# Model Predictive Control of Vehicle Formations*

Fernando A.C.C. Fontes,[1] Dalila B.M.M. Fontes[2], and Amélia C.D. Caldeira[3]

[1] Depart. de Matemática para a Ciência e Tecnologia and ISR-Porto,
Universidade do Minho,
4800-058 Guimarães, Portugal
ffontes@mct.uminho.pt

[2] Faculdade de Economia and LIAAD-INESC Porto L.A.,
Universidade do Porto
Rua Dr. Roberto Frias, 4200-464 Porto, Portugal
fontes@fep.up.pt

[3] Departamento de Matemática,
Instituto Superior de Engenharia do Porto,
R. Dr. Ant. Bernardino de Almeida, 431, 4200-072 Porto, Portugal
acd@isep.ipp.pt

**Abstract.** We propose a two-layer scheme to control a set of vehicles moving in a formation.

The first layer, the trajectory controller, is a nonlinear controller since most vehicles are nonholonomic systems and require a nonlinear, even discontinuous, feedback to stabilize them. The trajectory controller, a model predictive controller, computes centrally a bang-bang control law and only a small set of parameters need to be transmitted to each vehicle at each iteration.

The second layer, the formation controller, aims to compensate for small changes around a nominal trajectory maintaining the relative positions between vehicles. We argue that the formation control can be, in most cases, adequately carried out by a linear model predictive controller accommodating input and state constraints. This has the advantage that the control laws for each vehicle are simple piecewise affine feedback laws that can be pre-computed off-line and implemented in a distributed way in each vehicle.

Although several optimization problems have to be solved, the control strategy proposed results in a simple and efficient implementation where no optimization problem needs to be solved in real-time at each vehicle.

## 1 Introduction

In this chapter we propose a control scheme for a set of vehicles moving in a formation. Vehicle formations are used in several applications, both in military and civilian operations, such as surveillance, forest fire detection, search missions, automated highways, exploration robots, among many others (see e.g., [20]).

* Research supported by FCT/POCI 2010/FEDER through Project POCTI/MAT/61842/2004.

M.J. Hirsch et al. (Eds.): Optimization & Cooperative Ctrl. Strategies, LNCIS 381, pp. 371–384.
springerlink.com

The control methodology selected is a two-layer control scheme where each layer is based on model predictive control (MPC). Control of multi-vehicle formations and/or distributed MPC schemes have been proposed in the literature. See the recent works [5,22] and references therein.

The reason why two-layers are used in the control scheme is because there are two intrinsically different control problems:

- the trajectory control problem: to devise a trajectory, and corresponding actuator signals, for the formation as a whole.
- maintain the formation: change the actuator signals in each vehicle to compensate for small changes around a nominal trajectory and maintain the relative position between vehicles.

These control problems are intrinsically different because, on the one hand, most vehicles (cars, planes, submarines, wheeled vehicles) are nonholonomic (cannot move in all directions). On the other hand, while the vehicles are in motion, the relative position between them in a formation can be changed in all directions (as if they were holonomic).

As an example consider a vehicle whose dynamics we are familiar with: a car. Consider the car performing a parking maneuver or performing an overtaking maneuver. See figures 1 and 2.

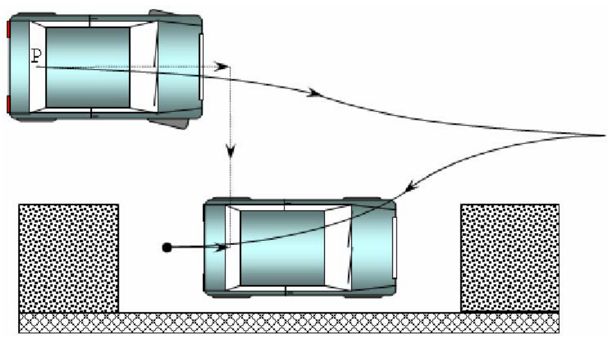

**Fig. 1.** Car in a parking maneuver: cannot move sideways

In the first situation, we are limited by the fact that the car cannot move sideways: it is nonholonomic. It is a known result that we need a nonlinear controller, allowing discontinuous feedback laws, to stabilize this system in this situation [23,3].

In the second, the vehicle is in motion and small changes around the nominal trajectory can be carried out in all directions of the space. In an overtaking maneuver we can move in all directions relative to the other vehicle. This fact simplifies considerably the controller design. In fact, a linear controller is an appropriate controller to deal with small changes in every spatial direction around a determined operating point.

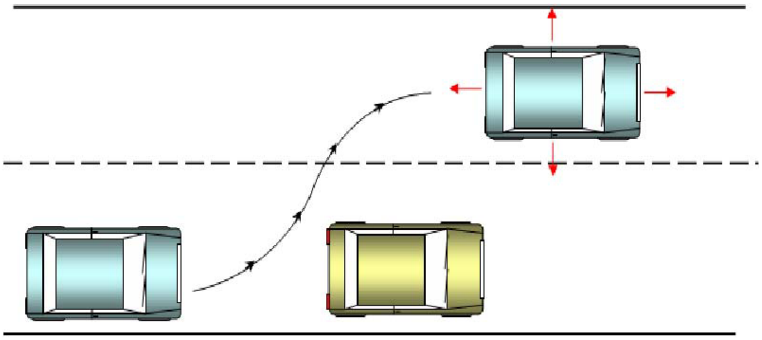

**Fig. 2.** Car in an overtaking maneuver: can move in all directions relative to the other car

For a different approach that also decouples the path following from the inter-vehicle coordination problem see [11].

The other option here is to design and use controllers based on the Model Predictive Control technique. Several reasons support this option.

MPC is known to be a technique that deals appropriately and explicitly with constraints. In fact, many researchers argue that the capacity of dealing naturally and effectively with constraints is the main reason for the industrial success of MPC; see e.g., [18,15,21].

In the problem of controlling a vehicle formation, the constraints on the trajectory are an important problem characteristic:

- to avoid collisions between the vehicles in the formation;
- to avoid obstacles in the path; and,
- for other reasons of safety or reliability of operation (e.g., minimum altitude or velocity in a plane).

Therefore, constraints should be appropriately dealt with. A possible way to deal with constraints is to use unconstrained design methods combined with a “cautious” approach of operating far from the constraints (e.g., imposing large distances between the vehicles in the formations). But, such procedures would, in general, reduce the performance that would be achievable by operating closer to the limits [12].

Another possible approach is to push the trajectory away from the constraints by penalizing a vehicle close to an obstacle or other vehicle (e.g., potential field approaches, or other optimization-based approaches that do not impose explicitly the constraints), but these methods do not guarantee that the constraints will be satisfied.

Here, we use MPC imposing explicit constraints both on the vehicle inputs and on the vehicle trajectory.

Another advantage of MPC is that, because MPC is an optimization-based method, it has desirable performance properties according to the performance criteria defined. In addition, it has intrinsic robustness properties [17].

The major concern in using such a technique is often the need to solve optimization problems in real-time which might be difficult for fast systems. However, there are recent results on particular parameterizations for the optimal control problems involved that allow MPC to address fast systems (see e.g., [1]). We shall see that the control strategy proposed appropriately deals with this problem.

## 2 The MPC Framework

Consider a nonlinear plant with input and state constraints, where the evolution of the state after time $t_0$ is predicted by the following model.

$$\begin{aligned} &\dot{x}(s) = f(s, x(s), u(s)) && \text{a.e. } s \geq t_0, && \text{(1a)}\\ &x(t_0) = x_{t_0} \in X_0, && && \text{(1b)}\\ &x(s) \in X \subset \mathbb{R}^n && \text{for all } s \geq t_0, && \text{(1c)}\\ &u(s) \in U && \text{a.e. } s \geq t_0. && \text{(1d)} \end{aligned}$$

The data of this model comprise a set $X_0 \subset \mathbb{R}^n$ containing all possible initial states at the initial time $t_0$, a vector $x_{t_0}$ that is the state of the plant measured at time $t_0$, a given function $f : \mathbb{R} \times \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$, and a set $U \subset \mathbb{R}^m$ of possible control values.

We assume this system to be asymptotically controllable on $X_0$ and that for all $t \geq 0$ $f(t, 0, 0) = 0$. We further assume that the function $f$ is continuous and locally Lipschitz with respect to the second argument.

The construction of the feedback law can be accomplished by using a sampled-data MPC strategy [10]. Consider a sequence of sampling instants $\pi := \{t_i\}_{i \geq 0}$, with a constant inter-sampling time $\delta > 0$ such that $t_{i+1} = t_i + \delta$ for all $i \geq 0$. Consider also the control horizon and predictive horizon, $T_c$ and $T_p$, with $T_p \geq T_c > \delta$, and an auxiliary control law $k^{aux} : \mathbb{R} \times \mathbb{R}^n \to \mathbb{R}^m$. The feedback control is obtained by repeatedly solving online open-loop optimal control problems $\mathcal{P}(t_i, x_{t_i}, T_c, T_p)$ at each sampling instant $t_i \in \pi$, every time using the current measure of the state of the plant $x_{t_i}$.

$\mathcal{P}(t, x_t, T_c, T_p)$: Minimize

$$\int_t^{t+T_p} L(s, x(s), u(s))ds + W(t + T_p, x(t + T_p)), \tag{2}$$

subject to:

$$\begin{aligned} &\dot{x}(s) = f(s, x(s), u(s)) && \text{a.e. } s \in [t, t + T_p], && \text{(3)}\\ &x(t) = x_t, && &&\\ &x(s) \in X && \text{for all } s \in [t, t + T_p], &&\\ &u(s) \in U && \text{a.e. } s \in [t, t + T_c], &&\\ &u(s) = k^{aux}(s, x(s)) && \text{a.e. } s \in [t + T_c, t + T_p], &&\\ &x(t + T_p) \in S. && && \text{(4)} \end{aligned}$$

The domain of this optimization problem is the set of admissible processes, namely pairs $(x, u)$ comprising a control function $u$ and the corresponding state trajectory $x$ which satisfy the constraints of $\mathcal{P}(t, x_t, T_c, T_p)$. A process $(\bar{x}, \bar{u})$ is said to solve $\mathcal{P}(t, x_t, T_c, T_p)$ if it minimizes (2) among all admissible processes.

Note that in the interval $[t + T_c, t + T_p]$ the control value is determined by $k^{aux}$ and therefore the optimization decisions are all carried out in the interval $[t, t + T_c]$.

We call *design parameters* those variables present in the open-loop optimal control problem that are not from the system model (i.e., variables we are able to choose); these comprise the control horizon $T_c$, the prediction horizon $T_p$, the running cost function $L$, the terminal cost function $W$, the auxiliary control law $k^{aux}$, and the terminal constraint set $S \subset \mathbb{R}^n$. The choice of these variables is important to obtain certain properties for the MPC strategy, such as stability, robustness, or performance (see e.g., [19,7,9] for a discussion on to choose the design parameters).

The MPC algorithm performs according to a receding horizon strategy, as follows.

1. Measure the current state of the plant $x^*(t_i)$.
2. Compute the open-loop optimal control $\bar{u} : [t_i, t_i + T_c] \to \mathbb{R}^n$ solution to problem $\mathcal{P}(t_i, x^*(t_i), T_c, T_p)$.
3. Apply to the plant the control $u^*(t) := \bar{u}(t; t_i, x^*(t_i))$ in the interval $[t_i, t_i+\delta)$ (the remaining control $\bar{u}(t), t \geq t_i + \delta$ is discarded).
4. Repeat the procedure from (1.) for the next sampling instant $t_{i+1}$ (the index $i$ is incremented by one unit).

The resultant control law $u^*$ is a "sampling-feedback" control since during each sampling interval, the control $u^*$ is dependent on the state $x^*(t_i)$. More precisely, the resulting trajectory is given by

$$x^*(t_0) = x_{t_0}, \quad \dot{x}^*(t) = f(t, x^*(t), u^*(t)) \quad t \geq t_0, \tag{5}$$

where

$$u^*(t) := \bar{u}(t; \lfloor t \rfloor_\pi, x^*(\lfloor t \rfloor_\pi)) \quad t \geq t_0, \tag{6}$$

and the function $t \mapsto \lfloor t \rfloor_\pi$ gives the last sampling instant before $t$, that is

$$\lfloor t \rfloor_\pi := \max_i \{t_i \in \pi : t_i \leq t\}. \tag{7}$$

Similar sampled-data frameworks using continuous-time models and sampling the state of the plant at discrete instants of time were adopted in [4,7,8,6,16] and are becoming the accepted framework for continuous-time MPC. It can be shown that with this framework it is possible to address — and guarantee stability, and robustness of the resultant closed-loop system — a very large class of systems, which are possibly nonlinear, time-varying, and nonholonomic.

## 3 The Two-Layer Control Scheme

As discussed, we propose a two-layer control scheme. The top layer, **the trajectory controller**, is applied to the group of vehicles as a whole. The trajectory controller is a nonlinear controller since most vehicles are nonholonomic systems and require a nonlinear, even discontinuous, feedback to stabilize them. The main ideas for the controller used in this section follow closely the results from [9].

The bottom layer, **the formation controller**, is computed and applied to each vehicle individually. The formation controller aims to compensate for small changes around a nominal trajectory maintaining the relative positions between vehicles. We argue that the formation control can be adequately carried out by a linear model predictive controller accommodating input and state constraints. This has the advantage that the control laws for each vehicle are simple piecewise affine feedback laws that can be pre-computed off-line and implemented in a distributed way in each vehicle. The main reference for this controller is [2].

## 4 The Vehicle Formation Models

### 4.1 Nonholonomic Vehicle Model for the Trajectory Controller

The framework developed here could easily be adapted to vehicles moving in 2D or 3D and having various dynamics. Nevertheless, we will explore a simple case of a a differential-drive mobile robot moving on a plane, Figure 3, and represented by the following kinematic model:

$$\dot{x}(t) = (u_1(t) + u_2(t)) \cdot \cos\theta(t) \tag{8}$$

$$\dot{y}(t) = (u_1(t) + u_2(t)) \cdot \sin\theta(t) \tag{9}$$

$$\dot{\theta}(t) = (u_1(t) - u_2(t)), \tag{10}$$

with $\theta(t) \in [-\pi, \pi]$, and the controls $u_1(t), u_2(t) \in [-1, 1]$.

The coordinates $(x, y)$ are the position in the plane of the midpoint of the axle connecting the rear wheels, and $\theta$ denotes the heading angle measured from the $x$-axis. The controls $u_1$ and $u_2$ are the angular velocity of the right and left wheels, respectively. If the same velocity is applied to both wheels, the robot moves along a straight line (maximum forward velocity when $u_1 = u_2 = u_{max} = 1$). The robot can turn by choosing $u_1 \neq u_2$ (when $u_1 = -u_2 = 1$ the robot turns counter-clockwise around the midpoint of the axle).

The velocity vector is always orthogonal to the wheel axis. This is the non-slip or nonholonomic constraint

$$(\dot{x}, \dot{y})^T (\sin\theta, -\cos\theta) = 0. \tag{11}$$

Therefore, the vehicle model is a nonholonomic system. It is completely controllable but instantaneously it cannot move in certain directions. It is known that

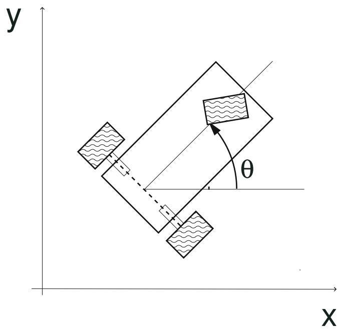

**Fig. 3.** A differential-drive mobile robot

this class of systems requires a nonlinear controller to stabilize it. Furthermore, the controller must allow discontinuous (or alternatively time-varying) feedback control laws. See [13] for a discussion on the control of nonholonomic systems, [3,23] for the need to use discontinuous feedbacks, and [8] for MPC of nonholonomic systems.

### 4.2 Linearized Vehicle Model for the Formation Controller

For the reasons explained above, to control the relative position between the vehicles in the formation, compensating for small deviations around a nominal trajectory, we might use linear control methods.

Consider that one of the vehicles is the reference vehicle. It might be the formation leader, or any of the vehicles in the formation, or we might even want to consider an additional nonexistent vehicle for modeling purposes. Consider that the formation moves at the nominal linear velocity $v_n = (u_{1n} + u_{2n})/2$, with $u_{1n} = u_{2n}$ being the nominal velocities of the wheels. Let $v_l$ be the linear velocity added to the nominal linear velocity, and $v_w$ the angular velocity. Assume that $\theta$ is small, so that $\cos\theta \simeq 1$ and $\sin\theta \simeq \theta$. We thus have the simplified model

$$\dot{x}(t) = (v_n(t) + v_l(t)) \tag{12}$$

$$\dot{y}(t) = (v_n(t) + v_l(t))\theta(t) \tag{13}$$

$$\dot{\theta}(t) = v_w(t). \tag{14}$$

We consider a linearized model for each of the differential drive mobile robots with the $z_1$ axis aligned with the velocity of the *reference vehicle* .

$$\begin{bmatrix} \dot{z}_1 \\ \dot{z}_2 \\ \dot{z}_3 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & v_n \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v_l \\ v_w \end{bmatrix} \tag{15}$$

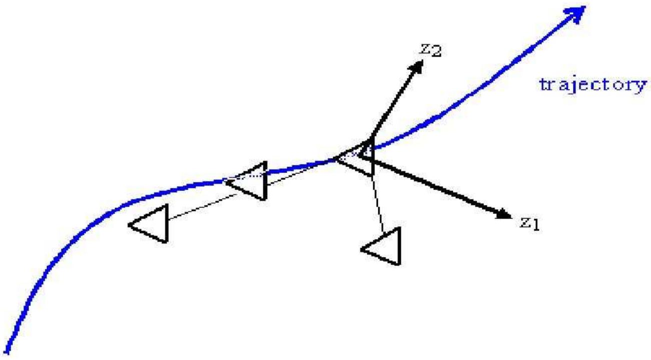

**Fig. 4.** Reference frame for the Linearized vehicle model aligned with the reference trajectory

For computation purposes, it is convenient to use a discrete-time model here. The discrete-time model, converted using zero order hold with sample time $h$, is

$$\begin{bmatrix} z_1(t+h) \\ z_2(t+h) \\ z_3(t+h) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & v_n h \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} z_1(t) \\ z_2(t) \\ z_3(t) \end{bmatrix} + \begin{bmatrix} h & 0 \\ 0 & 0 \\ 0 & h \end{bmatrix} \begin{bmatrix} v_l(t) \\ v_w(t) \end{bmatrix} \tag{16}$$

which we will simply denote as

$$\dot{z}(t+h) = Az(t) + Bv(t). \tag{17}$$

### 4.3 Formation Connections Model

Consider $M$ vehicles and a *reference vehicle* (that might not exist) that follows exactly the trajectory predicted by the trajectory controller.

Consider that the $M+1$ vehicles are nodes of a directed graph $G = (V, E)$, which is a directed tree rooted at the reference vehicle. That is, all vertices are connected, there are no cycles, and all edges are directed in a path away from the root. Associate to each vertex $i$ a triplet $z^i(t) = (z_1^i(t), z_2^i(t), z_3^i(t))$ with the relative position with respect to the reference vehicle at time $t$. Also, associate with each edge $(i,j) \in E$ a pair $\tilde{z}^{ij} = (\tilde{z}_1^{ij}, \tilde{z}_2^{ij})$ with the desired position in the plane of vehicle $j$ with respect to vehicle $i$. The desired relative positions are defined a priori and define the geometry of the formation that we want to achieve. For node $i$ we define its parent node $\mathcal{P}_i = j$ such that $(i,j) \in E$. See Figure 5.

The objective of the formation controller is to maintain the geometry of the formation, that is, the relative positions between the vehicles should be kept as close as possible to the desired relative positions. We define for each vehicle $i$ with parent $j$ a performance index that penalizes the difference between the desired and actual relative positions of the vehicles.

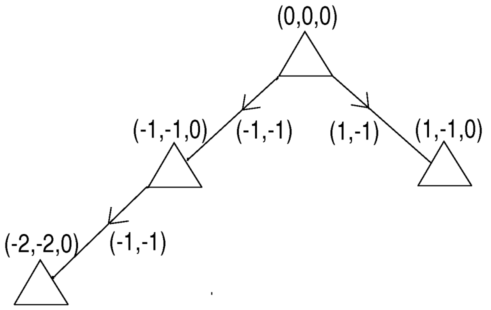

**Fig. 5.** A tree modelling the formation connections

$$
\begin{aligned}
J^i = &\|(\tilde{z}_1^{ij}, \tilde{z}_2^{ij}, 0) - (z^j(N) - z^i(N))\|_P^2 + \\
&+ \sum_{t=1}^{N-1} \|(\tilde{z}_1^{ij}, \tilde{z}_2^{ij}, 0) - (z^j(t) - z^i(t))\|_Q^2 + \|(v(t), w(t))\|_R^2. \qquad (18)
\end{aligned}
$$

# 5 The Controllers

## 5.1 Trajectory Controller

The controller used here follows closely the results in [9]. We use bang-bang feedback controls. That is, for each state, the corresponding control must be at one of the extreme values of its range. The exception is the target set $\Theta$ (here a small ball centered at the origin) where the control is chosen to be zero.

The feedbacks, when outside the target set, are defined by a switching surface $\sigma(x) = 0$. The control will attain its maximum or minimum value depending on which side of the surface the state is. More precisely, for each component $j = 1, \ldots, m$ of the control vector, define

$$
k_j(x) = \begin{cases} 0 & \text{if } x \in \Theta, \\ u_j^{max} & \text{if } \sigma_j(x) \geq 0, \\ u_j^{min} & \text{if } \sigma_j(x) < 0. \end{cases} \qquad (19)
$$

The function $\sigma_j$ is a component of the vector function $\sigma$, and is associated with the switching surface $\sigma_j(x) = 0$ which divides the state-space in two.

These surfaces must be parameterized in some way to be chosen in an optimization problem. Therefore, we define them to have a fixed part $\sigma^{aux}$, possibly nonlinear, and a variable part $\sigma^{\Lambda}$ which is affine and defined by the parameter matrix $\Lambda$.

$$
\sigma(x) = \sigma^{aux}(x) + \sigma^{\Lambda}(x). \qquad (20)
$$

For each component $j = 1, 2, \ldots, m$, the equation $\sigma_j^{\Lambda} = 0$ is the equation of a hyperplane which is defined by $n + 1$ parameters as

$$
\sigma_j^{\Lambda}(x) := \lambda_{j,0} + \lambda_{j,1}\, x_1 + \ldots + \lambda_{j,n}\, x_n. \qquad (21)
$$

The half-spaces $\sigma_j^\Lambda(x) \geq 0$ and $\sigma_j^\Lambda(x) < 0$ are not affected by multiplying all parameters by a positive scalar, therefore we can fix one parameter, say $\lambda_{j,0}$, to be in $\{-1, 0, 1\}$. In total, for all components of the control vector, there will be $m \times (n+1)$ parameters to choose from. By selecting the parameter matrix

$$\Lambda := \begin{bmatrix} \lambda_{1,0} \cdots \lambda_{1,n} \\ \cdots \\ \lambda_{m,0} \cdots \lambda_{m,n} \end{bmatrix}, \tag{22}$$

we define the function

$$\sigma^\Lambda(x) = \Lambda \begin{bmatrix} 1 \\ x \end{bmatrix}, \tag{23}$$

and therefore we define the switching function $\sigma$ by (20) and we define the feedback law $k^\Lambda$ by (19). Each component of the feedback law can be described as

$$k_j^\Lambda(x) = \begin{cases} 0 & \text{if } x \in \Theta, \\ u_j^{max} & \text{if } \left[\sigma^{aux}(x) + \Lambda \begin{bmatrix} 1 \\ x \end{bmatrix}\right]_j \geq 0, \\ u_j^{min} & \text{if } \left[\sigma^{aux}(x) + \Lambda \begin{bmatrix} 1 \\ x \end{bmatrix}\right]_j < 0. \end{cases} \tag{24}$$

In our example the switching surfaces, for each control component, are planes in the state-space $\mathbb{R}^3$ and can be described by 4 real parameters. The surface, and therefore the feedback, are defined by a parameter matrix $\Lambda \in \mathbb{R}^{2\times 4}$.

The feedbacks are obtained by solving, in a receding horizon strategy, the following optimal control problems with respect to matrices $\Lambda \in \mathbb{R}^{2\times 4}$.

Minimize$_{\Lambda_1,\ldots,\Lambda_{N_c} \in \mathbb{R}^{2\times 4}}$

$$\int_t^{t+T_p} L(x(s), u(s))ds + W(x(t+T_p)) \tag{25}$$

subject to (26)

$$\begin{aligned} & x(t) = x_t, \\ & \dot{x}(s) = f(x(s), u(s)) && \text{a.e. } s \in [t, t+T_p], && (27) \\ & x(s) \in X && \text{for all } s \in [t, t+T_p], \\ & x(t+T_p) \in S, && && (28) \end{aligned}$$

where

$$\begin{aligned} u(s) &= k^{\Lambda_i}(x(\lfloor s \rfloor_\pi)) && s \in [t+(i-1)\delta, t+i\delta), \quad i = 1, \ldots, N_c, && (29) \\ u(s) &= k^{aux}(x(\lfloor s \rfloor_\pi)) && s \in [t+(i-1)\delta, t+i\delta), \quad i = N_c+1, \ldots, N_p. && (30) \end{aligned}$$

In this optimal control problem, the control horizon $T_c$ and prediction horizon $T_p$ satisfy $T_c = N_c\delta$ and $T_p = N_p\delta$ with $N_c, N_P \in \mathbb{N}$ and $N_c \leq N_p$.

The guarantee of stability of the resulting closed loop system can be given by a choice of design parameters satisfying a sufficient stability condition. The following set of design parameters [9] guarantees stability:

$$L(x,y,\theta) = x^2 + y^2 + \theta^2, \tag{31}$$

$$W(x,y,\theta) = \frac{1}{3}(r^3 + |\theta|^3) + r\theta^2 \quad \text{with } r = \sqrt{x^2+y^2}, \tag{32}$$

$$T_p = \frac{2\pi}{3}, \tag{33}$$

$$\begin{aligned} S := \{(x,y,\theta) \in \mathbb{R}^2 \times [-\pi,\pi] : \phi_m(x,y) \le \theta \le \phi_M(x,y) \vee \\ \vee\ (x,y,\theta) \in \Theta\ \vee\ (x,y) = (0,0)\}. \end{aligned} \tag{34}$$

### 5.2 Formation Controller

For each vehicle $i$, with parent $j$, compute the control solving the constrained linear quadratic optimal control problem:

Minimize over sequences $\{v_1, \ldots, v_{N-1}\}$

$$\begin{aligned} J =& \|(\tilde{z}_1^{ij}, \tilde{z}_2^{ij}, 0) - (z^j(N) - z(N))\|_P^2 + \\ &+ \sum_{t=1}^{N-1} \|(\tilde{z}_1^{ij}, \tilde{z}_2^{ij}, 0) - (z^j(t) - z(t))\|_Q^2 + \|v(t)\|_R^2 \end{aligned} \tag{35}$$

subject to

$$\begin{aligned}
&z(t+h) = Az(t) + Bv(t) && t = 0, \ldots, N-1, \\
&z(0) = z_0, \\
&v_{min} \le v(t) \le v_{max} && t = 0, \ldots, N-1, && (36)\\
&(\underline{z}_1^{ij}, \underline{z}_2^{ij}, 0) \le z(t) - z^j(t) \le (\bar{z}_1^{ij}, \bar{z}_2^{ij}, 0) && t = 1, \ldots, N, && (37)\\
&Dz(t) \le d && t = 1, \ldots, N. && (38)
\end{aligned}$$

Here, constraints (36) are limits on the inputs. Constraints (37) set a maximum and minimum distance to the parent vehicle, and inequalities (38) are general constraints to accommodate, for example, forbidden zones of the state-space.

The matrices involved in the performance index are chosen to satisfy $Q = Q' \succeq 0, R = R' \succ 0$, and $P$ solving the Lyapunov equation $P = A'PA + Q$. This strategy has, for each vehicle and in the conditions stated, guaranteed exponential stability [2,19].

One of the major advantages of this controller is the fact that the feedback laws obtained from solving the linear quadratic regulator with constraints are

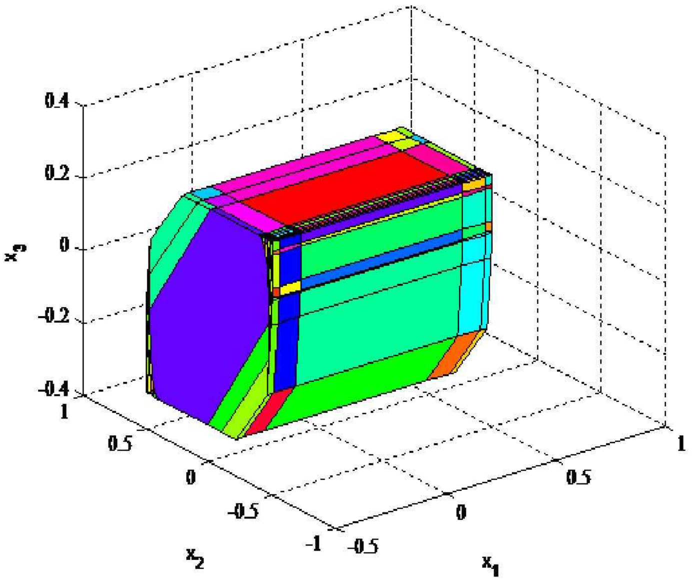

**Fig. 6.** State-space regions for the reference vehicle PWA control law

continuous piecewise affine (PWA) functions [2], i.e. , for a certain region of the state-space $\mathcal{R}_k$ (which is a polytope) the control law is affine

$$u(t) = F_k z(t) + G_k \qquad \text{if } z(t) \in \mathcal{R}_k. \tag{39}$$

In Figure 6 we can see the different state-space regions (195 polytopes) corresponding to different affine control laws for the reference vehicle.

This way, the parameters of the PWA feedback (matrices $F_k$ and $G_k$ for each region $\mathcal{R}_k$) can be determined explicitly a priori, off-line, by multi-parametric programming (e.g., using MPT - Multi Parametric Matlab Toolbox [14]). Each vehicle just has to store the parameters of the PWA feedback function. No optimization nor other complex computations are involved in real-time. Only lookup table operations are needed for each vehicle.

The stability of the whole formation is easy to establish. This is because, with the strategy above, the trajectory of each vehicle is exponentially stable with respect the desired relative position to its parent vehicle. As there is exactly one path from each vehicle to the reference vehicle, stability of any vehicle easily follows recursively.

If more general graphs are allowed, comprising not only trees but also admitting loops, the stability analysis is considerably more complex. For results on stability considering more general graphs see [5].

### 5.3 Integration of the Two Control Layers

In each vehicle, the control given by the Trajectory Controller *plus* the control given by the Formation Controller are applied. Therefore, we have to consider control limits in each layer in such a way that the physical limits are respected. The Trajectory Controller (bang-bang feedback) can only use part of the limit (70% in our simulations) leaving the rest to the Formation Controller.

For the Trajectory Controller, the bang-bang feedback is computed centrally and then the switching surface parameters (a 2×4 matrix per sampling period) are communicated to each vehicle. The frequency of update might be relatively low, and is dictated by the information from the outside (the formation) world of new obstacles in the trajectory, possibly identified by the sensors, that might alter the main path.

For the Formation Controller, the PWA feedback law is computed a priori, off-line, and implemented in each vehicle.

## 6 Conclusions

Model Predictive Control was shown to be an adequate tool for control of vehicles in a formation: it deals explicitly and effectively with the constraints that are an important problem feature; recent results on parameterized approaches to the optimal control problems allow addressing fast systems as well as allowing efficient implementations.

Although several optimization problems have to be solved, none need to be solved in real time by each of the vehicles. Part of the problem can be solved a priori, off-line, and an explicit control law can be implemented in each vehicle as a lookup table; the other part can be solved centrally and only a few parameters (a 2×4 matrix) need to be transmitted to all vehicles in each sampling period.

## References

1. Alamir, M.: Stabilization of Nonlinear Systems Using Receding-Horizon Control Schemes: A Parameterized Approach for Fast System. Lecture Notes in Control and Information Sciences. Springer, London (2006)
2. Bemporad, A., Morari, M., Dua, V., Pistikopoulos, E.N.: The explicit linear quadratic regulator for constrained systems. Automatica 38, 3–20 (2002)
3. Brockett, R.W.: Asymptotic stability and feedback stabilization. In: Brockett, R.W., Millman, R.S., Sussmann, H.S. (eds.) Differential Geometric Control Theory, pp. 181–191. Birkhouser, Boston (1983)
4. Chen, H., Allgöwer, F.: Nonlinear model predictive control schemes with guaranteed stability. In: Berber, R., Kravaris, C. (eds.) Nonlinear Model Based Process Control. Kluwer, Dordrecht (1998)
5. Dunbar, W.B., Murray, R.M.: Distributed receding horizon control for multi-vehicle formation stabilization. Automatica 42, 549–558 (2006)
6. Findeisen, R., Imsland, L., Allgöwer, F., Foss, B.: State and output feedback nonlinear model predictive control: an overview. European Journal of Control 9, 190–206 (2003)
7. Fontes, F.A.C.C.: A general framework to design stabilizing nonlinear model predictive controllers. Systems & Control Letters 42, 127–143 (2001)

8. Fontes, F.A.C.C.: Discontinuous feedbacks, discontinuous optimal controls, and continuous-time model predictive control. International Journal of Robust and Nonlinear Control 13(3–4), 191–209 (2003)
9. Fontes, F.A.C.C., Magni, L.: Min-max model predictive control of nonlinear systems using discontinuous feedbacks. IEEE Transactions on Automatic Control 48, 1750–1755 (2003)
10. Fontes, F.A.C.C., Magni, L., Gyurkovics, E.: Sampled-data model predictive control for nonlinear time-varying systems: Stability and robustness. In: Allgower, F., Findeisen, R., Biegler, L. (eds.) Assessment and Future Directions of Nonlinear Model Predictive Control. Lecture Notes in Control and Information Systems, vol. 358, pp. 115–129. Springer, Heidelberg (2007)
11. Ghabcheloo, R., Pascoal, A., Silvestre, C., Kaminer, I.: Non-linear co-ordinated path following control of multiple wheeled robots with bidirectional communication constraints. International Journal of Adaptive Control and Signal Processing 21, 133–157 (2007)
12. Goodwin, G.C., Seron, M.M., De Doná, J.A.: Constrained Control and Estimation: An Optimisation Approach. Springer, London (2004)
13. Kolmanovsky, I., McClamroch, N.H.: Developments in nonholonomic control problems. IEEE Control Systems, 20–36 (December 1995)
14. Kvasnica, M., Grieder, P., Baotic, M., Christophersen, F.J.: MPT - Multiparametric Toolbox. Technical report, ETH - Swiss Federal Institute of Technology, Zurich (2006)
15. Maciejowski, J.M.: Predictive Control with Constraints. Prentice Hall, Harlow (2002)
16. Magni, L., Scattolini, R.: Model predictive control of continuous-time nonlinear systems with piecewise constant control. IEEE Transactions on Automatic Control 49, 900–906 (2004)
17. Magni, L., Sepulchre, R.: Stability margins of nonlinear receding horizon control via inverse optimality. Systems and Control Letters 32, 241–245 (1997)
18. Mayne, D.Q.: Control of constrained dynamic systems. European Journal of Control; Special issue: Fundamental Issues in Control 7, 87–99 (2001)
19. Mayne, D.Q., Rawlings, J.B., Rao, C.V., Scokaert, P.O.M.: Constrained model predictive control: Stability and optimality. Automatica 36, 789–814 (2000)
20. Murray, R.M.: Recent research in cooperative control of multi-vehicle systems. Journal of Dynamic Systems, Measurement and Control 129, 571–583 (2007)
21. Qin, S., Badgwell, T.: An overview of industrial model predictive control technology. Control Engineering Practice 11, 733–764 (2003)
22. Raimondo, D.M., Magni, L., Scattolini, R.: Decentralized MPC of nonlinear systems: An input-to-state stability approach. International Journal Of Robust And Nonlinear Control 17, 1651–1667 (2007)
23. Sontag, E.D., Sussman, H.J.: Remarks on continuous feedback. In: IEEE Conference on Decision and Control, Albuquerque, pp. 916–921 (1980)