# Distance based Dynamical System Modulation for Reactive Avoidance of Moving Obstacles

Matteo Saveriano and Dongheui Lee

*Abstract*— **An algorithm which allows the robot to avoid moving obstacles and to reach the assigned goal is proposed. For this purpose, a dynamical system (DS) modulation matrix is calculated using the distance from the obstacles and their velocity, without the need of an analytical representation of the obstacles. This matrix modulates a generic first order DS, used to generate the desired path, saving the equilibrium points of the modulated system. The effectiveness of the proposed approach is validated with numerical simulations and experiments on a 7 DOF KUKA light weight arm.**

## I. Introduction

In human-robot interaction scenarios the robot is required to adapt quickly to various situations and to eventual external disturbances ensuring the operator safety. A feasible solution for reacting in real-time to the external perturbations consists in representing the task as a dynamical system (DS). Indeed, a DS is robust against perturbations, and it ensures the convergence to the goal [1][2].

The trajectory generated from the DS must be modified, in real-time, to avoid possible collisions when unknown obstacles and humans enter in the scene. Moreover, after the avoidance, it is desirable that the robot fulfils the assigned task as long as possible.

In the literature, two categories of approaches have been proposed to generate collisions free paths: path planning approach and reactive motion generation approach. The former is capable to find, if it exists, the shortest collision free path even in very complex scenarios with multi degree-of-freedom robots [3]. These algorithms are computationally expensive and, despite the possibility to parallelise them [4], at present they can be applied only off-line.

The latter includes local algorithms which modify the robot trajectory on-line. A widely used approach is based on an artificial potential field [5]. The idea is to assign an attractive force to the goal and to shape the obstacles as repulsive forces, so as to reach the target avoiding obstacles. In [6], for example, an approach to calculate the repulsive force directly from the image plane of an RGB-D sensor is proposed. One drawback of the potential field approach is that the motion can stop in a local minima even if a collision-free path to the goal exists.

A solution to skip the local minima is proposed in [7]. The initial elastic band (collision-free path) is computed off-line using a path planning algorithm. During the motion, the band is deformed by applying repulsive forces to avoid collisions with moving and unknown obstacles. However, the described reshaping method fails to avoid obstacles coming towards the robot, and an off-line replanning step is needed [8].

The gradient of an *harmonic potential field* is used in [9] to eliminate the local minima problem. The harmonic fields are solutions of the Laplace's equation, and their analytical expression is known only for simple shaped obstacles. Because it is hard to find numerical solutions of the Laplace's equation in real-time, some algorithms to approximate complex shaped obstacles have been proposed [10]. However, these algorithms can work in real-time only in the planar case.

Other researchers propose to avoid local minima by modifying the dynamics of a particular system of differential equations. For example in [11] an additive term is applied to a discrete Dynamic Movement Primitive (DMP) [2] in order to deform the trajectory and avoid a point obstacle. The global stability of the modified system is proved with static obstacles using the *Lyapunov theorem*. In [12] a combination of potential fields and circular fields is applied to a second order system to generate a smooth collision-free path. The mentioned approaches work only with a specific dynamical system, reducing the possibility of encoding many different tasks, such as periodic motions.

A technique to modulate a generic first order DS is proposed in [13]. Given the analytical representations of the obstacles surface, a modulation matrix is computed that locally deforms the original system. This approach can be applied on a variety of DS (both stable and unstable) and it guarantees the impenetrability of static obstacles, without modifying the equilibrium points of the modulated system. This approach has been extended in [14], in which the modulation matrix is computed directly from a point cloud and the analytical representation is no longer needed.

In this paper, we further extend our previous work [14], proposing a modified modulation technique to guarantee the avoidance and impenetrability of moving obstacles. As in the static case, we show that the modulation does not affect the equilibrium points of the modulated DS. The effectiveness of our approach is proved with numerical simulations and experiments on a KUKA LWR IV+.

The rest of the paper is organised as follows. Section II describes the DS modulation with moving objects. Section III discusses the impenetrability of convex and concave obstacles. Section IV analyses the equilibrium point of the modulated system. Finally, Section V presents the experimental results, and Section VI states the conclusions and the future works.

Authors are with Fakultät für Elektrotechnik und Informationstechnik, Technische Universität München, Munich, Germany matteo.saveriano@tum.de, dhlee@tum.de.

## II. Dynamical System Modulation

### A. Problem Definition

The modulation algorithm is based on the assumption that the path to follow is generated by a first order dynamical system. This system can be autonomous (time-independent) or non-autonomous (time-dependent). Considering the position $\mathbf{p} \in \mathbb{R}^d$ of the robot as the state of the system, the DS becomes:

$$\dot{\mathbf{p}}(t) = \mathbf{f}(\mathbf{p}(t)), \qquad \text{autonomous} \tag{1}$$

$$\dot{\mathbf{p}}(t) = \mathbf{f}(\mathbf{p}(t), t), \qquad \text{non-autonomous} \tag{2}$$

where $\mathbf{f}(\cdot)$ is a continuous function and $\dot{\mathbf{p}}$ is the first time derivative of $\mathbf{p}$. Knowing the starting point $\mathbf{p}_0$ the desired trajectory can be calculated by integrating $\mathbf{f}$.

By modulating (1) or (2) with a suitable matrix $\mathbf{M}(\mathbf{p})$ by

$$\dot{\mathbf{p}} = \mathbf{M}(\mathbf{p})\mathbf{f} \tag{3}$$

one can avoid obstacles and keep the stability properties of the DS. In the following pages, $\mathbf{p}$ denotes the generic point, $\bar{\mathbf{p}}$ a point on the object surface, and $\tilde{\mathbf{p}} = \mathbf{p} - \bar{\mathbf{p}}$ a generic point with respect to $\bar{\mathbf{p}}$.

### B. Modulation with a Fixed Obstacle

We consider that a $d$-dimensional obstacle is present in the scene[1], and that the normal vector to the obstacle surface $\hat{\mathbf{n}}(\bar{\mathbf{p}}) = [\hat{n}_1(\bar{\mathbf{p}}) \ \cdots \ \hat{n}_d(\bar{\mathbf{p}})]^T$ is defined for all $\bar{\mathbf{p}}$. A tangential hyperplane can be defined at each point on the surface, using the normal vector. One particular basis of the tangential hyperplane is

$$\hat{v}_i^j(\bar{\mathbf{p}}) = \begin{cases} -\hat{n}_{i+1}(\bar{\mathbf{p}}) & j = 1 \\ \hat{n}_1(\bar{\mathbf{p}}) & j = i+1 \quad i = 1..d-1, j = 1..d \\ 0 & j \neq 1, j \neq i+1 \end{cases} \tag{4}$$

where $\hat{v}_i^j$ corresponds to the $j$-th component of the $i$-th basis vector. Then, the matrix $\mathbf{V}(\bar{\mathbf{p}}) = [\hat{\mathbf{n}}(\bar{\mathbf{p}}) \ \hat{\mathbf{v}}_1(\bar{\mathbf{p}}) \ \cdots \ \hat{\mathbf{v}}_{d-1}(\bar{\mathbf{p}})]$ is an orthonormal basis of the $d$-dimensional space.

Now, let us call $\Phi' = \Phi(\tilde{\mathbf{p}}) - \alpha$ the distance between the robot and the surface of the obstacle, where the positive scalar $\alpha$ is a *safety margin* determining how close to the object the robot can pass. The function $\Phi'$ is continuous, positive definite, and $\Phi' \to \infty$ if $\|\tilde{\mathbf{p}}\| \to \infty$. We can define the diagonal matrix $\mathbf{E}(\tilde{\mathbf{p}}) = diag\,(\lambda_1(\tilde{\mathbf{p}}), \dots, \lambda_n(\tilde{\mathbf{p}}))$, where

$$\begin{cases} \begin{cases} \lambda_1 = 1 - \dfrac{1-\epsilon}{(\Phi'+1)^{\frac{1}{\rho}}} & \dot{\mathbf{p}}^T\tilde{\mathbf{p}} < 0 \text{ or } m = 1 \\ \lambda_1 = 1 & \dot{\mathbf{p}}^T\tilde{\mathbf{p}} \geq 0 \text{ and } m = 0 \end{cases} \\ \lambda_i = 1 + \dfrac{1}{(\Phi'+1)^{\frac{1}{\rho}}} \qquad i = 2, 3, \dots, d \end{cases} \tag{5}$$

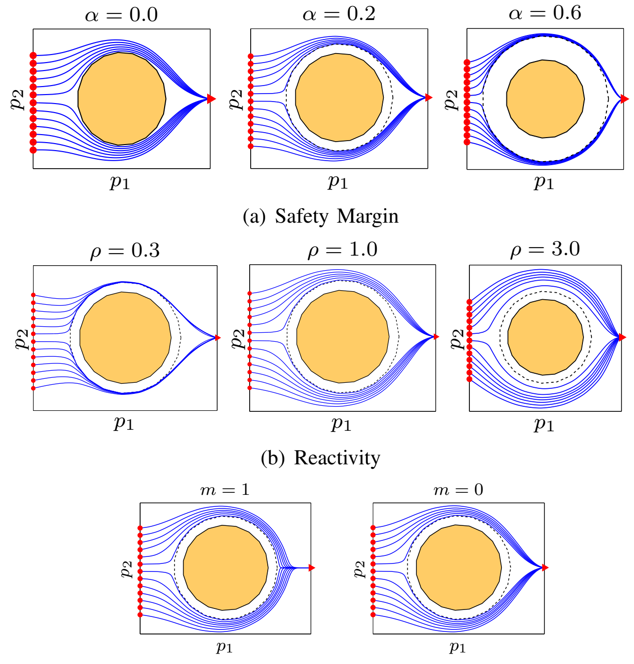
In (5), the positive scalar $\rho$ is the *reactivity* parameter, used to change the magnitude of the modulation, the boolean variable $m = 0, 1$ is used to interrupt the modulation ($m = 0$) after passing the obstacle[2] ($\dot{\mathbf{p}}^T\tilde{\mathbf{p}} < 0$), and $\epsilon > 0$ is an arbitrary small positive scalar, introduced to guarantee the positive definiteness of the modulation matrix. The effects on the modulated path, obtained with different values of *safety margin*, *reactivity* and *modulation interrupt*, are shown in Fig. 1.

Then, the modulation matrix in (3) can be calculated as

$$\mathbf{M}(\tilde{\mathbf{p}}) = \mathbf{V}(\bar{\mathbf{p}})\mathbf{E}(\tilde{\mathbf{p}})\mathbf{V}(\bar{\mathbf{p}})^{-1} \tag{6}$$

By modulating a DS with the matrix (6), it is possible to prove that a trajectory $\mathbf{p}(t)$ starting from outside an obstacle can never penetrate the convex obstacle[3]. Moreover, the modulation does not affect the equilibrium points of the modulated DS, saving its stability properties.

The modulation consists basically in a local deformation of the DS, that generates collision free paths. The effect of the modulation is maximum at the boundary of the obstacle, and vanishes for points far from it.Note that, by construction, the modulation matrix is symmetric and positive definite for all $\tilde{\mathbf{p}}$.



(a) Safety Margin

(b) Reactivity

(c) Modulation Interrupt (Off/On)

Fig. 1. Characterizing the path during the obstacle avoidance. The scalars $\alpha$ and $\rho$ are used to modify the size of the safe area and the magnitude of the modulation. The boolean value $m$ is used to interrupt the modulation after passing the obstacle.

### C. Distance and Normal Vector Estimation

The distance between the robot and the obstacle, such as the normal vector in the point of minimum distance, can be computed knowing an analytical representation of the obstacle surface [13]. In a realistic scenario, however, it is not possible to get, in real-time, an analytical representation of

[1] The extension to the multiple obstacles case is discussed in Sec. II-C.

[2] Note that only $\lambda_1(\tilde{\mathbf{p}}) = 1$, since the modulation of the other components is anyway necessary to ensure the continuity of the velocity.

[3] The impenetrability does not depend on the choice made for the base $[\hat{\mathbf{v}}_1(\bar{\mathbf{p}}) \ \cdots \ \hat{\mathbf{v}}_{d-1}(\bar{\mathbf{p}})]$.

arbitrary objects from raw sensory data. In [14] we proposed a solution to this problem, briefly described in what follows.

Let us consider a single object and a set of $\bar{\mathbf{p}}_g$, $g = 1, \ldots, G$, points on the obstacle boundary. The distance $\Phi(\tilde{\mathbf{p}})$ can be approximated with the Euclidean distance of $\mathbf{p}$ from the surface points

$$D(\tilde{\mathbf{p}}) = min\left(\sum_{i=1}^{d}(\bar{p}_{1,i} - p_i)^2, \ldots, \sum_{i=1}^{d}(\bar{p}_{G,i} - p_i)^2\right) \quad (7)$$

$$\Phi'_a = \sqrt{D(\tilde{\mathbf{p}})} - \alpha \quad (8)$$

where $\bar{p}_{g,i}$ denotes the $i$-th component of the $g$-th point in $P_b$, $p_i$ the $i$-th component of the current position, and $\alpha$ the safety margin. Hereafter, we indicate the approximated distance $\Phi'_a$ only with $\Phi'$.

To compute the modulation matrix (6), it is also necessary to estimate the normal vector at the point of minimum distance $\bar{\mathbf{p}}_m$. This estimation is carried out using a parallel implementation of the algorithm in [15], which allows us to quickly reconstruct a surface from an unordered point cloud. For each point $\bar{\mathbf{p}}_g$ a weighted least squares plane $\pi_g$ is calculated using points in a neighbourhood of $\bar{\mathbf{p}}_g$. Then, the normal at $\bar{\mathbf{p}}_g$ is chosen equal to the normal to the plane $\pi_g$. This algorithm is robust to noise, which is beneficial when using the RGB-D cameras. Compared to other approaches, it does not require that the surface is smooth nor a certain density of points.

Since the density of the point cloud is not fixed, the direction of the normal vector may vary significantly also in close points. In order to obtain a smooth modulated path, we calculate the normal $\hat{\mathbf{n}}(\bar{\mathbf{p}}_m)$ by using a weighted average on the $L$ points[4] closer to $\bar{\mathbf{p}}_m$:

$$\begin{aligned}\hat{\mathbf{n}}(\bar{\mathbf{p}}_m) &= c\hat{\mathbf{n}}(\bar{\mathbf{p}}_m) + (1-c)\hat{\mathbf{n}}_{av}(\bar{\mathbf{p}}_m) \\ &= c\hat{\mathbf{n}}(\bar{\mathbf{p}}_m) + (1-c)\frac{1}{L}\sum_{i=1, i\neq k}^{L}\hat{\mathbf{n}}(\bar{\mathbf{p}}_i)\end{aligned} \quad (9)$$

where the scalar $0 \leq c \leq 1$. To ensure the impenetrability, it must be $c = 1$ if $\Phi' = 0$. In addition, $c$ should be small when the robot is far from the obstacles, and it should continuously increase as $\Phi'$ decreases. For this reason $c$ is calculated as:

$$c = \frac{1}{(\Phi' + 1)^{\beta}} \quad (10)$$

where the scalar $\beta \geq 0$ is a tunable parameter. For $\beta = 0$, $c$ becomes 1 and the contribution of $\hat{\mathbf{n}}_{av}(\bar{\mathbf{p}}_m)$ in (9) is neglected. Fig. 2 shows that greater value of $\beta$ allows to obtain smoother trajectories, because the contribution given to the normal by $\hat{\mathbf{n}}_{av}(\bar{\mathbf{p}}_m)$ is dominant till the proximity to the surface.

When multiple obstacles exist in the work space, we simply calculate the distance from the closest point (8), (7) and the normal at this point (9), (10). So, the number of objects does not affect the performance of our algorithm.

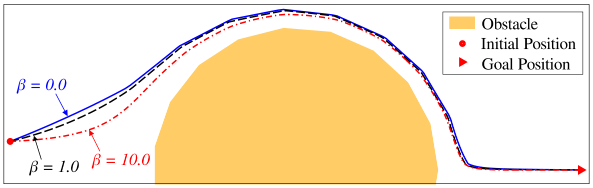


Fig. 2. Trajectories generated calculating the modulation matrix with different values of $\beta$.

### D. Extension to Moving Obstacles[5]

In the case of moving obstacles the modulation in (3) does not guarantee the impenetrability. Indeed, increase the *reactivity* or the *safety margin* does not guarantee to find a collision-free path to the goal in dynamic environments (Fig. 3(a)).

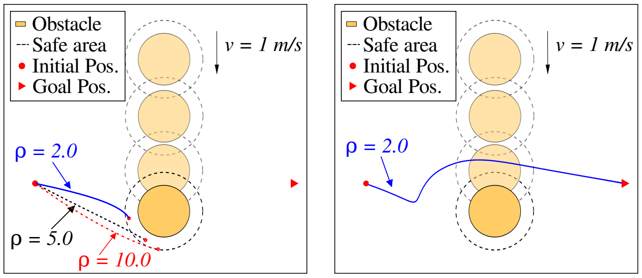
Let us consider one moving obstacle with translational and angular velocities $\dot{\mathbf{p}}_T$ and $\dot{\mathbf{p}}_A$. To guarantee the impenetrability the modulated system becomes

$$\begin{aligned}\dot{\mathbf{p}} &= \mathbf{M}(\tilde{\mathbf{p}})(\mathbf{f}(\mathbf{p},t) - \dot{\mathbf{p}}_T - \dot{\mathbf{p}}_A \times \tilde{\mathbf{p}}) + \dot{\mathbf{p}}_T + \dot{\mathbf{p}}_A \times \tilde{\mathbf{p}} \\ &= \mathbf{M}(\tilde{\mathbf{p}})(\mathbf{f}(\mathbf{p},t) - \dot{\mathbf{p}}_O) + \dot{\mathbf{p}}_O \\ &= \mathbf{M}(\tilde{\mathbf{p}})\mathbf{f}(\mathbf{p},t) + (\mathbf{I} - \mathbf{M}(\tilde{\mathbf{p}}))\dot{\mathbf{p}}_O\end{aligned} \quad (11)$$

where $\dot{\mathbf{p}}_O = \dot{\mathbf{p}}_T + \dot{\mathbf{p}}_A \times \tilde{\mathbf{p}}$, $\mathbf{I}$ is the $d$-dimensional identity matrix and $\mathbf{M}(\tilde{\mathbf{p}})$ is calculated using (6).

The term $\mathbf{M}(\mathbf{f} - \dot{\mathbf{p}}_O)$ is a modulation in the obstacle coordinate system, that guarantees the impenetrability in the current instant. The additional term $\dot{\mathbf{p}}_O$ puts the system in the robot coordinate system and guarantees collisions avoidance in the following time instant.

The effect of the proposed approach are shown in Fig. 3. A spherical obstacle is moving with velocity $v = 1\ m/s$ in the vertical direction, while the robot is moving horizontally towards the goal. Increase the reactivity does not work in this case (Fig. 3(a)), while the proposed modulation is able to find a collision-free path to the goal (Fig. 3(b)).



(a) Increase the *reactivity*

(b) Consider obstacle velocity

Fig. 3. Impenetrability and collision avoidance cannot be guaranteed only increasing the *reactivity* (a). Instead, a collision-free path is found considering the effect of the obstacle velocity (b).

When multiple obstacles exist in the work space, we simply consider the velocity of the closest (most dangerous) one to calculate the modulation in (11). The complete modulation

[4] In experiments we use $L = 1\%$ of the number of points.

[5] The same approach to consider moving obstacles has been proposed in ”S.M. Khansari-Zadeh, A Dynamical System-based Approach to Modeling Stable Robot Control Policies via Imitation Learning, PhD Thesis, 2012.“

algorithm, able to avoid, in real-time, $K$ moving obstacles, is summarised in Algorithm 1.

**Algorithm 1** DS Modulation with Moving Obstacles

```
Given $\mathbf{f}$, a point cloud, the normal vector at each point
and the velocity of the closest obstacle $\dot{\mathbf{p}}_O$
 1. Calculate $\Phi'$ using (7), and (8)
 2. Smooth the relative normal using (9), (10)
 3. Calculate $\mathbf{E} = diag\,(\lambda_1, \dots, \lambda_n)$ using (5)
 4. $\mathbf{V} = [\hat{\mathbf{n}} \;\; \hat{\mathbf{v}}_1 \cdots \hat{\mathbf{v}}_{d-1}]$
 5. $\mathbf{M} = \mathbf{V}\mathbf{E}\mathbf{V}^{-1}$
return $\dot{\mathbf{x}} = \mathbf{M}\mathbf{f} + (\mathbf{I} - \mathbf{M})\dot{\mathbf{p}}_O$
```

## III. Impenetrability

The impenetrability of convex and concave object has been proved in [13] and [14] respectively. In this section we recall those results, adapting them to the case of moving obstacles.

### A. Impenetrability of Convex Obstacles

Let us assume that the robot is on the boundary of the unique, fixed, convex obstacle in the scene. The impenetrability is ensured if the velocity of the robot along the obstacle normal direction vanishes:

$$\hat{\mathbf{n}}(\bar{\mathbf{p}})^T \dot{\mathbf{p}} = 0, \qquad \forall \bar{\mathbf{p}} \tag{12}$$

Substituting (3) and (6) in (12) yields:

$$\begin{aligned} \hat{\mathbf{n}}(\bar{\mathbf{p}})^T \dot{\mathbf{p}} &= \hat{\mathbf{n}}(\bar{\mathbf{p}})^T \mathbf{V}(\bar{\mathbf{p}})\mathbf{E}(\tilde{\mathbf{p}})\mathbf{V}(\bar{\mathbf{p}})^{-1}\mathbf{f} \\ &= [\epsilon \;\; \mathbf{0}_{d-1}^T]\mathbf{V}(\bar{\mathbf{p}})^{-1}\mathbf{f} \approx 0 \end{aligned} \tag{13}$$

where we considered that $\hat{\mathbf{n}}(\bar{\mathbf{p}})$ is equal to the first column of $\mathbf{V}(\bar{\mathbf{p}})$, that the columns of $\mathbf{V}(\bar{\mathbf{p}})$ are orthogonal, and that $\epsilon$ is arbitrary small.

In real scenarios, choosing $\epsilon \approx 0$ and a safety margin $\alpha > 0$, the residual motion in the normal direction can be neglected and the impenetrability ensured.

The impenetrability with moving obstacles is obtained modulating the system using (11). Indeed, the term $\mathbf{M}(\mathbf{f} - \dot{\mathbf{p}}_O)$ prevents the motion along the surface normal direction, guaranteeing the impenetrability in the current time instant $t_i$. The additional term $\dot{\mathbf{p}}_O$ guarantees to avoid collisions (due to the obstacle motion) in the next time instant $t_{i+1}$.

### B. Impenetrability of Concave Obstacles

In the static case, if (12) holds, the speed in a point $\bar{\mathbf{p}}$ on the boundary of the object has significant components only on the tangential hyperplane. Since the tangential hyperplane never intersects a convex surface, the impenetrability of a convex obstacle can be immediately assumed[6].

If the object is concave, the tangential hyperplane can intersect the surface, as shown in Fig. 4. Therefore, the point $\mathbf{p} = \bar{\mathbf{p}} + \mathbf{M}(\bar{\mathbf{p}})\mathbf{f}\delta t$, calculated by integrating (3), may be located within the object. To prove the impenetrability, consider the set $I = \{\mathbf{p}_I | \Phi(\mathbf{p}_I) \geq 0, \forall \mathbf{p}_I \in \mathbb{R}^n\}$ that is the set of all the points external or belonging to the surface. A subset of $I$ is defined by

$$J_r(\bar{\mathbf{p}}) = \left\{\mathbf{p}_{J_r} | r \geq max(\mathbf{p} - \bar{\mathbf{p}}), \forall \mathbf{p}_{J_r} \in I \subset \mathbb{R}^n\right\} \tag{14}$$

where $\mathbf{p} - \bar{\mathbf{p}} = \mathbf{M}(\bar{\mathbf{p}})\mathbf{f}\delta t$. By construction, $J_r(\bar{\mathbf{p}})$ is an intersection-free neighbourhood of $\bar{\mathbf{p}}$.

The impenetrability of a concave object is ensured if a neighbourhood like (14) exists for each point on the boundary of the obstacle. If this holds, we can conclude that each point $\mathbf{p}$, calculated by integrating (3), is external or belonging to the surface. In the case of moving obstacles, the same conditions still hold substituting $\mathbf{p} - \bar{\mathbf{p}} = \mathbf{M}(\mathbf{f} - \dot{\mathbf{p}}_O)\delta t + \dot{\mathbf{p}}_O\delta t$ in (14).

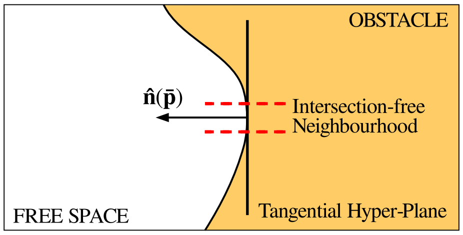


Fig. 4. In a concave object the tangent plane can intersect the surface. If the surface is smooth, a neighbourhood of the point of tangency with no intersections (here is the area between the red dashed lines) can exist.

## IV. Equilibria of the Modulated System

Let us assume that the velocity of the obstacle $\dot{\mathbf{p}}_O$ is continuous and bounded[7] $(0 \leq \|\dot{\mathbf{p}}_O\| \leq \dot{p}_{max})$ for all $t \geq t_0$. Moreover, let us call $\hat{\mathbf{p}}$ the globally asymptotically stable equilibrium for the original system (1) or (2). Since the system is globally asymptotically stable, the velocity vanishes only at the equilibrium point $\hat{\mathbf{p}}$

$$\mathbf{f}(\hat{\mathbf{p}}) = \mathbf{0}, \qquad \text{autonomous} \tag{15}$$

$$\mathbf{f}(\hat{\mathbf{p}}, t) = \mathbf{0}, \quad \forall t \geq t_0, \qquad \text{non-autonomous} \tag{16}$$

Under these assumptions, an augmented version of the original system in the form

$$\dot{\boldsymbol{\xi}} = \begin{bmatrix} \dot{\phi} \\ \dot{\mathbf{p}} \end{bmatrix} = \begin{bmatrix} -\alpha(\phi - 1) \\ \mathbf{f} \end{bmatrix} \tag{17}$$

has a globally asymptotically stable equilibrium $\hat{\boldsymbol{\xi}} = [1 \; \hat{\mathbf{p}}^T]^T$ if $\alpha > 0$.

Using (17), and assuming that $\phi(0) = 1$, the modulated system (11) can be rewritten as

$$\dot{\boldsymbol{\xi}} = \underbrace{\left[\begin{array}{c|c} \alpha & \mathbf{0}^T \\ \hline (\mathbf{M} - \mathbf{I})\dot{\mathbf{p}}_O & \mathbf{M} \end{array}\right]}_{\boldsymbol{\Lambda}(\mathbf{p},t)} \begin{bmatrix} -\phi(t) \\ \mathbf{f} \end{bmatrix} + \begin{bmatrix} \alpha \\ \mathbf{0} \end{bmatrix} \tag{18}$$

Considering that the lower block-triangular matrix $\boldsymbol{\Lambda}$ in (18) is full rank ($\alpha > 0$ and $\mathbf{M}$ positive definite) for all $\mathbf{p}$, $t \geq t_0$,

[6] Note that the time discrete implementation of the algorithm can compromise the impenetrability if the integration time step is not sufficiently small (so that it makes sense to assume to work in the continuous time). In all experiments, it will be used $\delta t = 1ms$.

[7] These assumptions are not restrictive since the obstacles obey to the Newtons's laws and there are no impacts.

we can conclude that the modulated system (18) has the same equilibrium of (17). Indeed, the velocity $\dot{\boldsymbol{\xi}}$ vanishes only at $\hat{\boldsymbol{\xi}} = [1 \; \hat{\mathbf{p}}^T]^T$. This means that the modulation does not affect the equilibria of the modulated system.

Despite the proposed algorithm saves the equilibrium points of the modulated DS, the convergence to the goal can be, in some cases, really slow. Let us consider a static scenario with the robot lying on the obstacle surface, and assume that the vector field $\mathbf{f}$ is parallel to the obstacle normal vector. These condition can be expressed as:

$$\hat{\mathbf{n}}(\bar{\mathbf{p}})^T \frac{\mathbf{f}}{\|\mathbf{f}\|} = \pm 1 \quad \text{and} \quad \Phi(\tilde{\mathbf{p}}) = \Phi(\bar{\mathbf{p}}) = 0 \qquad (19)$$

Considering that $\mathbf{V}(\bar{\mathbf{p}}) = [\hat{\mathbf{n}}(\bar{\mathbf{p}}) \;\; \hat{\mathbf{v}}_1(\bar{\mathbf{p}}) \;\; \cdots \;\; \hat{\mathbf{v}}_{d-1}(\bar{\mathbf{p}})]$, it is easy to show that the velocity components along the tangential directions $\hat{\mathbf{v}}_i$ are zero. Since, from (13), $\hat{\mathbf{n}}^T\dot{\mathbf{p}} \approx 0$, the robot will, in practise, stuck in a spurious equilibrium[8].

An on-line algorithm to escape this equilibria has been proposed in [13] and summarised in Algorithm 2. First, the algorithm detects when the robot is in a local minimum, checking the conditions $\dot{\mathbf{p}}_i \leq \tau_\epsilon$ and $\Phi'(\tilde{\mathbf{p}}_i) = 0$, where $\tau_\epsilon$ is a constant depending on the chosen $\epsilon$. Then, a small perturbation $\gamma$ is applied in one of the tangential directions $\hat{\mathbf{v}}_i$, until the robot exits from the basin of attraction of the spurious equilibrium. The positive scalar $\gamma$ controls the amplitude of the movements along $\hat{\mathbf{v}}_i$. Small values of $\gamma$ reduce the drift due to the integration error (depending on the $\delta t$), guaranteeing a safe avoidance motion. However, a very small value of $\gamma$ highly reduces the avoiding speed and it is not recommended in real-time applications.

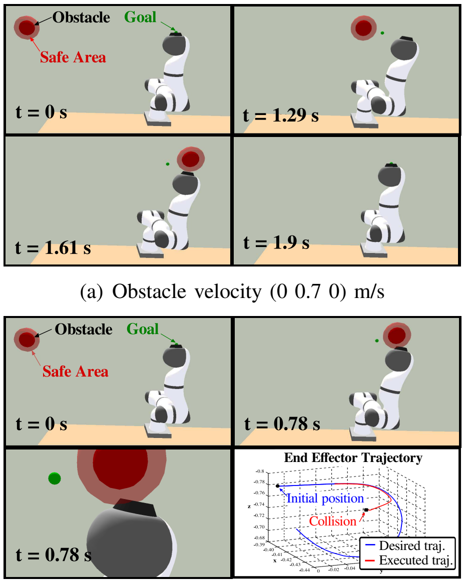
**Algorithm 2** Avoid Slow Convergence

```
Given $\tilde{\mathbf{p}}_i$, $\dot{\mathbf{p}}_i$ and the integration time step $\delta t$
if $\dot{\mathbf{p}}_i \leq \tau_\epsilon$ and $\Phi'_i = 0$ then
    Choose one of the tangential directions $\hat{\mathbf{v}}_i$
    Define a (small) positive scalar $\gamma > 0$
    while continue do
        $\mathbf{p}_{i+1} \rightarrow \mathbf{p}_i + \gamma\hat{\mathbf{v}}_i\delta t$
        Calculate $\dot{\mathbf{p}}_{i+1}$ using (11)
        if $\hat{\mathbf{n}}^T\dot{\mathbf{p}}_{i+1} > \tau_\epsilon$ or $\hat{\mathbf{v}}_i^T\dot{\mathbf{p}}_{i+1} > 0$ then
            continue = false
        end if
        $t \rightarrow t + 1$
    end while
end if
```

## V. Experimental Results

### A. Numerical Simulations

*Simulation 1:* The robot task consists in keeping the end-effector in a fixed position, while a fast spherical obstacle moves toward it. The initial distance between the robot and the obstacle is $1m$, and the obstacle moves with a constant velocity along a fixed direction.



(a) Obstacle velocity (0 0.7 0) m/s

(b) Obstacle velocity (0 1.4 0) m/s

Fig. 5. The robot has to avoid a fast obstacle and return to the initial position. The obstacle is avoided until its velocity reaches 1.4 $m/s$.

The end-effector trajectory is generated integrating the linear DS $\dot{\mathbf{p}} = 3(\mathbf{g} - \mathbf{p})$, where $\mathbf{g}$ is the goal position. To consider the physical limitations of the robot, we generate the desired joint trajectories using a first-order CLICK algorithm [16], saturating the joint positions (velocity) that exceed the limits. For the KUKA LWR IV+ robot, used in the simulation, the joint ranges are $\mathbf{q}_{max} = -\mathbf{q}_{min} = [170, 120, 170, 120, 170, 120, 170] \; deg$, and the joint maximum velocity $\dot{\mathbf{q}}_{max} = -\dot{\mathbf{q}}_{min} = [100, 110, 100, 130, 130, 180, 180] \; deg/s$.

We performed several tests varying the velocity in the range $0.5 \sim 1.4 \; m/s$. The robot behaviour in four different time instants, and for two different trials, is shown in Fig. 5. The results are obtained choosing $\beta = 10, \; \alpha = 0.03, \; \rho = 3, \; m = 0$.

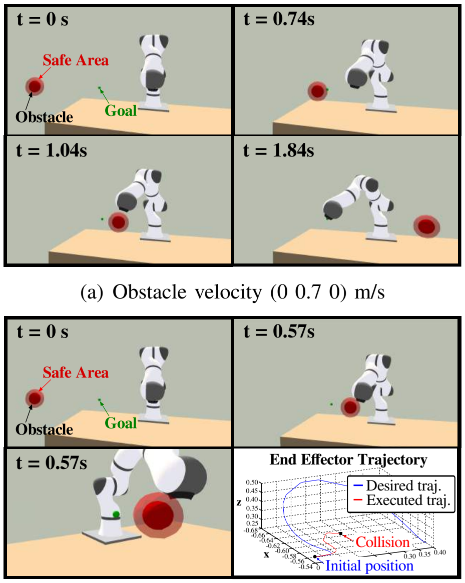
The robot is able to avoid the obstacle, coming back in the goal position, until the obstacle velocity reaches $1.4 \; m/s$. In this case, indeed, the joint limits are exceeded, and the robot is not able to follow the desired trajectory. The desired trajectory (blue) and the path the robot follows due to the joints saturation (red) are depicted in Fig. 5(b).

*Simulation 2:* In this simulation the task is to reach a goal while avoiding a fast moving obstacle. The parameters are the same as in Simulation 1. We performed several tests varying the velocity in the range $0.5 \sim 1.3 \; m/s$. The robot is able to avoid the obstacle until its velocity reaches $1.3 \; m/s$. The robot behaviour in four different time instants, and for two different trials, as well as the desired and executed trajectories in the failure case, are shown in Fig. 6.

### B. Human-Robot collisions avoidance

In this experiment the robot has to keep the end-effector in a fixed position while the user tries to hit it. The desired Cartesian pose is sent at $1000 Hz$ using the *Fast Reaserch*

[8] Notice that we assumed only fixed obstacle because this problem hardly affects the modulation in (11) due to the contribution given by $\dot{\mathbf{p}}_O$.



(a) Obstacle velocity (0 0.7 0) m/s

(b) Obstacle velocity (0 1.3 0) m/s

Fig. 6. The robot has to avoid a fast obstacle and converge to the goal position. The obstacle is avoided until its velocity reaches 1.3 $m/s$.

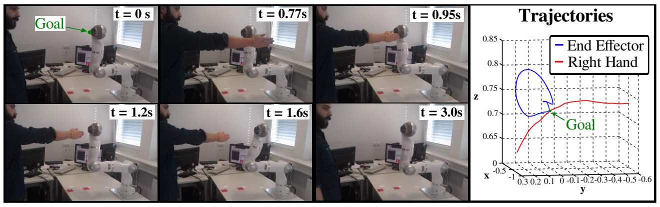


Fig. 7. The robot has to avoid collisions with the human and return to the initial position. The norm of the estimated hand velocity comes from 0.45 to 0.6 $m/s$.

*Interface* [17]. The robot trajectory is generated integrating the linear DS $\dot{\mathbf{p}} = 3(\mathbf{g} - \mathbf{p})$, where $\mathbf{g}$ is the goal (initial) position.

The human is tracked at $30Hz$ using an RBG-D camera and the *OpenNI* library[9]. A Kalman filter is used to reduce the noise on the hand position estimation and to estimate the hand velocity. To implement the filter we assumed a constant velocity in each time step. The robot is removed from the sensor depth map using a shader-based filter[10].

The robot behaviour in six different time instants, together with the robot end effector and the human hand trajectories, are shown in Fig. 7. The results are obtained choosing $\beta = 10,\ \alpha = 0.05,\ \rho = 0.5\ m = 0$.

## VI. Conclusion and Future Work

In this paper we presented a distance based modulation approach to collision avoidance in dynamic scenarios. Given the closest obstacle position with respect to the robot's end-effector, the unit normal vector at the closest point, and the obstacle velocity, a modulation matrix is computed that guarantees the impenetrability without affecting the equilibrium points of the modulated system. The of the proposed approach is shown using numerical simulations and experiment on a real robot.

In the future, we will focus on testing the Distance Modulation in more complex scenarios, with multiple moving obstacles. This approach will also be extended to avoid collisions with the whole robot by projecting the movements in the robot null space.

## Acknowledgements

This work has been partially supported by the European Community within the FP7 ICT-287513 SAPHARI project and Technical University Munich, Institute for Advanced Study, funded by the German Excellence Initiative. The authors are solely responsible for its content.

## References

[1] S. M. Khansari-Zadeh and A. Billard, "Learning stable non-linear dynamical systems with gaussian mixture models," *Transaction on Robotics*, vol. 27, no. 5, pp. 943–957, 2011.

[2] A. Ijspeert, J. Nakanishi, P. Pastor, H. Hoffmann, and S. Schaal, "Dynamical Movement Primitives: learning attractor models for motor behaviors," *Neural Computation*, no. 25, pp. 328–373, 2013.

[3] H. Choset, K. M. Lynch, S. Hutchinson, G. A. Kantor, W. Burgard, L. E. Kavraki, and S. Thrun, *Principles of Robot Motion: Theory, Algorithms, and Implementations*. MIT Press, 2005.

[4] A. Vazquez-Otero, J. Faigl, and A. P. Munuzuri, "Path planning based on reaction-diffusion process," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012, pp. 896–901.

[5] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," *International Journal of Robotics Research*, vol. 5, no. 1, pp. 90–98, 1986.

[6] F. Flacco, T. Kröger, A. De Luca, and O. Khatib, "A depth space approach to human-robot collision avoidance," in *Proc. IEEE International Conference on Robotics and Automation*, 2012, pp. 338–345.

[7] O. Brock and O. Khatib, "Elastic strips: A framework for motion generation in human environments," *The International Journal of Robotics Research*, vol. 21, no. 12, pp. 1031–1052, 2002.

[8] E. Yoshida and F. Kanehiro, "Reactive robot motion using path replanning and deformation," in *Proc. IEEE International Conference on Robotics and Automation*, 2011, pp. 5456 – 5462.

[9] H. Feder and J.-J. Slotine, "Real-time path planning using harmonic potentials in dynamic environments," in *Proc. IEEE International Conference on Robotics and Automation*, 1997, pp. 874–881.

[10] R. Daily and D. Bevly, "Harmonic potential field path planning for high speed vehicles," in *American Control Conference*, 2008, pp. 4609–4614.

[11] H. Hoffmann, P. Pastor, D.-H. Park, and S. Schaal, "Biologically-inspired dynamical systems for movement generation: automatic real-time goal adaptation and obstacle avoidance," in *Proc. IEEE international conference on Robotics and Automation*, 2009, pp. 1534–1539.

[12] S. Haddadin, S. Belder, and A. Albu-Schäffer, "Dynamic motion planning for robots in partially unknown environments," in *IFAC World Congress*, 2011, pp. 6842–6850.

[13] S. M. Khansari-Zadeh and A. Billard, "A dynamical system approach to realtime obstacle avoidance," *Autonomous Robots*, vol. 32, no. 4, pp. 433–454, 2012.

[14] M. Saveriano and D. Lee, "Point cloud based dynamical system modulation for reactive avoidance of convex and concave obstacles," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2013, pp. 5380–5387.

[15] Z. C. Marton, R. B. Rusu, and M. Beetz, "On fast surface reconstruction methods for large and noisy datasets," in *Proc. IEEE International Conference on Robotics and Automation*, 2009, pp. 3218–3223.

[16] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Robotics - Modelling, Planning and Control*. Springer, 2009.

[17] G. Schreiber, A. Stemmer, and R. Bischoff, "The fast research interface for the kuka lightweight robot," in *Proc. IEEE ICRA Workshop on Innovative Robot Control Architectures for Demanding (Research) Applications*, 2010, pp. 15–21.

[9] `www.openni.org`

[10] `github.com/jhu-lcsr-forks/realtime_urdf_filter`