of the overall control. The feedback control acts only in the case in which the error is not zero, assisting the action of the open-loop controls. In the case in which the error is zero and the robot is placed exactly in the desired configuration, only the open loop controls will operate.

Another possible solution is to not use the open-loop controls and empower a law of feedback with the entire task of tracking control. In this case the algorithm must be such as to provide the necessary controls to carry out the motion task even in the case in which the error is nil; therefore there is a unique algorithm for feedback in respect to the current position and the desired one, which serves as both a planner and a controller.

3.5.1. Controllability

It has been seen that the imposition of no slippage non-holonomic constraints takes you directly to the construction of a model of the system that is not linear, the type

$$\dot{q} = G(q)v = g_1 v_1 + g_2 v_2 + ... + g_m v_m \tag{3-23}$$

that represents the kinematic model of the system and can be used for the control study. q is the vector of n configuration variables, v the vector of m input speed, with $m < n$. The columns $g_i\,(i = 1,..m)$ of the matrix G they are continuous vector functions of configuration variables. The system (3-23) is called *drift* as $v = 0 \Rightarrow \dot{q} = 0$.

If a control vector function $v(t)$ continuous or continuous in intervals and an initial condition are chosen $q(0) = q_0$, there is a single system solution for $t \geq 0$, $q(t, 0, q_0, v)$.

The system is *controllable* if $\forall q_1, q_2$ fixed, $\exists T < \infty, \exists v(t)$ defined in the pause $[0, T]$ such that $q(T, 0, q_1, v) = q_2$.

The characteristics of the non-holonomic systems make the level of complexity of the task inverse with respect to the robot manipulators. In fact, for the manipulators the stabilization on a fixed configuration is simpler than in pursuit of the trajectory; this also applies to any mechanical system in which the number of degrees of freedom is equal to that of the controls. For a non-holonomic system such as a robot on wheels, in which it is always $m < n$ (Underactuated), the complexity of the task is reversed. From the qualitative point of view this is evident if one considers that the control task in the point-to-point motion is a stabilization problem around a fixed configuration and corresponds to control n dof with $m = 2$ inputs. For the control task in the pursuit of path the problem is to control a single output (the distance from the route) with $m = 1$

inputs (such as the steering angle, while keeping the other speed inputs constant) and the other $n-1$ d.f are controlled indirectly. Similarly to the pursuit of path, even in pursuit of the trajectory the number of inputs, $m=2$, is equal to the number of controlled outputs (the coordinate errors e_x and e_y), and the others $n-2$ d.f are controlled indirectly. Therefore, the more complex problem is that of the stabilization of a fixed configuration, in which the number of inputs is less than the number of controlled outputs, while for both the path tracking and for the trajectory tracking the number of inputs is equal to the number of outputs, i.e. they are "square" problems.

If [17] an analysis of controllability is carried out it is possible to verify whether the problems mentioned above allow an approximate solution with linear control techniques.

In linear systems controllability involves the *stabilizability* around an equilibrium configuration q_e with continuous linear feedback on the status, of the type

$$v(q) = k(q - q_e)$$

If, in addition, the system linearly approximated around q_e

$$\begin{cases} \delta\dot{q} = J\delta q + B\delta v \\ \delta q = q - q_e \\ \delta v = k\delta q \end{cases}$$

is controllable, then the original system can be stabilized locally around q_e (or even around a trajectory $q_d(t)$) with a continuous feedback.

There are several classes of applicable feedback:

- Feedback *continuous* of status, for linear or linearised systems
  $v = k(q - q_d)$, where q_d is the desired configuration, also variable in time
- Feedback *discontinuous*
  $v = f(q)$, where f is a function continuous in intervals
- Feedback *time-variant*
  $v = f(q,t)$, where f is a continuous function of q, but time-dependent.

In the case of non-holonomic systems, if the system is linearised around a fixed configuration, the resulting linear system *It is not controllable*. This stems directly from

Brockett's theorem: A drift-free system with m g_i linearly independent vectors, it is asymptotically stabilizable with a continuous status feedback if, and only if, the number of controls is greater than or equal to the number of states (configuration variables) $m \geq n$.

By contrast, if the system is linearised around a continuous trajectory the result is a controllable linear time-varying system (Since the number of checks is equal to that of the controlled variables) on condition that the trajectory meets certain conditions on the travelling speed (should never be null).

Defining $\tilde{v}(t) = v(t) - v_d(t)$ the variation of the inputs and as $\tilde{q}(t) = q(t) - q_d(t)$ the tracking error (or tracking), the system's approximation (3-23) to a linear system around the reference trajectory is obtained as follows

$$\dot{\tilde{q}} = J(t)\tilde{q} + B(t)\tilde{v} \qquad (3\text{-}24)$$

where the matrix $n \times n$ $J(t)$ and the matrix $n \times m$ B (t) are obtained as

$$J(t) = \sum_{i=1}^{m} v_{di(t)} \left.\frac{\partial g_i}{\partial q}\right|_{q=q_d(t)} \qquad (3\text{-}25)$$
$$B(t) = G(q_d(t))$$

3.5.2. Static linearised feedback

A closed loop control method is described for the trajectory tracking obtained from the linear approximation of the model car in the chain form [5].

Consider the single-chain model (3-6) (2.4) and assume that you have obtained the open cycle law of motion $[u_{d1}(t), u_{d2}(t)]$ to describe the trajectory $[x_{d1}(t), x_{d2}(t), x_{d3}(t), x_{d4}(t)]$. Indicate respectively the tracking errors and changes in controls such as:

$$\tilde{x}_i = x_i - x_{di} \text{ , } i = 1,..,4$$
$$\tilde{v}_j = v_j - v_{dj} \text{ , } j = 1,2$$

The non-linear system that describes the evolution of tracking error is:

$$\dot{\tilde{x}}_1 = \tilde{u}_1$$
$$\dot{\tilde{x}}_2 = \tilde{u}_2$$
$$\dot{\tilde{x}}_3 = x_2 u_1 - x_{d2} u_{d1}$$
$$\dot{\tilde{x}}_4 = x_3 u_1 - x_{d3} u_{d1}$$

By linearising the system around the reference trajectory according to (3-25) the linear system with time dependence is obtained:

$$\dot{\tilde{x}} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & u_{d1}(t) & 0 & 0 \\ 0 & 0 & u_{d1}(t) & 0 \end{bmatrix} \tilde{x} + \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ x_{d2}(t) & 0 \\ x_{d3}(t) & 0 \end{bmatrix} \tilde{u} = J(t)\tilde{x} + B(t)\tilde{u}$$

From an analysis of controllability (omitted here) it is seen that the linearised system is *controllable* and therefore the original system can be locally stabilized around a reference trajectory via a linear feedback.

The matrix $A_{ac}(t)$ called closed ring, so that $\dot{\tilde{x}} = A_{ac}(t)\tilde{x}$, indicates how to define the feedback guideline $\tilde{u}(t)$ so that the eigenvalues of $A_{ac}(t)$ are constant or with negative part. If the law of feedback is defined

$$\tilde{u}_1 = -k_1 \tilde{x}_1$$
$$\tilde{u}_2 = -k_2 \tilde{x}_2 - \frac{k_3}{u_{d1}} \tilde{x}_3 - \frac{k_4}{u_{d1}^2} \tilde{x}_4$$

The closed loop matrix becomes:

$$A_{ac} = \begin{bmatrix} -k_1 & 0 & 0 & 0 \\ 0 & -k_2 & -k_3/u_{d1}(t) & -k_4/u_{d1}^2(t) \\ -k_1 x_{d2}(t) & u_{d1}(t) & 0 & 0 \\ -k_1 x_{d3}(t) & 0 & u_{d1}(t) & 0 \end{bmatrix}$$

and his four eigenvalues are $-k_1$ and the three roots of the characteristic polynomial $k_4 + k_3\lambda + k_2\lambda^2 + \lambda^3$. Therefore, if $k_1 > 0$ and k_2, k_3, k_4 are chosen so that the characteristic polynomial is a polynomial of Hurwitz, the eigenvalues of $A_{cc}(t)$ are constant and with negative true part, which allows to obtain zero convergence of the trajectory tracking error.

The expression of feedback control $\tilde{u}_2$ shows that it is not defined for $u_{d1} = 0$. The problem can be solved if you assign to the eigenvalues, instead of a constant value, a function of u_{d1}. For example you can calculate the coefficients of the characteristic polynomial so that it has, in addition to $-k_1$, three real roots coincident and equal to $-\gamma\left|u_{d1}\right|$, with $\gamma > 0$. In this way one obtains:

$$\tilde{u}_2 = -3\gamma\left|u_{d1}\right|\tilde{x}_2 - 3\gamma^2 u_{d1}\tilde{x}_3 - \gamma^3\left|u_{d1}\right|\tilde{x}_4 \qquad (3\text{-}26)$$

With this method (*input scaling*), the second feedback control, rather than endless, tends to zero when the variable x_{d1} of the desired trajectory tends to stop ($u_{d1} = 0$).

The overall control of the system in the chain form will be given by

$$u = u_d + \tilde{u}$$

that is, it consists of the sum of a time period calculated for the open loop, feed-forward, and of a closed loop term, of feedback. In order to calculate the actual controls $v(t)$ of the car model the function of processing should be used (3-5) and therefore the driving speed and the steering speed will be nonlinear feedback and time-varying laws.

3.5.3. Application of linearised feedback to the rectilinear trajectory control

The application of the control algorithm with static linearised feedback in pursuit of straight line with speeds is reported $v_{d1}(t)$.

These

- v_1 speed inputs of the rear wheels
- v_2 the input of angular steering speed
- x_d and y_d the desired Cartesian coordinates, in relative coordinates with respect to the trajectory ($y_d \equiv 0$)
- α the steering angle
- δ the vehicle balance in relative coordinates with respect to the straight line to follow

The reference trajectory in relative Cartesian coordinates is

$$x_d(t) = \left|v_d(t)\right| t$$
$$y_d(t) = 0$$

With the desired inputs

$$v_{d1}(t) = |v_{d1}(t)|$$
$$v_{d2}(t) = 0$$

The trajectory in transformed coordinates for a chain system according to (3-4) and (3-5), bearing in mind that $\alpha_d(t) = 0, \delta_d(t) = 0$, is

$$x_{d1}(t) = |v_d(t)|t$$
$$x_{d2}(t) = 0$$
$$x_{d3}(t) = 0$$
$$x_{d4}(t) = 0$$

While the transformed desired inputs are

$$u_{d1}(t) = \dot{x}_{d1} = |v_{d1}(t)|$$
$$u_{d2}(t) = 0$$

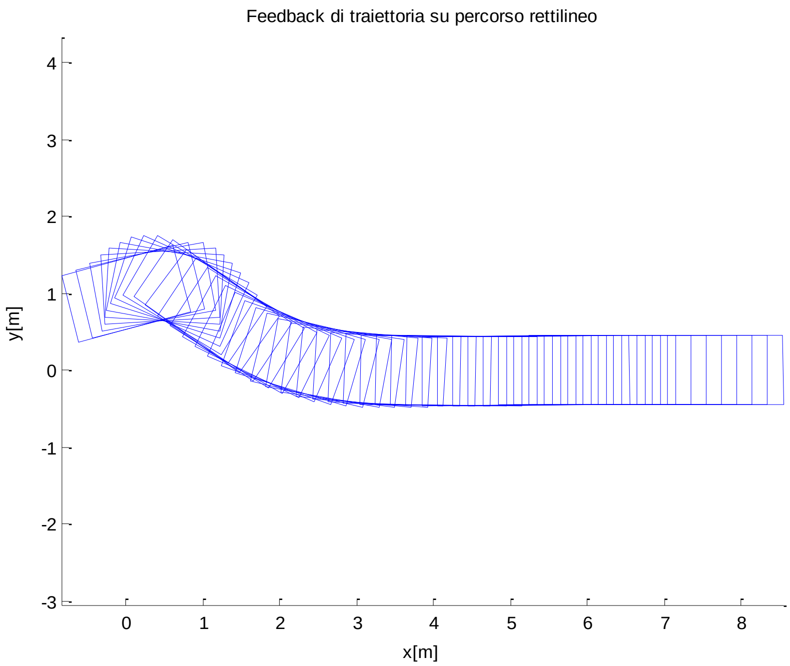
As seen in the previous paragraph the control law, using the (3-4) to express the $\tilde{x}_2, \tilde{x}_3, \tilde{x}_4$ in function of x, y, is:

$$u_1(t) = u_{d1}(t) + \tilde{u}_1 = |v_d(t)| - k_1(x - x_d)$$
$$u_2(t) = \frac{3\gamma |v_d(t)| \tan\alpha}{b\cos^3\delta} + 3\gamma^2 |v_d(t)| \tan\delta + \gamma^3 |v_d(t)| y$$

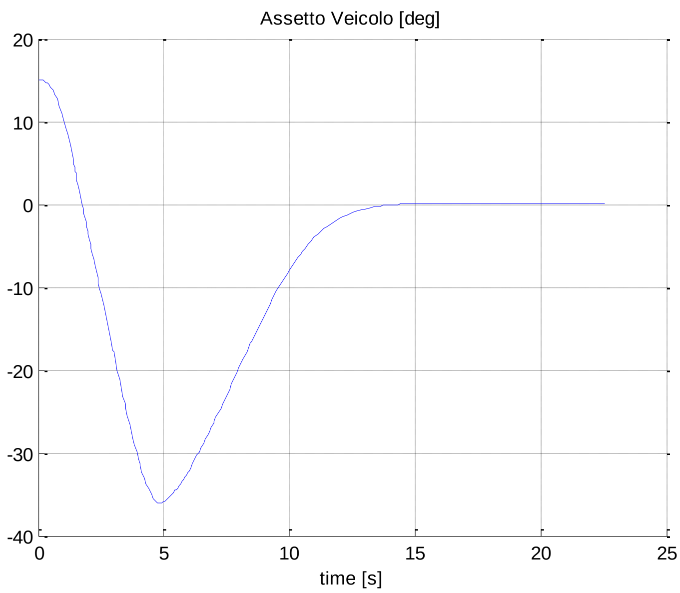
The total actual inputs, according to (3-5) are therefore:

$$v_1(t) = \frac{u_1(t)}{\cos\delta} = \frac{|v_d(t)| - k_1(x - x_d)}{\cos\delta}$$
$$v_2(t) = \frac{-3\sin\delta \sin^2(\alpha) u_1(t)}{b\cos^2\delta} + b\cos^3\delta\cos^2\alpha\, u_2(t)$$

By changing the parameter γ the convergence to zero tracking error is more or less fast, and the algorithm's stability depends on the distance of the actual initial conditions from the nominal conditions and by the value of γ.



Picture 28: Simulation of the trajectory control for alignment on a straight line. Example with initial conditions: $y_0 = 1m$; $\delta_0 = 15°$; speed = 0.4 m / s



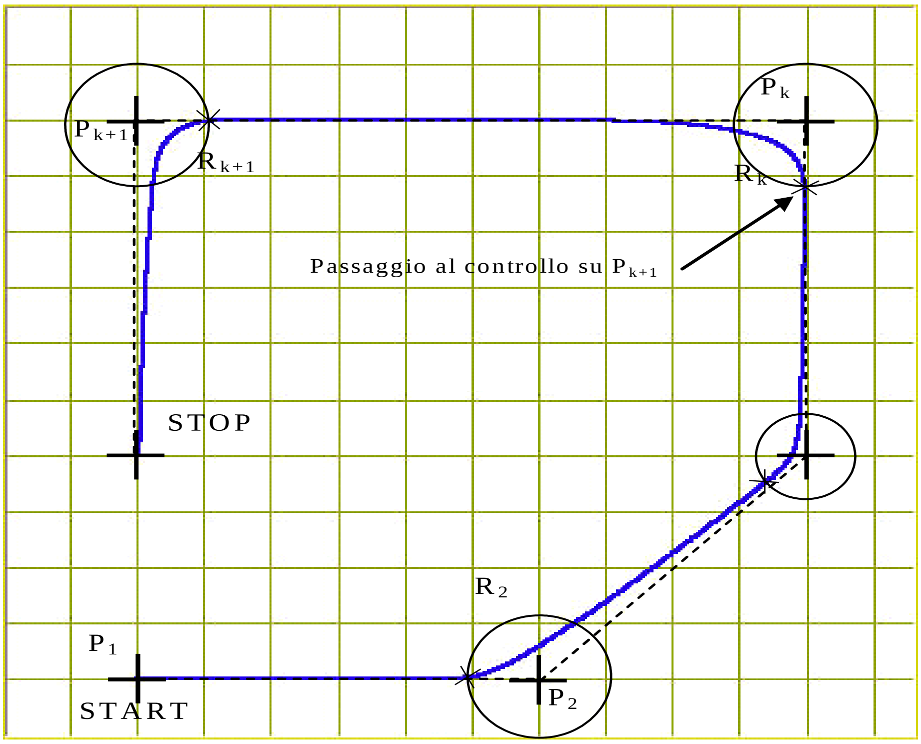
Picture 29: balance pace of the vehicle during the simulation of Picture 30

3.5.4. Heuristic methods for tracking the path

For the tracking of path or trajectory it is possible to resort to heuristic control algorithms. In this case, the control performance is not excellent in a theoretical and mathematical sense, and are evaluated by simulation or experimental trials. The algorithm is validated through a series of tests that allow to verify the behaviour of the system in terms of convergence and stability in a given set of nominal conditions.

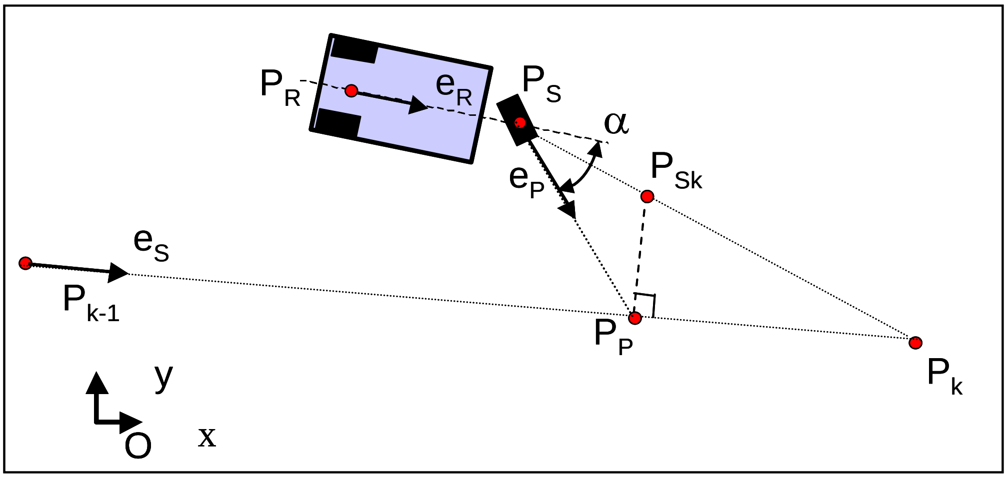
By way of example, a simple type of control for the path tracking is to schedule (and store in a data matrix) the path to be executed by the vehicle through a series of points of passage. It is evident that the broken line that joins the points of passage is not an admissible path for non-holonomic robots the speed of which is always not null. The actual path resulting from the application will therefore be an admissible path that tends toward the broken line of reference.

With reference to Picture 30 each point P_i (x_{the}, y_{the}, R_{the}) is associated with a set of components whose values are: location *x* and *y* of the point with respect to a fixed reference system in the environment and *R* the distance from the point below which the reference passes to the next point. The references for the control are: the point already reached, the next point to be reached and the straight line joining them, so that the trajectory described is similar to the broken line that connects all of the set points in a sequential manner. If the vehicle is monitored to track the P segment$_{k-1}$-P_k and the distance from the point falls below the corresponding value of R_k control may pass to the next segment P_k-P_{k+1}. In particular, thinking of using a tricycle type vehicle, the control acts so as to align the vehicle steering according to the direction of the straight line which is joining the steering axis with a suitable point located on the joining line between point P_k and point P_{k+1}.



Picture 30: Navigation Simulation with tracking

Picture 31 schematises the geometric parameters that regulate the trajectory control. P_k is the point to reach imposed by the planner, P_{k-1} is the point before which, connected with the current point, defines the straight line of reference .



Picture 31: Heuristic algorithm description. The steering setpoint is chosen so as to point to the steering towards a point belonging to the current target segment and dependent on the projection of the steering position on that segment.

The steering angle is controlled by aligning the steering wheel in the direction of the straight line joining the following two points: the point at which there is the steering axis, P_S, and a point P_P identified by projecting on the joining line$P_{k-1}P_k$ point P_{sk}. The latter is chosen on the joining line P_SP_k and its position can vary from P_k to P_S according to factor K that deducts the distance between P_S and P_k:

$$\left|P_{sk}P_k\right| = K\left|P_sP_k\right|$$

and$_R$, δ_R	unit vector of the direction of balance of the vehicle, the unit vector forms the angle with respect to the absolute reference system
and$_P$, δ_P	unit vector in the direction of segment P_SP_P, the unit vector forms the angle with respect to the absolute reference system
and$_S$	unit vector in the direction of segment P_{k-1} - P_k

Table 1: Geometric control parameters

The control law on point P_k and the segment P_{k-1} - P_k determines the reference for the steering, calculated as follows:

$$\alpha = \delta_P - \delta_R$$

In Picture 30 a simulated trajectory obtainable with this type of control is represented. The traction speed is the second control and its variation law can be scheduled independently.

4. WORKS CITED

[1] L. Baglivo, M. De Cecco, "Navigazione di veicoli autonomi. Sensor Fusion ", teaching handout.

[2] J Boisonnat, A Cérézo, J Leblond, "Shortest Paths of Bounded Curvature in the Plane", Proc ICRA, pp. 2315-2320, Nice, France 1992

[3] R. W. Brockett, "Asymptotic stability and feedback stabilization", in Differential Geometric Control Theory, R. W. Brockett, R. S. Millman, H. J. Sussmann (Eds.), Birkhäuser, Boston, MA, pp. 181-191, 1983.

[4] H Delingette, M. Herbert, and K Ikeuchi, "Trajectory Generation with Curvature Constraint based on Energy Minimization", Proc, IROS, pp 206-211, Osaka, Japan, 1991.
[5] A. De Luca, G. Oriolo, C. Samson, “Feedback control of a nonholonomic car.like robot”, in Robot Motion Planning and Control (J. P. Laumond Ed.), Springer-Verlag, 1998.
[6] L. E. Dubins, On Curves of Minimal Length with a Constraint on Average Curvature and with Prescribed Initial and Terminal Positions and Tangents. American Journal of Mathematics, 79:497-516,1957.
[7] M. Fliess, J. Lévine, P. Martin, and P. Rouchon, “Design of trajectory stabilizing feedback for driftless flat systems", 3rd European Control Conf., Rome, I,pp. 1882-1887, 1995.
[8] T. Fraichard and A. Scheuer, “From Reeds and Shepp’s to Continuous-Curvature Paths”, IEEE Trans. on Robotics, Vol. 20, No. 6, pp.1025-1035, 2004
[9] Y. Kanayama, N.Miyake, "Trajectory Generation for Mobile Robots", Robotics Research, MIT Press, Cambridge,1985.
[10] Y. Kanayama and B.I. Hartman, "Smooth Local Path Planning for Autonomous Vehicles" Technical Report, Dept. of Computer Science, University of California, Santa Barbara, 1988.
[11] H. Kano, M. Egerstedt, H. Nakata, and C.F. Martin. B-Splines and Control Theory. To appear in Applied Mathematics and Computation, 2003.
[12] Kelly and B. Nagy, “Reactive Nonholonomic Trajectory Generation via Parametric Optimal Control”, find pages and magazine!!!
[13] G. Laferriere and H. J. Sussmann, “Motion planning for controllable systems without drift", 1991 IEEE Int. Conf. on Robotics and Automation, Sacramento, CA, pp. 1148-1153, 1991.
[14] G. Lafferriere and H.J. Sussmann, “A differential geometric approach to motion planning", Nonholonomic Motion Planning, Zexiang Li and J.F. Canny Eds, The Kluwer International Series in Engineering and Computer Science 192, 1992.
[15] J. C. Latombe, Robot Motion Planning, Kluwer Academic Publishers, 1991.
[16] J. P. Laumond (editor) Robot Motion Planning and Control, LAAS report 97438
[17] J.-P. Laumond, “Controllability of a multibody mobile robot", IEEE Trans. on Robotics and Automation, vol. 9, no. 6, pp. 755-763, 1993.
[18] R. M. Murray and S. S. Sastry, “Nonholonomic motion planning: Steering using sinusoids", IEEE Trans. on Automatic Control, vol. 38, no. 5, pp. 700-716, 1993.
[19] R. M. Murray, “Control of nonholonomic systems using chained forms", Fields Institute Communications, vol. 1, pp. 219-245, 1993. 3rd European Control Conf., Rome, I, pp. 2620-2625, 1995.
[20] B. Nagy and A. Kelly, “Trajectory Generation for Car-Like Robots Using Cubic Curvature Polynomials”, in Field and Service Robots, June 11, 2001, Helsinki, Finland.
[21] J. A. Reeds and R. A. Shepp, Optimal paths for a car that goes both forwards and backwards”, Pacific Journal of Mathematics, 145 (2), 1990.
[22] Johannes Reuter, “Mobile Robot Trajectories With Continuously Differentiable Curvature: An Optimal Control Approach”, Proceedings of the 1998 IEEE/RSJ Conference on Intelligent Robots and Systems, Victoria, B.C, Canada, October 1998.
[23] C. Samson and K. Ait-Abderrahim, “Feedback stabilization of a nonholonomic wheeled mobile robot”, 1991 IEEE/RSJ Int. Work. on Intelligent Robots and Systems, Osaka, J, pp. 1242-1247, 1991.
[24] D. H. Shin and S. Singh, "Path Generation for Robot Vehicles Using Composite Clothoid Segments", Technical Report, CMU-RITR- 90-31, The Robotics Institute, Carnegie Mellon University, 1990.
[25] O. J. Sordalen, “Conversion of the kinematics of a car with n trailers into a chained form", 1993 IEEE Int. Conf. on Robotics and Automation, Atlanta, GA, vol. 1, pp. 382-387, 1993.
[26] O. J. Sordalen, “Feedback control of nonholonomic mobile robots", Ph. D. Thesis, The Norwegian Institute of Technology, Trondheim, NO, Mar. 1993.
[27] D. Tilbury, J.P. Laumond, R. Murray, S. Sastry and G. Walsh, “Steering car-like systems with trailers using sinusoids" in IEEE Conf. on Robotics and Automation, pp. 1993-1998, Nice, 1992.