

Dynamic Movement Primitives

Michele Ginesi, Ph.D.

<https://sites.google.com/view/michele-ginesi>

Modeling Week 2022 - University of Verona

These notes are only a quick recall of the DMP theory. Please see the references for an in-depth discussion.

Dynamic Movement Primitives (DMPs) is a framework for trajectory learning based on a second order Ordinary Differential Equation (ODE) of spring-mass-damper type in which a *forcing* (also called *perturbation*) term is “learned” in such a way to encode the shape of the trajectory of the solution.

1 Original Formulation

DMPs [7, 8] are used to model both rhythmic and discrete movements (in this work we will focus on the latter). They consist of a system of second order ODEs (one for each dimension of the ambient space) of mass-spring-damper type with a perturbation term.

The aim of DMPs is to model the perturbation term in such a way to being able to generalize the trajectory to new start and goal position while maintaining the shape.

The one-dimensional formulation of DMP is:

$$\tau \dot{v} = K(g - x) - Dv + (g - x_0)f(s) \quad (1a)$$

$$\tau \dot{x} = v \quad (1b)$$

where $x \in I \subseteq \mathbb{R}$ and $v \in \mathbb{R}$ are respectively position and velocity of a prescribed point of the system. s is a re-parametrization of time $t \in [0, T]$ defined by the so-called *canonical system*:

$$\tau \dot{s} = -\alpha s, \quad \alpha \in \mathbb{R}^+. \quad (2)$$

$x_0 \in I$ is the initial position, and $g \in \mathbb{R}$ is the *goal* position. $K, D \in \mathbb{R}^+$ are, respectively, the spring and damping term, chosen in such a way that the associated homogeneous system is critically damped: $D = 2\sqrt{K}$. $\tau \in \mathbb{R}^+$ is a temporal scaling factor, f is a real-valued, non-linear *perturbation term*,

The forcing term f is written in terms of basis functions as

$$f(s) = \frac{\sum_{i=0}^N \omega_i \psi_i(s)}{\sum_{i=0}^N \psi_i(s)} s, \quad (3)$$

where

$$\psi_i(s) = \exp(-h_i(s - c_i)^2) \quad (4)$$

are Gaussian basis functions with centers c_i and widths h_i defined respectively as:

$$c_i = \exp\left(-\alpha i \frac{T}{N}\right), \quad i = 0, 1, \dots, N, \quad (5)$$

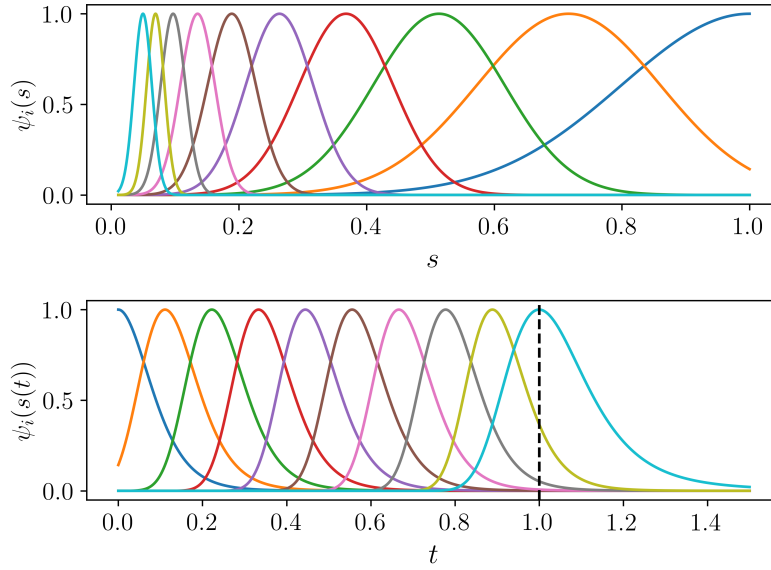


Figure 1: Example of Gaussian Basis Functions. The upper plot shows the evolution of the functions w.r.t. the parameter s , while the lower plot shows them as function of t . The parameters are: $T = 1$, $\alpha = 3$, and $N = 9$.

and

$$\begin{aligned} h_i &= (c_{i+1} - c_i)^{-2}, \quad i = 0, 1, \dots, N-1, \\ h_N &= h_{N-1}. \end{aligned} \tag{6}$$

Figure 1 shows an example of such basis functions. We remark the following properties:

- being defined as Gaussian function with respect to s , each basis function is symmetric in s ;
- the set of centers c_i is equispaced in t ;
- since s is a decreasing function of t , the order of the basis functions change from s to t : the first (left-most) basis function in t is the last (right-most) in s .

The learning process focuses on the computation of the weights ω_i that best approximate the desired forcing term, obtained by solving (1a) for f . We will discuss the details in Section 3.

One of the main advantages of DMPs (1) is that convergence to the goal position is ensured.

Proposition 1.1. *Dynamical system (1) converges to the unique equilibrium $(x, v) = (g, 0)$.*

Proof. Let us firstly prove the result in the case $f(s) \equiv 0$. With this assumption, equation (1) reads

$$\tau \dot{v} = K(g - x) - Dv \tag{7a}$$

$$\tau \dot{x} = v \tag{7b}$$

To compute the equilibria we must solve

$$\begin{cases} K(g - x) - Dv = 0 \\ v = 0 \end{cases}.$$

Substituting the second identity into the first, we get

$$K(g - x) = 0,$$

which yields $x = g$. Thus the unique equilibrium is $x = g, v = 0$.

Let us now re-write the dynamical system (7) as

$$\tau^2 \ddot{x} + \tau D \dot{x} - K(g - x) = 0,$$

and let us substitute $z = x - g$ obtaining

$$\tau^2 \ddot{z} + \tau D \dot{z} + Kz = 0.$$

Using the eigenvalues method we get that the general solution is:

$$z = c_1 \exp(\lambda_1 t) + c_2 \exp(\lambda_2 t)$$

with

$$\lambda_{1,2} = \frac{-\tau D \pm \sqrt{\tau^2 D^2 - 4\tau^2 K}}{2\tau^2}.$$

From condition $D = 2\sqrt{K}$ we obtain $\lambda_1 = \lambda_2 = -\frac{D}{2\tau}$, from which $z = c \exp(-\lambda_1 t)$. Since λ_1 is negative, we have that \dot{z} is negative for any time t . Thus the equilibrium is stable.

The result for the non-homogeneous system (1) follows from the fact that $f(s)$ goes to 0 as $t \rightarrow \infty$. \square

A second, important, property of dynamical system (1) is that it is invariant under translations.

Proposition 1.2. *Dynamical System (1) is invariant under translations.*

Proof. To prove it, let us consider a translation $x' = x - x^*$ with fixed x^* . From this follows $x'_0 = x_0 - x^*$ and $g' = g - x^*$. By substituting in (1b) we obtain

$$\tau \dot{x}' = \tau \frac{d(x - x^*)}{dt} = v$$

since x^* is fixed.

Similarly, by substituting in (1a) we obtain

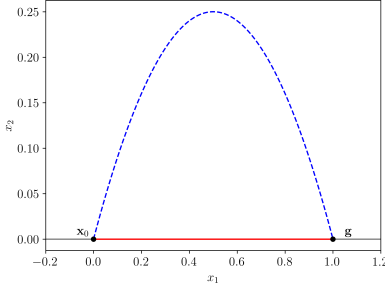
$$\begin{aligned} \tau \dot{v} &= K(g' - x') - Dv + (g' - x'_0)f(s) \\ &= K(g - x^* - x + x^*) - Dv + (g - x^* - x_0 + x^*)f(s) \\ &= K(g - x) - Dv + (g - x_0)f(s) \end{aligned}$$

Thus, the vector field governing the evolution of the solution is the same of (1), hence the thesis. \square

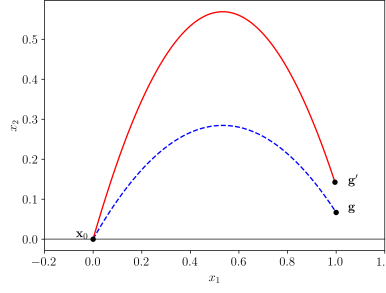
2 New Formulation

Three main drawbacks characterize DMP formulation (1) (see [10] and Figure 2):

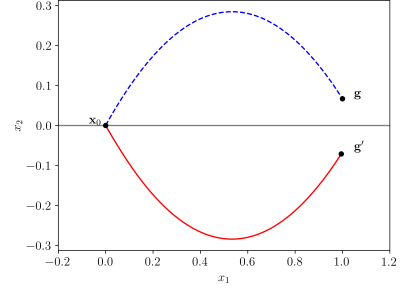
1. if the goal position coincide with the initial one, $g = x_0$, the perturbation term does not contribute to the evolution of the solution;
2. if $g - x_0$ is “small” the scaling of the perturbation f by $g - x_0$ may produce unexpected (and undesired) behaviors when changing it;
3. if the scaling factor $g - x_0$ changes sign from the learned trajectory to the new one, the trajectory will result mirrored, since the forcing term f changes sign.



(a) Drawback 1.



(b) Drawback 2.



(c) Drawback 3.

Figure 2: Drawbacks of DMP formulation (1). In all plots, the blue dashed line is the desired behavior, while the solid red line shows the obtained DMP.

To overcome these disadvantages, a modified formulation has been proposed in [9, 10, 6] by considering the system

$$\tau \dot{v} = K(g - x) - Dv - K(g - x_0)s + Kf(s) \quad (8a)$$

$$\tau \dot{x} = v \quad (8b)$$

where the evolution of s is still described by the canonical system (2), and the forcing term is still written as in (3). The most important modification lies in the fact that in this new formulation the forcing term f is not multiplied by the quantity $g - x_0$. The additional term $K(g - x_0)s$ is added to avoid initial abrupt accelerations.

While overcoming the aforementioned disadvantages, this new formulation maintains the desirable properties of the original DMP formulation.

Proposition 2.1. *Dynamical system (8) converges to the unique equilibrium $(x, v) = (g, 0)$.*

Proof. The proof follows the same argument of Proposition 1.1. □

Proposition 2.2. *Dynamical system (8) is invariant under translations.*

Proof. The proof follows the same argument of Proposition 1.2. □

We remark that, in d -dimensions, we make d decoupled copies of system (8). We can then write system (8) in vector form as

$$\tau \dot{\mathbf{v}} = \mathbf{K}(\mathbf{g} - \mathbf{x}) - \mathbf{D}\mathbf{v} - \mathbf{K}(\mathbf{g} - \mathbf{x}_0)s + \mathbf{K}\mathbf{f}(s) \quad (9a)$$

$$\tau \dot{\mathbf{x}} = \mathbf{v} \quad (9b)$$

where $\mathbf{x}, \mathbf{v}, \mathbf{g}, \mathbf{x}_0, \mathbf{f}(s) \in \mathbb{R}^d$, and $\mathbf{K}, \mathbf{D} \in \mathbb{R}^{d \times d}$ are diagonal matrices $\mathbf{K} = \text{diag}(K_1, K_2, \dots, K_d)$, $\mathbf{D} = \text{diag}(D_1, D_2, \dots, D_d)$ to maintain each direction decoupled from the others. Thanks to the decoupling of the system, the forcing term can be learned direction by direction. Additionally, still thanks to the decoupling of the system, Proposition 2.1 and Proposition 2.2 are trivially extended to the vector formulation (9).

Moreover, this new formulation is invariant under general affine transformations:

Theorem 2.1 (Hoffmann et al. [6]). *DMP formulation (9) is invariant under affine transformations of the coordinate system.*

Proof. Since invariance under translations has already been proved in Proposition 2.2, we need to prove the invariance only under linear invertible transformations.

To do so, let us consider an invertible matrix $\mathbf{S} \in \mathbb{R}^{d \times d}$, and let us perform the following substitutions:

$$\mathbf{x} = \mathbf{S}\mathbf{x}', \quad \mathbf{x}_0 = \mathbf{S}\mathbf{x}_0', \quad \mathbf{g} = \mathbf{S}\mathbf{g}', \quad \mathbf{v} = \mathbf{S}\mathbf{v}', \quad \dot{\mathbf{v}} = \mathbf{S}\dot{\mathbf{v}}', \quad \mathbf{K} = \mathbf{S}\mathbf{K}'\mathbf{S}^{-1}, \quad \mathbf{D} = \mathbf{S}\mathbf{D}'\mathbf{S}^{-1}. \quad (10)$$

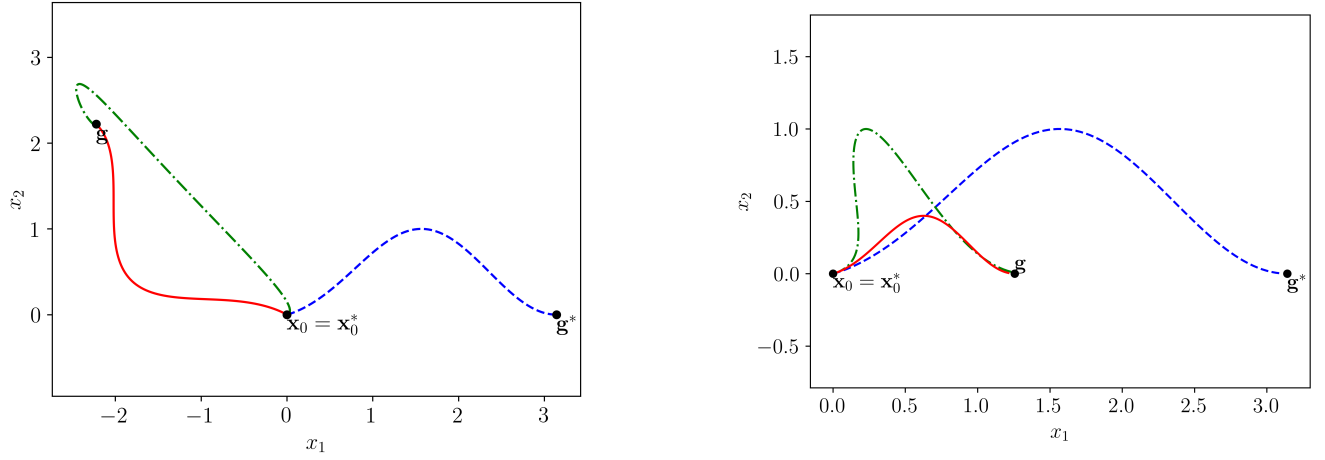


Figure 3: Different behaviors between DMPs formulation (9) (green dash-dotted) and the formulation from Theorem 2.1 with a roto-dilatation affine transformation. The learned behavior is shown in blue.

With this substitution (9b) reads

$$\tau \mathbf{S} \dot{\mathbf{x}}' = \mathbf{S} \mathbf{v}' \quad \Leftrightarrow \quad \tau \dot{\mathbf{x}}' = \mathbf{v}'.$$

Now, by substituting (10) into (9a) we obtain

$$\begin{aligned} \mathbf{f}(s) &= \mathbf{K}^{-1} (\tau \dot{\mathbf{v}} - \mathbf{K}(\mathbf{g} - \mathbf{x}) + \mathbf{D}\mathbf{v} + \mathbf{K}(\mathbf{g} - \mathbf{x}_0)s) \\ &= \tau \mathbf{K}^{-1} \dot{\mathbf{v}} - (\mathbf{g} - \mathbf{x}) + \mathbf{K}^{-1} \mathbf{D}\mathbf{v} + (\mathbf{g} - \mathbf{x}_0)s \\ &= \tau \underbrace{\mathbf{S} \mathbf{K}'^{-1} \mathbf{S}^{-1}}_{\mathbf{K}^{-1}} \underbrace{\mathbf{S} \dot{\mathbf{v}}'}_{\dot{\mathbf{v}}} - \underbrace{\mathbf{S}(\mathbf{g}' - \mathbf{x}')}_{\mathbf{g} - \mathbf{x}} + \underbrace{\mathbf{S} \mathbf{K}'^{-1} \mathbf{S}^{-1}}_{\mathbf{K}^{-1}} \underbrace{\mathbf{S} \mathbf{D}' \mathbf{S}^{-1}}_{\mathbf{D}} \underbrace{\mathbf{S} \mathbf{v}'}_{\mathbf{v}} + \underbrace{\mathbf{S}(\mathbf{g}' - \mathbf{x}_0')}_{\mathbf{g} - \mathbf{x}_0} s \\ &= \tau \mathbf{S} \mathbf{K}'^{-1} \dot{\mathbf{v}}' - \mathbf{S}(\mathbf{g}' - \mathbf{x}') + \mathbf{S} \mathbf{K}'^{-1} \mathbf{D}' \mathbf{v}' + \mathbf{S}(\mathbf{g}' - \mathbf{x}_0')s \\ &= \mathbf{S} (\tau \mathbf{K}'^{-1} \dot{\mathbf{v}}' - (\mathbf{g}' - \mathbf{x}') + \mathbf{K}'^{-1} \mathbf{D}' \mathbf{v}' + (\mathbf{g}' - \mathbf{x}_0')s) \\ &= \mathbf{S} \mathbf{f}'(s). \end{aligned}$$

Thus, if we substitute (10) and

$$\mathbf{f} = \mathbf{S} \mathbf{f}' \tag{11}$$

in (9), we obtain for the prim variables the same equation (9), thus proving the invariance under these transformations. \square

This property allows to obtain an invariance under different reference frame [5] as long as we are able to define an invertible matrix \mathbf{S} such that $(\mathbf{g}' - \mathbf{x}_0') = \mathbf{S}(\mathbf{g} - \mathbf{x}_0)$.

For instance, matrix \mathbf{S} can be a "roto-dilatation" matrix [5]:

$$\mathbf{S} = \frac{\|\mathbf{g}' - \mathbf{x}_0'\|}{\|\mathbf{g} - \mathbf{x}_0\|} \mathbf{R},$$

where \mathbf{R} is a rotation matrix that maps the versor $\widehat{\mathbf{g} - \mathbf{x}_0}$ to the versor $\widehat{\mathbf{g}' - \mathbf{x}_0'}$.

An example on how this formulation results in a better behavior can be seen in Figure 3. As it can be seen, the improved formulation maintains a trajectory of similar shape of the learned one even when there is a significant change in goal position.

3 Learning of the Forcing Term

In the following, we focus on formulations (8) and (9), remarking that both the learning and execution phases can straightforward be applied to formulation (1).

In the learning phase, we record a trajectory $\tilde{\mathbf{x}}(t)$ and its velocity $\tilde{\mathbf{v}}(t)$, $t \in [0, T]$. Then we set $\tau = 1$. We remark that s can be expressed analytically. Indeed, since τ and α are constant, the solution to the canonical system 2 is

$$s = \exp(-\alpha t), \quad t \in [0, T].$$

The forcing term can then be computed using (8a) as

$$\tilde{f}_j(s) = \frac{\dot{v}_j + D_j \tilde{v}_j}{K_j} - (g_j - \tilde{x}_j) + (g_j - \tilde{x}_{0j})s,$$

where the subscript j refers to the component of the vector (we recall that \mathbf{K} and \mathbf{D} are diagonal matrices, otherwise it would not be possible to decouple the problem along each direction).

At this point, we have to compute the weights ω_i that best approximate the desired forcing term \tilde{f} (we drop the index of direction for notation simplicity). Thus, we are searching for the vector $\boldsymbol{\omega}^* \in \mathbb{R}^{N+1}$ that realizes

$$\boldsymbol{\omega}^* = \arg \min_{\boldsymbol{\omega} \in \mathbb{R}^{N+1}} \left\| \frac{\sum_{i=1}^N \omega_i \psi_i(s)}{\sum_{i=0}^N \psi_i(s)} s - \tilde{f}(s) \right\|_2^2 \quad (12a)$$

$$= \arg \min_{\boldsymbol{\omega} \in \mathbb{R}^{N+1}} \underbrace{\int_{s_0}^{s_1} \left(\frac{\sum_{i=0}^N \omega_i \psi_i(s)}{\sum_{i=0}^N \psi_i(s)} s - \tilde{f}(s) \right)^2 ds}_{F(\boldsymbol{\omega})}. \quad (12b)$$

Function $F : \mathbb{R}^{N+1} \rightarrow \mathbb{R}^+$ is continuous and strictly convex, which means that the minimum exists and is unique. Moreover, since F is smooth, the minimum can be obtained by nullifying its gradient:

$$\nabla_{\boldsymbol{\omega}} F(\cdot)|_{\boldsymbol{\omega}^*} = \mathbf{0}.$$

To do so, we start by calling G the gradient of F : $G(\cdot) \doteq \nabla_{\boldsymbol{\omega}} F(\cdot)$. Its h -th component is

$$G_h(\boldsymbol{\omega}) = \partial_{\omega_h} F(\boldsymbol{\omega}) \quad (13a)$$

$$= 2 \int_{s_0}^{s_1} \left(\frac{\sum_{i=0}^N \omega_i \psi_i(s)}{\sum_{i=0}^N \psi_i(s)} s - \tilde{f}(s) \right) \left(\frac{\psi_h(s)}{\sum_{i=0}^N \psi_i(s)} \right) ds. \quad (13b)$$

Function G is linear in $\boldsymbol{\omega}$, thus the minimization problem can be written as a linear system $\mathbf{A}\boldsymbol{\omega} = \mathbf{b}$. The component in row h and column k of \mathbf{A} , a_{hk} , is, for $h, k \in \{0, 1, \dots, N\}$,

$$a_{hk} = \partial_{\omega_k} G_h(\boldsymbol{\omega}) \quad (14)$$

$$= 2 \int_{s_0}^{s_1} \frac{\psi_h(s) \psi_k(s)}{(\sum_{i=0}^N \psi_i(s))^2} s^2 ds, \quad (15)$$

while the h -th component of the vector \mathbf{b} , b_h , is, for $h = 0, 1, \dots, N$,

$$b_h = 2 \int_{s_0}^{s_1} \left(\frac{\psi_h(s)}{\sum_{i=0}^N \psi_i(s)} f(s) s \right) ds. \quad (16)$$

Using any quadrature formula (e.g. Simpson's rule), the integrals in equations (15) and (16) can be computed. Thus, we can solve the linear problem and obtain $\boldsymbol{\omega}^*$.

We remark that this procedure has to be repeated for each component $j = 1, 2, \dots, d$.

4 Execution of a Learned Trajectory

To execute a learned trajectory, we simply solve numerically the differential problem (9) using the weights learned previously. Since the goal position can change during time, the system may need “more time” to reach convergence than the learned one. Thus, instead of solving the problem up to a final time, one may decide to let the system run until convergence to the goal (which is guaranteed by Proposition 2.1). In such a case, the final part of the trajectory may not have an influent forcing term, since $f(s(t)) \approx 0$ when $t > T$.

We remark that the DMPs can be used as a classical planner of trajectories: by ignoring the forcing term, i.e. by setting $f \equiv 0$, we are still guaranteed to converge to the goal position with null velocity.

We decided to use an exponential integrator (in particular the *exponential Euler* method [1]) to compute the evolutions of both the DMP (8) and of the canonical system (2). In order to do so, let us write the 1-dimensional DMP system (8) in matrix form, making explicit the linear component:

$$\tau \begin{bmatrix} \dot{v} \\ \dot{x} \end{bmatrix} = \begin{bmatrix} -D & -K \\ 1 & 0 \end{bmatrix} \begin{bmatrix} v \\ x \end{bmatrix} + \begin{bmatrix} K(g(1-s) + x_0 s + f(s)) \\ 0 \end{bmatrix}. \quad (17)$$

Now, we can write the vector formulation of DMPs (9) as a single system:

$$\tau \mathbf{z} = \Xi \mathbf{z} + \beta(s) \quad (18)$$

where \mathbf{z} contains both positions x and velocities v for each dimension d :

$$\mathbf{z} \doteq [v_1, x_1, v_2, x_2, \dots, v_n, x_n]^\top. \quad (19)$$

Matrix Ξ is a block-diagonal matrix which describes the linear component of the system (9) (the blank spaces should be intended as zeros in (20)):

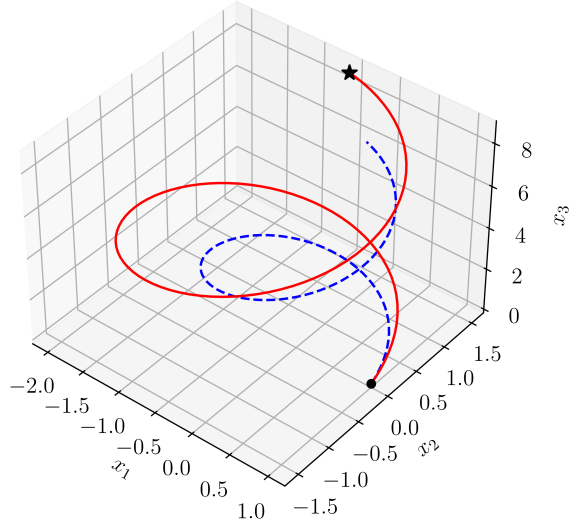
$$\Xi = \begin{bmatrix} -D_{11} & -K_{11} & & & & \\ 1 & 0 & & & & \\ & & -D_{22} & -K_{22} & & \\ & & 1 & 0 & & \\ & & & \ddots & \ddots & \\ & & & & -D_{nn} & -K_{nn} \\ & & & & 1 & 0 \end{bmatrix}, \quad (20)$$

and $\beta(s)$ contains all the remaining terms

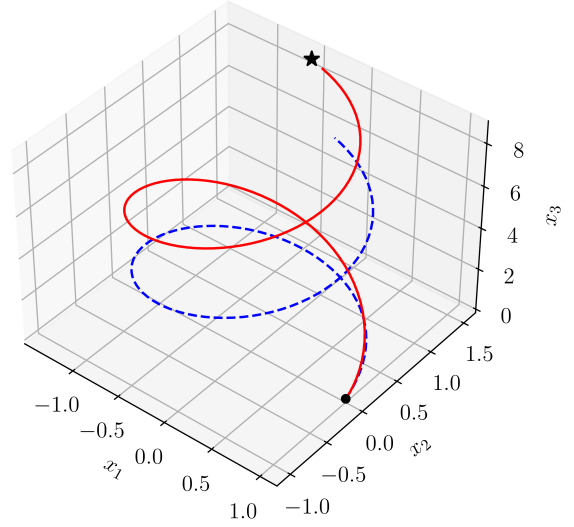
$$\beta(s) = \begin{bmatrix} K_{11}(g_1(1-s) + x_{01} + f_1(s)) \\ 0 \\ K_{22}(g_2(2-s) + x_{02} + f_2(s)) \\ 0 \\ \vdots \\ \vdots \\ K_{nn}(g_n(n-s) + x_{0n} + f_n(s)) \\ 0 \end{bmatrix}. \quad (21)$$

The numerical scheme works as follow. Let us denote by \mathbf{z}_n the solution at the present time t_n , and by \mathbf{z}_{n+1} the solution at time t_{n+1} that need to be computed, and let k being the time step: $k = t_{n+1} - t_n$. In the *Exponential Euler* method, \mathbf{z}_{n+1} is computed as

$$\mathbf{z}_{n+1} = \mathbf{z}_n + k\varphi_1(k\Xi) (\Xi \mathbf{z}_n + \beta(s(t_n))), \quad (22)$$



(a) DMP formulation (1)



(b) DMP formulation (9)

Figure 4: Example of execution of a DMP with $d = 3$. In both figures, the blue dashed line shows the desired trajectory, the black dot the starting point \mathbf{x}_0 , the black star the new goal point \mathbf{g}'

where the function φ_1 is defined as

$$\varphi_1(x) \doteq \int_0^1 e^{(1-\xi)x} d\xi \quad (23)$$

and is computed using Padé approximation.

In Figure 4 is shown an example of execution of DMPs.

5 Other Classes of Basis Functions

As we mentioned in Section 1, Gaussian basis functions are adopted in the literature to approximate the forcing term $f(s)$. However, they are not the only possibility. Indeed, any class $\{\psi_i\}_{i=0,1,\dots,N}$ of basis functions satisfying $\lim_{t \rightarrow \infty} \psi_i(t) = 0$ would be a valid choice. Moreover, compactly supported basis functions are preferable since we are guaranteed that there exists a time T^* such that $\psi_i(t) = 0$ for any $t > T^*$ [5].

In the following, we introduce and compare different classes of compactly supported basis functions. For all of them we maintain the definition of the centers c_i as in (5), but we define the widths

$$\begin{aligned} a_i &= (c_i - c_{i-1})^{-1}, \quad i = 1, 2, \dots, N, \\ a_0 &= a_1. \end{aligned} \quad (24)$$

Well known compactly supported basis functions are *Wendland's functions* [12, 11]. Wendland's functions in one dimension can be defined for any smoothness requirement. We tested the following ones (where the

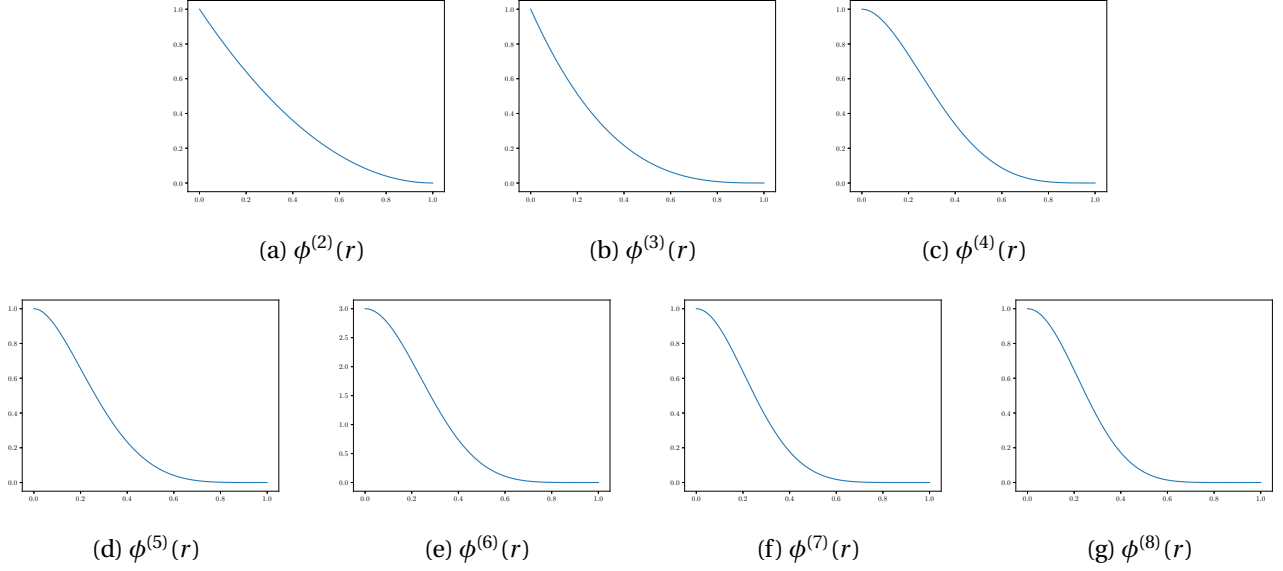


Figure 5: Wendlan's basis functions (25) as functions of r . We omit the dependency on the index i since it change the definition of r , but not the definition of ϕ w.r.t. r .

operator $(\cdot)_+$ denotes the *positive part* functions: $(x)_+ = \max\{0, x\}$:

$$\phi_i^{(2)}(r) = (1 - r)_+^2, \quad (25a)$$

$$\phi_i^{(3)}(r) = (1 - r)_+^3, \quad (25b)$$

$$\phi_i^{(4)}(r) = (1 - r)_+^4(4r + 1), \quad (25c)$$

$$\phi_i^{(5)}(r) = (1 - r)_+^5(5r + 1), \quad (25d)$$

$$\phi_i^{(6)}(r) = (1 - r)_+^6(35r^2 + 18r + 3), \quad (25e)$$

$$\phi_i^{(7)}(r) = (1 - r)_+^7(16r^2 + 7r + 1), \quad (25f)$$

$$\phi_i^{(8)}(r) = (1 - r)_+^8(32r^3 + 25r^2 + 8r + 1), \quad (25g)$$

where we defined

$$r = |a_i(s - c_i)|.$$

In Figure 5, basis functions (25) are shown.

Let us now recall the following example of *mollifier* $\varphi : \mathbb{R} \rightarrow \mathbb{R}$ defined as

$$\varphi(x) = \begin{cases} \frac{1}{I_n} \exp\left(-\frac{1}{1-|x|^2}\right) & \text{if } |x| < 1 \\ 0 & \text{otherwise} \end{cases},$$

where I_n is set so that $\int_{\mathbb{R}} \varphi(x) dx = 1$, which is smooth and compactly supported.

It is then natural to define the set of *mollifier-like basis functions* $\{\varphi_i(s)\}$ given by

$$\varphi_i(s) = \begin{cases} \exp\left(-\frac{1}{1-|a_i(s-c_i)|^2}\right) & \text{if } |a_i(s-c_i)| < 1 \\ 0 & \text{otherwise} \end{cases} \quad (26)$$

where c_i are the centers as in (5), a_i are the widths as in (24), and the usual normalization term I_n is omitted since it is not necessary to our purposes.

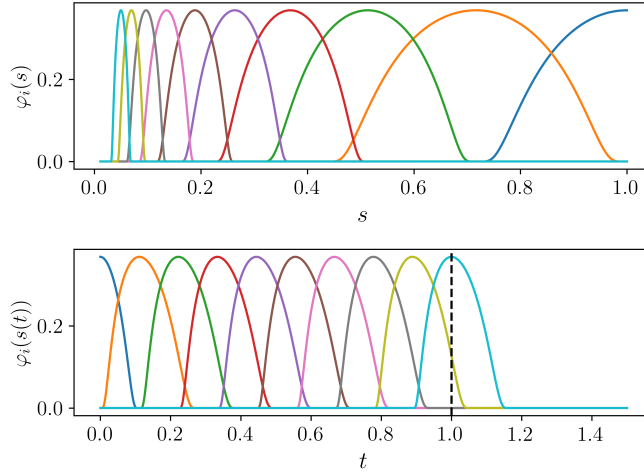


Figure 6: Example of Mollifier-like Basis Functions. The upper plot shows the evolution of the functions w.r.t. the parameter s , while the lower plot shows them as function of t . The parameters are: $T = 1$, $\alpha = 3$, and $N = 9$.

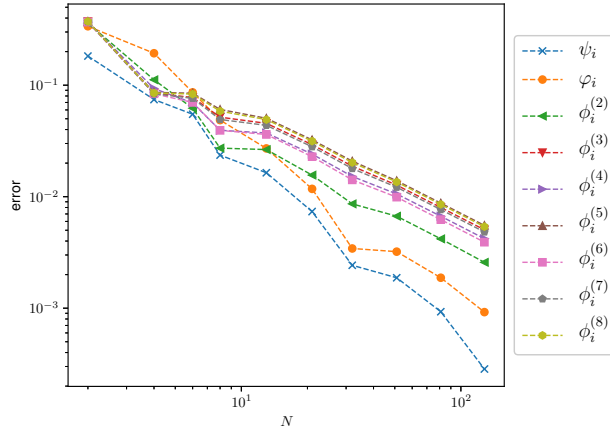


Figure 7: Plot of the L^2 error done by approximating hat function (27). ψ_i refers to Gaussian basis functions (4), φ_i refers to mollifier-like basis functions (26), and $\phi_i^{(k)}$ refers to the respective Wendland's functions (25).

In Figure 6 we plot an example of these functions both as function of s and of t . We remark that all these new basis functions satisfies the same properties we highlighted for Gaussian basis functions in Section 1, i.e. they are symmetric in s , the centers are equispaced in t and the order of the basis functions is flipped when changing from t to s .

We tested the goodness of these classes of basis functions by computing the error done when approximating the following *hat function*:

$$\begin{cases} -4|x - \frac{1}{2}| + 1 & \text{if } x \in (\frac{1}{4}, \frac{3}{4}) \\ 0 & \text{otherwise.} \end{cases} \quad (27)$$

Figure 7 shows the convergence of the approximant to the desired function w.r.t. the L^2 -norm with respect to the number of basis functions N . From this plot emerges that Gaussian basis functions are the best approximant, while mollifier-like basis functions are the best compactly supported approximant.

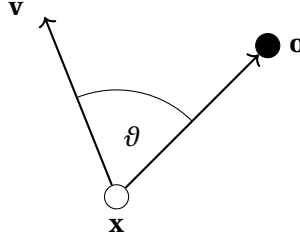


Figure 8: Definition of the steering angle ϑ .

6 Obstacle Avoidance

To account for the avoidance behavior, an additional term $\mathbf{p}(\mathbf{x}, \mathbf{v})$ can be added to the differential equation (9a):

$$\tau \dot{\mathbf{v}} = K(\mathbf{g} - \mathbf{x}) - D\mathbf{v} - K(\mathbf{g} - \mathbf{x}_0) + K\mathbf{f}(s) + \mathbf{p}(\mathbf{x}, \mathbf{v}). \quad (28)$$

6.1 Point Obstacles

Two main approaches exist to implement obstacle avoidance for point obstacles. In [9] a strategy based on potential fields is presented, in which each obstacle generates a potential field, whose negative gradient is used as additional forcing term. On the other hand, in [6] a strategy based on the concept of *steering angle* (that we will explain later) is introduced.

In [9] two potential fields are defined. The first one is the *static potential*

$$U_s(\mathbf{x}) = \begin{cases} \frac{\eta}{2} \left(\frac{1}{r(\mathbf{x})} - \frac{1}{r_0} \right)^2 & \text{if } r(\mathbf{x}) \leq r_0 \\ 0 & \text{otherwise} \end{cases} \quad (29)$$

where $r(\mathbf{x})$ is the distance between the current position \mathbf{x} and the obstacle, r_0 is the radius of influence, and η is a parameter. The additional forcing term is thus defined as $\mathbf{p}(\mathbf{x}, \mathbf{v}) = \mathbf{p}(\mathbf{x}) = -\nabla_{\mathbf{x}} U_s(\mathbf{x})$.

This potential does not allow smooth obstacle avoidance since it ignores the velocity \mathbf{v} of the system. Thus, the following *dynamic potential* is introduced:

$$U_d(\mathbf{x}, \mathbf{v}) = \begin{cases} \lambda (-\cos(\theta))^\beta \frac{\|\mathbf{v}\|}{r(\mathbf{x})} & \text{if } \theta \in (\frac{\pi}{2}, \pi] \\ 0 & \text{otherwise} \end{cases}, \quad (30)$$

where λ, β are parameters, $r(\mathbf{x})$ is defined as before, \mathbf{v} is the velocity of the system. The angle θ is taken between the velocity \mathbf{v} and the position \mathbf{x} relative to the position of the obstacle:

$$\cos(\theta) = \frac{\langle \mathbf{v}, \mathbf{x} \rangle}{\|\mathbf{v}\| r(\mathbf{x})}.$$

The additional forcing term is thus defined as $\mathbf{p}(\mathbf{x}, \mathbf{v}) = -\nabla_{\mathbf{x}} U_d(\mathbf{x}, \mathbf{v})$.

Fajen et al. [2] described the behavior of a human avoiding an obstacle using a differential equation. This equation describes the *steering angle*, depicted in Figure 8, which is modeled to change according to the differential equation

$$\dot{\vartheta} = \gamma \vartheta \exp(-\beta |\vartheta|). \quad (31)$$

For large angles, $\dot{\vartheta}$ approaches zero, so that a movement away from the obstacle needs no corrections.

We can now combine equation (31) with the DMPs [6]. The change in steering direction changes the velocity vector \mathbf{v} as follows:

$$\dot{\mathbf{v}} = \mathbf{R}\mathbf{v}\dot{\vartheta},$$

where \mathbf{R} is the rotational matrix with axis $\mathbf{r} = (\mathbf{o} - \mathbf{x}) \times \mathbf{v}$ and angle of rotation $\frac{\pi}{2}$, where \mathbf{o} is the position of the obstacle.

The obstacle-induced change in velocity is embed in the motion equation by choosing

$$\mathbf{p}(\mathbf{x}, \mathbf{v}) = \gamma \mathbf{R} \mathbf{v} \vartheta \exp(-\beta \vartheta) \quad (32)$$

with

$$\vartheta = \arccos\left(\frac{\langle \mathbf{o} - \mathbf{x}, \mathbf{v} \rangle}{|\mathbf{o} - \mathbf{x}| \cdot |\mathbf{v}|}\right).$$

Proposition 6.1 (Hoffmann et al. [6]). *Equation (9a) with (32) converges to the goal position \mathbf{g} .*

Proof. Since $s \rightarrow 0$ as $t \rightarrow \infty$, we need to study the convergence of the reduced equation

$$\dot{\mathbf{v}} = \mathbf{K}(\mathbf{g} - \mathbf{x}) - \mathbf{D}\mathbf{v} + \gamma \mathbf{R} \mathbf{v} \vartheta \exp(-\beta \vartheta),$$

where we set $\tau = 1$ for simplicity.

Let us consider the Lyapunov function candidate $V(\mathbf{x}, \mathbf{v})$ defined as the energy of the linear spring system (with unit mass) $\dot{\mathbf{v}} = \mathbf{K}(\mathbf{g} - \mathbf{x}) - \mathbf{D}\mathbf{v}$:

$$V(\mathbf{x}, \mathbf{v}) = \frac{1}{2}(\mathbf{g} - \mathbf{x})^\top \mathbf{K}(\mathbf{g} - \mathbf{x}) + \frac{1}{2}\mathbf{v}^\top \mathbf{v}.$$

To prove convergence we have to prove that $\dot{V} < 0$ for $\mathbf{v} \neq 0$. Let us compute

$$\begin{aligned} \dot{V}(\mathbf{x}, \mathbf{v}) &= \langle \nabla_{\mathbf{x}} V(\mathbf{x}, \mathbf{v}), \dot{\mathbf{x}} \rangle + \langle \nabla_{\mathbf{v}} V(\mathbf{x}, \mathbf{v}), \dot{\mathbf{v}} \rangle \\ &= -(\mathbf{g} - \mathbf{x})^\top \mathbf{K} \mathbf{v} + \mathbf{v}^\top \dot{\mathbf{v}} \\ &= -\mathbf{v}^\top \mathbf{K}(\mathbf{g} - \mathbf{x}) + \mathbf{v}^\top \mathbf{K}(\mathbf{g} - \mathbf{x}) - \mathbf{v}^\top \mathbf{D} \mathbf{v} + \gamma \mathbf{v}^\top \mathbf{R} \mathbf{v} \vartheta \exp(-\beta \vartheta) \\ &= -\mathbf{v}^\top \mathbf{D} \mathbf{v} < 0. \end{aligned}$$

The damping matrix \mathbf{D} is positive definite by construction. The term $\mathbf{v}^\top \mathbf{R} \mathbf{v}$ is zero since the matrix \mathbf{R} is a rotation by 90 degrees. If $\mathbf{v} = \mathbf{0}$ and $\mathbf{x} \neq \mathbf{g}$, then $\dot{V} = 0$. However, if $\mathbf{x} \neq \mathbf{g}$ then $\mathbf{v} \neq \mathbf{0}$ and V changes. Thus, according to LaSalle's theorem, \mathbf{x} converges to \mathbf{g} . \square

In the case of moving obstacles, we need to consider the relative velocity between the end-effector \mathbf{x} and the obstacle \mathbf{o} , and not only the end-effector velocity \mathbf{v} itself. Therefore, we compute ϑ in the reference frame of the obstacle obtaining

$$\vartheta = \arccos\left(\frac{\langle \mathbf{o} - \mathbf{x}, \mathbf{v} - \dot{\mathbf{o}} \rangle}{|\mathbf{o} - \mathbf{x}| \cdot |\mathbf{v} - \dot{\mathbf{o}}|}\right),$$

where $\dot{\mathbf{o}}$ is the velocity of the obstacle. The additional forcing term $\mathbf{p}(\mathbf{x}, \mathbf{v})$ is defined as

$$\mathbf{p}(\mathbf{x}, \mathbf{v}) = \gamma \mathbf{R}(\mathbf{v} - \dot{\mathbf{o}}) \vartheta \exp(-\beta \vartheta).$$

We remark that the proof of Proposition 6.1 does not hold for this perturbation term.

We remark that when more than one obstacles are present in the scene, it is sufficient to formulate $\mathbf{p}(\mathbf{x}, \mathbf{v})$ as the sum of the contributions of each obstacle. Moreover, proof of Proposition 6.1 holds true even with multiple obstacles modeled as (32).

6.2 Volume Obstacles

The methods described above can deal with point obstacles. When dealing with real robotic situations, these methods may become numerically demanding since a dense point cloud may be necessary, and a different perturbation term \mathbf{p} has to be computed for each point, increasing the computational time.

For this reason, in [3] and [4], new methods to deal with obstacles with volume were developed.

The idea behind both methods lies in the definition of an *isopotential* C that describe the obstacles. This isopotential is a function $C : \mathbb{R}^d \rightarrow \mathbb{R}$ that vanishes on the boundary of the obstacle and increases as the distance from the obstacle boundary increases. An example of isopotential is the following, which model an ellipse-shaped obstacle:

$$C(\mathbf{x}) = \left(\frac{x_1 - \hat{x}_1}{\ell_1} \right)^2 + \left(\frac{x_2 - \hat{x}_2}{\ell_2} \right)^2 - 1,$$

where the ellipse is centered in (\hat{x}_1, \hat{x}_2) and has semi-axis ℓ_1 and ℓ_2 . In [3], a static potential was proposed:

$$U_S(\mathbf{x}) = \frac{A \exp(-\eta C(\mathbf{x}))}{C(\mathbf{x})}.$$

In [4] a dynamic potential was proposed:

$$U_D(\mathbf{x}, \mathbf{v}) = \begin{cases} \lambda (-\cos \theta)^\beta \frac{\|\mathbf{v}\|}{C^\eta(\mathbf{x})} & \text{if } \theta \in \left(\frac{\pi}{2}, \pi\right] \\ 0 & \text{if } \theta \in \left[0, \frac{\pi}{2}\right] \end{cases},$$

where

$$\cos \theta = \frac{\langle \nabla_{\mathbf{x}} C(\mathbf{x}), \mathbf{v} \rangle}{\|\nabla_{\mathbf{x}} C(\mathbf{x})\| \|\mathbf{v}\|}.$$

In both cases, the perturbation term is written as the negative gradient (w.r.t. the position \mathbf{x}) of the potential:

$$\mathbf{p}_S(\mathbf{x}) = -\nabla_{\mathbf{x}} U_S(\mathbf{x}), \quad \mathbf{p}_D(\mathbf{x}, \mathbf{v}) = -\nabla_{\mathbf{x}} U_D(\mathbf{x}, \mathbf{v}).$$

An example of all these methods can be seen in Figure 9

References

- [1] M. Caliari and A. Ostermann. Implementation of exponential rosenbrock-type integrators. *Applied Numerical Mathematics*, 59(3-4):568–581, 2009.
- [2] B. R. Fajen and W. H. Warren. Behavioral dynamics of steering, obstacle avoidance, and route selection. *Journal of Experimental Psychology: Human Perception and Performance*, 29(2):343, 2003.
- [3] M. Ginesi, D. Meli, A. Calanca, D. Dall’Alba, N. Sansonetto, and P. Fiorini. Dynamic movement primitives: Volumetric obstacle avoidance. In *2019 19th International Conference on Advanced Robotics (ICAR)*, pages 234–239, Dec 2019.
- [4] M. Ginesi, D. Meli, A. Roberti, N. Sansonetto, and P. Fiorini. Dynamic movement primitives: Volumetric obstacle avoidance using dynamic potential functions. *Journal of Intelligent & Robotic Systems*, 101(4):79, Mar 2021.
- [5] M. Ginesi, N. Sansonetto, and P. Fiorini. Overcoming some drawbacks of dynamic movement primitives. *Robotics and Autonomous Systems*, 144:103844, 2021.

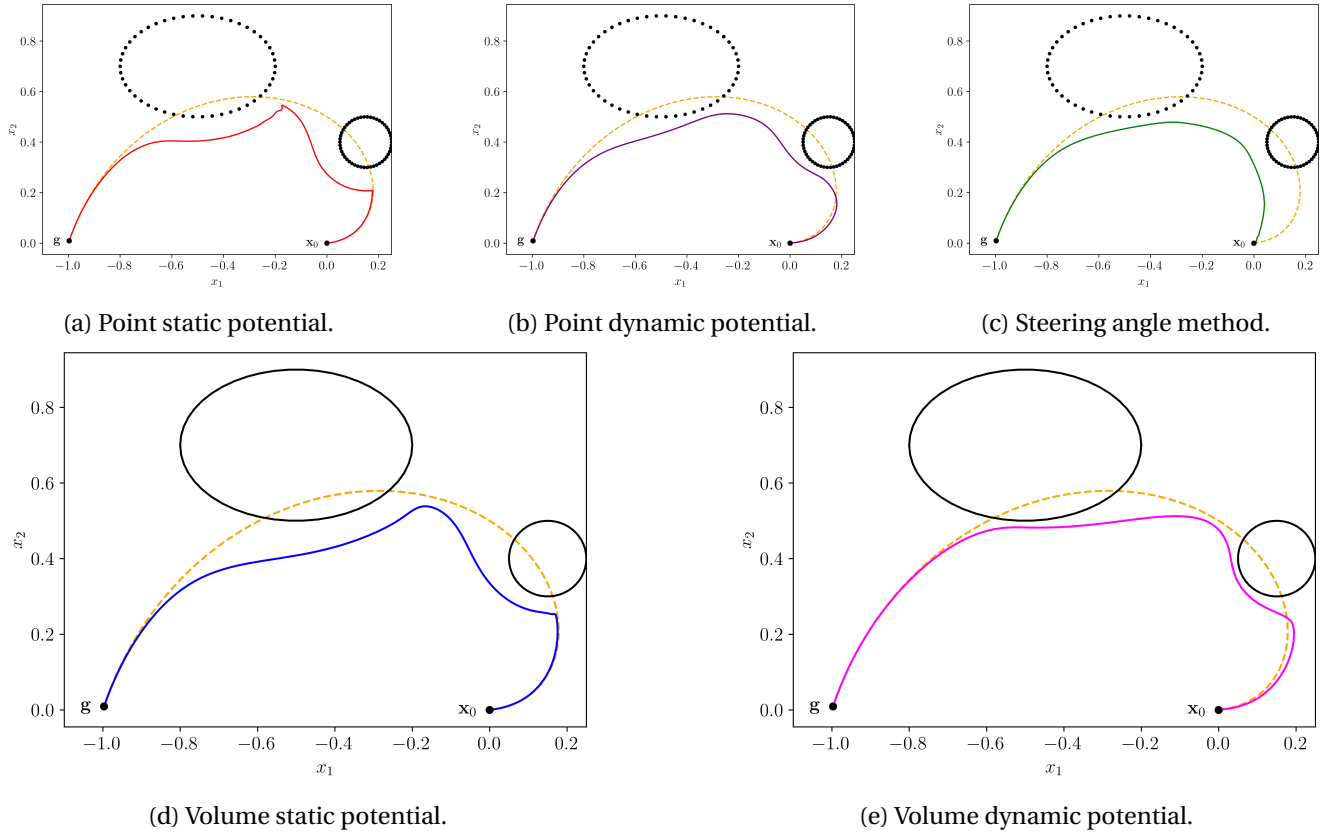


Figure 9: Examples of obstacle avoidance for DMPs.

- [6] 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 *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*, pages 2587–2592. IEEE, 2009.
- [7] A. J. Ijspeert, J. Nakanishi, and S. Schaal. Movement imitation with nonlinear dynamical systems in humanoid robots. In *Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on*, volume 2, pages 1398–1403. IEEE, 2002.
- [8] A. J. Ijspeert, J. Nakanishi, and S. Schaal. Learning attractor landscapes for learning motor primitives. In *Advances in neural information processing systems*, pages 1547–1554, 2003.
- [9] D.-H. Park, H. Hoffmann, P. Pastor, and S. Schaal. Movement reproduction and obstacle avoidance with dynamic movement primitives and potential fields. In *Humanoid Robots, 2008. Humanoids 2008. 8th IEEE-RAS International Conference on*, pages 91–98. IEEE, 2008.
- [10] P. Pastor, H. Hoffmann, T. Asfour, and S. Schaal. Learning and generalization of motor skills by learning from demonstration. In *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*, pages 763–768. IEEE, 2009.
- [11] R. Schaback. A practical guide to radial basis functions. *Electronic Resource*, 11, 2007.
- [12] H. Wendland. Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree. *Advances in computational Mathematics*, 4(1):389–396, 1995.